

**PENGEMBANGAN APLIKASI PERANGKAT BERGERAK
SISTEM INFORMASI *EVENT* DI BIDANG TEKNOLOGI
INFORMASI BERBASIS ANDROID**

SKRIPSI

Disusun oleh:
Ilham Dwi Muchlison
NIM: 185150200111074



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021

PERSETUJUAN

PENGEMBANGAN APLIKASI PERANGKAT BERGERAK SISTEM INFORMASI *EVENT* DI
BIDANG TEKNOLOGI INFORMASI BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ilham Dwi Muchlison
NIM: 185150200111074

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Agi Putra Kharisma, S.T., M.T.
NIK: 2013048604301001

Dosen Pembimbing 2



Issa Arwani, S.Kom., M.Sc.
NIP: 198309222012121003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Madiun, Desember 2021

Ilham Dwi Muchlison

NIM: 185150200111074

PRAKATA

Segala puji bagi Allah Azza Wa Jalla yang telah memberikan rahmat dan karunianya sehingga saya dapat menyelesaikan laporan skripsi dengan judul “Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi *Event* di bidang Teknologi Informasi berbasis Android”. Pada kesempatan ini penulis ingin menyampaikan terima kasih dan rasa hormat kepada semua pihak yang turut memberikan bantuan dalam menyelesaikan skripsi ini, diantaranya kepada:

1. Bapak Agi Putra Kharisma, S.T., M.T. selaku dosen pembimbing pertama dan Bapak Issa Arwani, S.Kom., M.Sc. selaku dosen pembimbing kedua yang telah meluangkan waktu untuk memberikan arahan kepada penulis dan memberikan saran dalam menyelesaikan laporan skripsi ini.
2. Bapak Achmad Basuki, S.T., M.MG., Ph.D. selaku Ketua Jurusan Teknik Informatika dan Bapak Adhitya Bhawiyuga, S.Kom., M.Sc. selaku Ketua Program Studi Teknik Informatika.
3. Keluarga saya yaitu, Bapak Tri Budihartoyo, Ibu Sih Nuryani dan Kakak Siti Mufidah yang telah memberikan dukungan moral dan finansial serta selalu mendoakan untuk kelancaran kuliah.
4. Rahmatsyah selaku mentor yang telah membantu dalam pengembangan kemampuan di bidang Android dan membimbing dalam menyelesaikan skripsi.
5. Kartika Sari yang membantu memotivasi penulis dalam menyelesaikan skripsi ini.
6. Yohannes Sibarani selaku partner yang penuh ambisi untuk melakukan bimbingan skripsi setiap minggu dan diskusi mengenai skripsi ini.
7. Seluruh responden yang membantu dalam pengumpulan data.
8. Annisaa Wulida Furqonia, Alfirsia Damasyifa Fauzulhaq, Nisrina Rosyida Noor Rifai, Hafizotun Tsaqifah, Munirotul Millah dan Halimah S. Mikayla yang menemani dalam berkeluh kesah selama pelajaran skripsi.
9. Ikuta Rira Yoasobi yang bernyanyi pada saat penulis mengerjakan skripsi.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, Desember 2021

Penulis

Ilham.uklison@gmail.com

ABSTRAK

Ilham Dwi Muchlison, Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi Event di bidang Teknologi Informasi berbasis Android

Pembimbing: Agi Putra Kharisma, S.T., M.T. dan Issa Arwani, S.Kom., M.Sc.

Event merupakan kegiatan yang dilakukan dengan bertujuan untuk memberikan informasi dengan cara mendatangkan masyarakat ke suatu tempat. *Event* di bidang teknologi informasi memiliki peran dalam mengembangkan kuantitas dan kualitas dari sumber daya manusia di bidang Teknologi Informasi. Berdasarkan hasil survei yang dilakukan kepada 103 masyarakat yang pernah mengikuti *event* di bidang teknologi informasi dan memiliki umur diatas 17 tahun didapatkan hasil bahwa 71,8 persen responden pernah mengalami kesulitan untuk mendapatkan informasi *event* di bidang teknologi informasi dengan topik yang diinginkan. 78,6 persen responden pernah melewatan *event* yang ingin anda datangi sebelumnya karena terlupa. 92,2 persen responden merasa diperlukan sistem informasi *event* teknologi informasi yang terdapat pembagian kategori dan pengingat *event* yang akan didatangi. Maka dari itu dibuatlah aplikasi yang menghubungkan antara penyelenggara *event* di bidang teknologi informasi dengan masyarakat memiliki minat di bidang teknologi informasi. Aplikasi dikembangkan dengan menggunakan bahasa pemrograman Kotlin, penyimpanan data dengan menggunakan layanan Firebase dan menggunakan *architectural patterns clean architecture*. Aplikasi dikembangkan dengan menggunakan *software development life cycle prototyping*. Aplikasi diuji dengan tiga pengujian yaitu, pengujian validasi dengan menggunakan *black-box*, pengujian *usability* dan *compatibility*. Pada pengujian *black-box* didapatkan hasil 100% *valid*. Pada pengujian *compatibility* dapat diketahui bahwa aplikasi dapat digunakan perangkat yang memiliki sistem operasi Android dengan versi SDK 23 sampai dengan SDK 30. Pada pengujian *usability* didapatkan rata-rata nilai aktual sebesar $86,5 \pm 4,5$ dengan nilai alpha level sebesar 10% untuk aplikasi pencari *event* dan nilai rata-rata aktual sebesar $89 \pm 5,1$ dengan nilai alpha level sebesar 10% untuk aplikasi penyelenggara *event* yang berarti kedua aplikasi masuk dalam kategori *best imaginable* dan memiliki predikat *acceptable*.

Kata kunci: Android, *Event*, Teknologi Informasi, *Clean Architecture*, Firebase

ABSTRACT

Ilham Dwi Muchlison, Development of Mobile Device Application for Event Information Systems in the field Information Technology with Android Based

Supervisor: Agi Putra Kharisma, S.T, M.T. dan Issa Arwani, S.Kom., M.Sc.

Events are activities provide information by bringing people to a place. Events in the field of information technology have a role in developing the quantity and quality of human resources in the field of Information Technology. Based on the results of a survey conducted to 103 people who have attended events in the field of information technology and are over 17 years old, the results show that 71.8 percent of respondents have experienced difficulties in obtaining information on events in the field of information technology with the desired topic. 78.6 percent of respondents have missed an event they wanted to attend before because they forgot it. 92.2 percent of respondents, they need an application where give the information in the field of information technology that includes category division and reminders of events to be attended. Therefore, an application was made that connects event organizers in the field of information technology with people who have an interest in the field of information technology. Applications are developed using the Kotlin programming language, data storage using Firebase services, and using architectural patterns in clean architecture. The application is developed using software development life cycle prototyping. Applications are tested with three tests, namely, validation testing using black-box, usability, and compatibility testing. In black-box testing, the results are 100% valid. Incompatibility testing, it can be seen that the device can be used on the Android operating system with versions of SDK 23 to SDK 30. In usability testing, the average actual value is 86.5 ± 4.5 with an alpha level value of 10% for event finder applications. and the actual average value is 89 ± 5.1 with an alpha level value of 10% for event organizer applications, which means both applications are in the best imaginable category and have acceptable predicate.

DAFTAR ISI

PENGEMBANGAN APLIKASI PERANGKAT BERGERAK SISTEM INFORMASI <i>EVENT</i> DI BIDANG TEKNOLOGI INFORMASI BERBASIS ANDROID	1
PERSETUJUAN	2
PERNYATAAN ORISINALITAS	3
PRAKATA.....	4
ABSTRAK.....	5
ABSTRACT.....	6
DAFTAR ISI.....	7
DAFTAR TABEL.....	12
DAFTAR GAMBAR.....	16
DAFTAR LAMPIRAN	19
BAB 1 PENDAHULUAN.....	20
1.1 Latar Belakang.....	20
1.2 Rumusan Masalah.....	21
1.3 Tujuan	21
1.4 Manfaat.....	21
1.5 Batasan Masalah.....	22
1.6 Sistematika Pembahasan.....	22
BAB 2 LANDASAN KEPUSTAKAAN	24
2.1 Kajian Pustaka	24
2.2 <i>Event</i>	25
2.2.1 <i>Event</i> Teknologi Informasi.....	25
2.3 Analisis Kebutuhan Perangkat Lunak.....	25
2.3.1 Kebutuhan Fungsional.....	25
2.3.2 Kebutuhan Non Fungsional.....	25
2.4 Aplikasi Perangkat Bergerak	25
2.5 Android	26
2.6 Android Studio	27
2.7 Firebase.....	27
2.7.1 <i>Firebase Authentication</i>	28
2.7.2 <i>Firebase Cloud Firestore</i>	28

2.7.3 Firebase <i>Cloud Storage</i>	28
2.8 Arsitektur Pengembangan Aplikasi Android.....	28
2.9 Pengembangan Perangkat Lunak	29
2.9.1 <i>Software Development Life Cycle (SDLC)</i>	29
2.9.2 <i>Prototype</i>	29
2.10 Pengujian Perangkat Lunak.....	30
2.10.1 Validasi	30
2.10.2 <i>Usability</i>	30
BAB 3 METODOLOGI	32
3.1 Strategi Penelitian.....	32
3.2 Subjek Penelitian	33
3.3 Lokasi Penelitian	33
3.4 Metode Pengumpulan Data.....	33
3.5 Peralatan Pendukung.....	34
3.6 Metode Pengembangan	34
3.6.1 Analisis Kebutuhan.....	35
3.6.2 Membangun Prototype.....	35
3.6.3 Evaluasi.....	36
3.6.4 Tinjauan dan Perbaikan.....	36
3.6.5 Perancangan.....	36
3.6.6 Implementasi	36
3.6.7 Pengujian.....	36
3.6.8 Kesimpulan dan Saran.....	37
BAB 4 ANALISIS KEBUTUHAN	38
4.1 Gambaran Umum Sistem.....	38
4.2 Identifikasi Pengguna.....	38
4.3 <i>Prototype</i> Iterasi Pertama.....	39
4.3.1 Kebutuhan Fungsional.....	39
4.3.2 Kebutuhan Non Fungsional.....	42
<i>Usability</i>	42
4.3.3 <i>Use Case Diagram</i>	43
4.3.4 <i>Use Case Scenario</i>	45

4.3.5 <i>Mockup</i> Aplikasi	56
4.3.6 Evaluasi.....	73
4.4 <i>Prototype</i> Iterasi Kedua	73
4.4.1 Kebutuhan Fungsional.....	74
4.4.2 <i>Use Case Diagram</i>	74
4.4.3 <i>Use Case Scenario</i>	76
4.4.4 <i>Mockup</i> Aplikasi	77
4.4.5 Evaluasi.....	80
BAB 5 Perancangan	81
5.1 Perancangan Arsitektur Perangkat Lunak	81
5.2 Perancangan <i>Sequence Diagram</i>	82
5.2.1 <i>Sequence Diagram</i> Registrasi Penyelenggara <i>Event</i>	83
5.2.2 <i>Sequence Diagram</i> Membuat <i>Event</i>	83
5.2.3 <i>Sequence Diagram</i> Melihat Daftar <i>Event</i>	83
5.2.4 <i>Sequence Diagram</i> Menyunting <i>Event</i>	84
5.2.5 <i>Sequence Diagram</i> Menghapus <i>Event</i>	84
5.3 Perancangan <i>Database</i>	84
5.3.2 Perancangan <i>Database</i> Entitas <i>Event Organizer</i>	87
5.3.3 Perancangan <i>Database</i> Entitas <i>Event Hunter</i>	87
5.3.4 Perancangan <i>Database</i> Entitas <i>Event</i>	88
5.3.5 Perancangan <i>Database</i> Entitas <i>Conference Category</i>	89
5.3.6 Perancangan <i>Database</i> Entitas <i>Competition Category</i>	89
5.4 Perancangan <i>Class Diagram</i>	89
5.4.1 <i>Class Diagram</i> Aplikasi Penyelenggara <i>Event</i>	89
5.4.2 <i>Class Diagram</i> Aplikasi Pencari <i>Event</i>	93
5.5 Perancangan <i>Pseudocode</i>	98
5.5.1 Perancangan <i>Pseudocode</i> Registrasi Penyelenggara <i>Event</i>	98
5.5.2 Perancangan <i>Pseudocode</i> Membuat <i>Event</i>	98
5.5.3 Perancangan <i>Pseudocode</i> Melihat Daftar <i>Event</i>	99
5.5.4 Perancangan <i>Pseudocode</i> Menyunting <i>Event</i>	99
5.5.5 Perancangan <i>Pseudocode</i> Menghapus <i>Event</i>	100
BAB 6 implementasi	101

6.1 Spesifikasi Lingkungan Pengembangan Perangkat Lunak	101
6.2 Implementasi <i>Database</i>	102
6.2.1 Implementasi <i>Database</i> Entitas <i>Event Organizer</i>	102
6.2.2 Implementasi <i>Database</i> Entitas <i>Event Hunter</i>	102
6.2.3 Implementasi <i>Database</i> Entitas <i>Event</i>	103
6.2.4 Implementasi <i>Database</i> Entitas <i>Conference Category</i>	103
6.2.5 Implementasi <i>Database</i> Entitas <i>Competition Category</i>	103
6.3 Implementasi Kode Program	104
6.3.1 Implementasi Kode Program Registrasi Penyelenggara <i>Event</i> .	104
6.3.2 Implementasi Kode Program <i>Membuat Event</i>	105
6.3.3 Implementasi Kode Program Melihat Daftar <i>Event</i>	106
6.3.4 Implementasi Kode Program <i>Menyunting Event</i>	107
6.3.5 Implementasi Kode Program Menghapus <i>Event</i>	107
6.4 Implementasi Antarmuka	108
6.4.1 Implementasi Tampilan Login Penyelenggara <i>Event</i>	108
6.4.2 Implementasi Tampilan Registrasi Penyelenggara <i>Event</i>	109
6.4.3 Implementasi Tampilan Beranda Penyelenggara <i>Event</i>	109
6.4.4 Implementasi Tampilan Tambahkan <i>Event</i>	110
6.4.5 Implementasi Tampilan Insight <i>Event</i>	110
6.4.6 Implementasi Tampilan Sunting <i>Event</i>	111
6.4.7 Implementasi Tampilan Profil Penyelenggara <i>Event</i>	111
6.4.8 Implementasi Tampilan Ubah Password Penyelenggara.....	112
6.4.9 Implementasi Tampilan Ubah Nama Penyelenggara.....	112
6.4.10 Implementasi Tampilan Login Pencari <i>Event</i>	113
6.4.11 Implementasi Tampilan Registrasi Pencari <i>Event</i>	113
6.4.12 Implementasi Tampilan Daftar <i>Event Conference</i>	114
6.4.13 Implementasi Tampilan Daftar <i>Event Kompetisi IT</i>	115
6.4.14 Implementasi Tampilan Detail <i>Event Conference</i>	115
6.4.15 Implementasi Tampilan Detail <i>Event Kompetisi IT</i>	116
6.4.16 Implementasi Tampilan Jadwal <i>Event</i>	116
6.4.17 Implementasi Tampilan <i>Event Favorit</i>	117
6.4.18 Implementasi Tampilan Profil Pencari <i>Event</i>	117

6.4.19 Implementasi Tampilan Ubah Nama Pengguna	118
6.4.20 Implementasi Tampilan Ubah Password	118
6.4.21 Implementasi Tampilan Pengaturan.....	119
6.4.22 Implementasi Tampilan Tentang Aplikasi	119
6.4.23 Implementasi Tampilan Pencarian <i>Event</i>	120
BAB 7 Pengujian	121
7.1 Pengujian Validasi	121
7.2 Pengujian <i>Usability</i>	130
7.2.1 Pengujian <i>Usability</i> Pencari <i>Event</i>	131
7.2.2 Pengujian <i>Usability</i> Penyelenggara <i>Event</i>	132
7.3 Pengujian <i>Compatibility</i>	133
BAB 8 Kesimpulan dan saran.....	135
8.1 Kesimpulan.....	135
8.2 Saran	136
DAFTAR REFERENSI	137
LAMPIRAN A WAWANCARA.....	140
A.1 Wawancara Tentang Penyelenggaraan <i>Event</i>	140
LAMPIRAN B KUESIONER	142
B.1 Hasil Kuesioner Kepada Peserta <i>Event</i> Teknologi Informasi	142
LAMPIRAN c HASIL EVALUASI PROTOTYPE	146
C.1 Hasil Evaluasi <i>Prototype</i> Pertama Kepada Pencari <i>Event</i>	146
C.2 Hasil Evaluasi <i>Prototype</i> Pertama Penyelenggara <i>Event</i>	151
C.3 Hasil Evaluasi <i>Prototype</i> Kedua Pencari <i>Event</i>	152
LAMPIRAN D PENGUJIAN USABILITY.....	157
D.1 Pengujian <i>Usability</i> Pencari <i>Event</i>	157

DAFTAR TABEL

Tabel 2.1 <i>Library Firebase BoM</i>	28
Tabel 2.2 <i>Library Firebase Authentication</i>	28
Tabel 2.3 <i>Library Firebase Cloud Firestore</i>	28
Tabel 2.4 <i>Library Firebase Cloud Firestore</i>	28
Tabel 2.5 Daftar Pertanyaan <i>System Usability Scale</i>	30
Tabel 2.6 Skor Pertanyaan <i>System Usability Scale</i>	31
Tabel 3.1 Daftar Pertanyaan Kuesioner	33
Tabel 3.2 Daftar Pertanyaan Wawancara	34
Tabel 4.1 Identifikasi Pengguna	38
Tabel 4.2 Kebutuhan Fungsional Tamu Penyelenggara <i>Event</i>	39
Tabel 4.3 Kebutuhan Fungsional Tamu Pencari <i>Event</i>	40
Tabel 4.4 Kebutuhan Fungsional Penyelenggara <i>Event</i>	40
Tabel 4.5 Kebutuhan Fungsional Pencari <i>Event</i>	41
Tabel 4.6 Kebutuhan Non Fungsional	42
Tabel 4.7 <i>Use Case Scenario Registrasi Penyelenggara Event</i>	45
Tabel 4.8 <i>Use Case Scenario Login Penyelenggara Event</i>	45
Tabel 4.9 <i>Use Case Scenario Registrasi Pencari Event</i>	46
Tabel 4.10 <i>Use Case Scenario Scenario Login Pencari Event</i>	46
Tabel 4.11 <i>Use Case Membuat Event</i>	47
Tabel 4.12 <i>Use Case Menyunting Event</i>	47
Tabel 4.13 <i>Use Case Scenario Menghapus Event</i>	48
Tabel 4.14 <i>Use Case Scenario Melihat Insight Dari Event</i>	48
Tabel 4.15 <i>Use Case Scenario Melihat Daftar Event</i>	49
Tabel 4.16 <i>Use Case Scenario Mengubah Nama Penyelenggara</i>	49
Tabel 4.17 <i>Use Case Scenario Mengubah Password Penyelenggara Event</i>	49
Tabel 4.18 <i>Use Case Scenario Logout Penyelenggara Event</i>	50
Tabel 4.19 <i>Use Case Scenario Melihat Daftar Event Conference</i>	50
Tabel 4.20 <i>Use Case Scenario Melihat Event Kompetisi IT</i>	51
Tabel 4.21 <i>Use Case Scenario Melihat Detail Event Conference</i>	51
Tabel 4.22 <i>Use Case Scenario Melihat Detail Event Kompetisi IT</i>	51
Tabel 4.23 <i>Use Case Scenario Menambahkan Notifikasi Pengingat</i>	52

Tabel 4.24 <i>Use Case Scenario</i> Menghapus Notifikasi Pengingat	52
Tabel 4.25 <i>Use Case Scenario</i> Mendaftar Event	53
Tabel 4.26 <i>Use Case Scenario</i> Melihat Jadwal Event	53
Tabel 4.27 <i>Use Case Scenario</i> Menambahkan Favorit	54
Tabel 4.28 <i>Use Case Scenario</i> Menghapus Favorit Event	54
Tabel 4.29 <i>Use Case Scenario</i> Melihat Daftar Favorit Event	55
Tabel 4.30 <i>Use Case Scenario</i> Mengubah Nama Pengguna	55
Tabel 4.31 <i>Use Case Scenario</i> Mengubah Password Pencari Event	55
Tabel 4.32 <i>Use Case Scenario</i> Logout Pencari Event	56
Tabel 4.33 Kebutuhan Fungsional Pencari Event	74
Tabel 4.34 <i>Use Case Scenario</i> Melakukan Pencarian Event Conference	76
Tabel 4.35 <i>Use Case Scenario</i> Melakukan Pencarian Event Kompetisi IT	76
Tabel 5.1 Entitas <i>Event Organizer</i>	85
Tabel 5.2 Entitas <i>Event Hunter</i>	85
Tabel 5.3 Entitas <i>Event</i>	85
Tabel 5.4 Entitas <i>Conference Category</i>	86
Tabel 5.5 Entitas <i>Competition Category</i>	87
Tabel 5.6 <i>Pseudocode</i> Registrasi Penyelenggara Event	98
Tabel 5.7 <i>Pseudocode</i> Membuat Event	98
Tabel 5.8 <i>Pseudocode</i> Melihat Daftar Event	99
Tabel 5.9 <i>Pseudocode</i> Menyunting Event	99
Tabel 5.10 <i>Pseudocode</i> Menghapus Event	100
Tabel 6.1 Spesifikasi Perangkat Keras Pengembang Perangkat Lunak	101
Tabel 6.2 Spesifikasi Perangkat Keras Perangkat Bergerak	101
Tabel 6.3 Spesifikasi Perangkat Lunak	101
Tabel 6.4 Kode Program Registrasi Penyelenggara Event	104
Tabel 6.5 Kode Program Membuat Event	105
Tabel 6.6 Kode Program Fungsi GetMyEvents	106
Tabel 6.7 Kode Program GetDocumentsWhereIds	106
Tabel 6.8 Kode Program Fungsi UpdateEvent	107
Tabel 6.9 Kode Program Fungsi SetDocument	107
Tabel 6.10 Kode Program Menghapus Event	107

Tabel 7.1 Pengujian Black Box Registrasi Penyelenggara Event	121
Tabel 7.2 Pengujian Black Box Login Penyelenggara Event	121
Tabel 7.3 Pengujian Registrasi Pencari Event	122
Tabel 7.4 Pengujian Black Box Login Pencari Event.....	122
Tabel 7.5 Pengujian Black Box Membuat Event.....	123
Tabel 7.6 Pengujian Black Box Menyunting Event	123
Tabel 7.7 Pengujian Black Box Menghapus Event	123
Tabel 7.8 Pengujian Black Box Melihat Insight Dari Event	124
Tabel 7.9 Pengujian Black Box Melihat Daftar Event.....	124
Tabel 7.10 Pengujian Black Box Mengubah Nama Penyelenggara.....	124
Tabel 7.11 Pengujian Black Box Mengubah Password Penyelenggara Event....	125
Tabel 7.12 Pengujian Black Box Logout Penyelenggara Event	125
Tabel 7.13 Pengujian Black Box Melihat Daftar Event Conference.....	125
Tabel 7.14 Pengujian Black Box Melihat Daftar Event Kompetisi IT.....	125
Tabel 7.15 Pengujian Black Box Melihat Detail Event Conference	126
Tabel 7.16 Pengujian Black Box Melihat Detail Event Kompetisi IT.....	126
Tabel 7.17 Pengujian Black Box Menambahkan Notifikasi Pengingat.....	126
Tabel 7.18 Pengujian Black Box Menghapus Notifikasi Pengingat	127
Tabel 7.19 Pengujian Black Box x Mendaftar Event.....	127
Tabel 7.20 Pengujian Black Box Melihat Jadwal Event	128
Tabel 7.21 Pengujian Black Box Menambahkan Favorit Event.....	128
Tabel 7.22 Pengujian Black Box Menghapus favorit event.....	128
Tabel 7.23 Pengujian Black Box Melihat Daftar Favorit Event.....	128
Tabel 7.24 Pengujian Black Box Mengubah Nama Pengguna	129
Tabel 7.25 Pengujian Black Box Mengubah Password Pencari Event	129
Tabel 7.26 Pengujian Black Box Logout Pencari Event	129
Tabel 7.27 Pengujian Black Box Melakukan Pencarian Event Conference	129
Tabel 7.28 Pengujian Black Box Melakukan Pencarian Event Kompetisi IT.....	130
Tabel 7.29 Rekapitulasi Nilai Kuesioner Responden Pencari Event	131
Tabel 7.30 Rekapitulasi Nilai SUS Responden Pencari Event	131
Tabel 7.31 Rekapitulasi Nilai Kuesioner Responden Penyelenggara Event	132
Tabel 7.32 Rekapitulasi Nilai SUS Responden Penyelenggara Event	133

Tabel 7.33 Hasil Pengujian Compatibility..... 133

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Aplikasi Android	27
Gambar 2.2 <i>Data Flow Clean Architecture</i> Android.....	29
Gambar 2.3 Ilustrasi Model Prototype	30
Gambar 3.1 Alur Metode Pengembangan	35
Gambar 4.1 Gambaran Umum Sistem	38
Gambar 4.2 <i>Use Case Diagram</i> Aplikasi Penyelenggara <i>Event</i>	43
Gambar 4.3 <i>Use Case Diagram</i> Aplikasi Pencari <i>Event</i>	44
Gambar 4.4 <i>Information Architecture</i> Aplikasi Penyelenggara <i>Event</i>	57
Gambar 4.5 <i>Information Architecture</i> Aplikasi Pencari <i>Event</i>	57
Gambar 4.6 <i>High-Fidelity</i> Halaman Login Penyelenggara <i>Event</i>	58
<i>Gambar 4.7 High-Fidelity</i> Halaman Registrasi Penyelenggara <i>Event</i>	58
Gambar 4.8 <i>High-Fidelity</i> Halaman Beranda Penyelenggara <i>Event</i>	59
Gambar 4.9 <i>High-Fidelity</i> Halaman Tambahkan <i>Event</i>	60
Gambar 4.10 <i>High-Fidelity</i> Halaman <i>Insight Event</i>	61
Gambar 4.11 <i>High-Fidelity</i> Halaman Sunting <i>Event</i>	62
Gambar 4.12 <i>High-Fidelity</i> Halaman Profil Penyelenggara.....	63
Gambar 4.13 <i>High-Fidelity</i> Halaman Ubah <i>Password</i> Penyelenggara	63
Gambar 4.14 <i>High-Fidelity</i> Halaman Ubah Nama Penyelenggara	64
Gambar 4.15 <i>High-Fidelity</i> Halaman <i>Login</i> Pencari <i>Event</i>	65
Gambar 4.16 <i>High-Fidelity</i> Halaman Registrasi Pencari <i>Event</i>	65
Gambar 4.17 <i>High-Fidelity</i> Halaman Daftar <i>Event Conference</i>	66
Gambar 4.18 <i>High-Fidelity</i> Halaman Daftar <i>Event Kompetisi IT</i>	67
<i>Gambar 4.19 High-Fidelity</i> Halaman <i>Detail Event</i>	67
Gambar 4.20 <i>High-Fidelity</i> Halaman Detail <i>Event</i>	68
Gambar 4.21 <i>High-Fidelity</i> Halaman Jadwal <i>Event</i>	69
Gambar 4.22 <i>High-Fidelity</i> Halaman Event Favorit.....	69
Gambar 4.23 <i>High-Fidelity</i> Halaman Profil Pencari <i>Event</i>	70
Gambar 4.24 <i>High-Fidelity</i> Halaman Ubah Nama Pengguna	71
Gambar 4.25 <i>High-Fidelity</i> Halaman Ubah <i>Password</i> Pengguna	71
Gambar 4.26 <i>High-Fidelity</i> Halaman Pengaturan	72
Gambar 4.27 <i>High-Fidelity</i> Halaman Tentang.....	73

Gambar 4.28 Use Case Diagram Aplikasi Pencari Event	75
Gambar 4.29 <i>Information Architecture</i> Aplikasi Pencari Event Iterasi kedua	77
Gambar 4.30 Perbaikan <i>High-Fidelity</i> Halaman Daftar Event Conference	77
Gambar 4.31 Perbaikan <i>High-Fidelity</i> Halaman Daftar Event Kompetisi IT	78
Gambar 4.32 <i>High-Fidelity</i> Halaman Pencarian Event	79
Gambar 4.33 Perbaikan <i>High-Fidelity</i> Halaman Jadwal Event	79
Gambar 5.1 Alur Pengelolaan Data Clean Architecture Android	81
Gambar 5.2 <i>Sequence Diagram</i> Registrasi Penyelenggara Event	83
Gambar 5.3 <i>Sequence Diagram</i> Membuat Event	83
Gambar 5.4 <i>Sequence Diagram</i> Melihat Daftar Dari Event	83
Gambar 5.5 <i>Sequence Diagram</i> Menyunting Event	84
Gambar 5.6 <i>Sequence Diagram</i> Menghapus Event	84
Gambar 5.7 Skema Struktur Data JSON Entitas Event Organizer	87
Gambar 5.8 Skema Struktur Data JSON Entitas Event Hunter	88
Gambar 5.9 Skema Struktur Data JSON Entitas Event	88
Gambar 5.10 Skema Struktur Data JSON Entitas Conference Category	89
Gambar 5.11 Skema Struktur Data JSON Entitas Competition Category	89
Gambar 5.12 <i>Class Diagram</i> Modul Domain Aplikasi Penyelenggara Event	90
Gambar 5.13 <i>Class Diagram</i> Modul Domain Aplikasi Penyelenggara Event	91
Gambar 5.14 <i>Class Diagram</i> Modul Features Aplikasi Penyelenggara Event	92
Gambar 5.15 Relasi <i>Class Diagram</i> Aplikasi Penyelenggara Event	93
Gambar 5.16 <i>Class Diagram</i> Modul Features Aplikasi Pencari Event	94
Gambar 5.17 <i>Class Diagram</i> Modul Features Aplikasi Pencari Event	95
Gambar 5.18 <i>Class Diagram</i> Modul Features Aplikasi Pencari Event	96
Gambar 5.19 Relasi <i>Class Diagram</i> Aplikasi Pencari Event	97
Gambar 6.1 Implementasi Database Entitas Event Organizer	102
Gambar 6.2 Implementasi Database Entitas Event Hunter	102
Gambar 6.3 Implementasi Database Entitas Event	103
Gambar 6.4 Implementasi Database Entitas Conference Category	103
Gambar 6.5 Implementasi Database Entitas Competition Category	104
Gambar 6.6 Implementasi Tampilan Login Penyelenggara Event	108
Gambar 6.7 Implementasi Tampilan Registrasi Penyelenggara Event	109

Gambar 6.8 Implementasi Tampilan Beranda Penyelenggara <i>Event</i>	109
Gambar 6.9 Implementasi Tampilan Tambahkan <i>Event</i>	110
Gambar 6.10 Implementasi Tampilan Insight <i>Event</i>	110
Gambar 6.11 Implementasi Tampilan Sunting <i>Event</i>	111
Gambar 6.12 Implementasi Tampilan Profil Penyelenggara <i>Event</i>	112
Gambar 6.13 Implementasi Tampilan Ubah Password Penyelenggara.....	112
Gambar 6.14 Implementasi Tampilan Ubah Nama Penyelenggara.....	113
Gambar 6.15 Implementasi Tampilan Login Pencari <i>Event</i>	113
Gambar 6.16 Implementasi Tampilan Registrasi Pencari <i>Event</i>	114
Gambar 6.17 Implementasi Tampilan Daftar <i>Event Conference</i>	114
Gambar 6.18 Implementasi Tampilan Daftar <i>Event Kompetisi IT</i>	115
Gambar 6.19 Implementasi Tampilan Detail <i>Event Conference</i>	115
Gambar 6.20 Implementasi Tampilan Detail <i>Event Kompetisi IT</i>	116
Gambar 6.21 Implementasi Tampilan Jadwal <i>Event</i>	116
Gambar 6.22 Implementasi Tampilan <i>Event Favorit</i>	117
Gambar 6.23 Implementasi Tampilan Profil Pencari <i>Event</i>	117
Gambar 6.24 Implementasi Tampilan Ubah Nama Pengguna.....	118
Gambar 6.25 Implementasi Tampilan Ubah <i>Password</i>	118
Gambar 6.26 Implementasi Tampilan Pengaturan	119
Gambar 6.27 Implementasi Tampilan Tentang Aplikasi	119
Gambar 6.28 Implementasi Tampilan Pencarian <i>Event</i>	120

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA.....	140
A.1 Wawancara Tentang Penyelenggaraan Event.....	140
LAMPIRAN B KUESIONER	142
B.1 Hasil Kuesioner Kepada Peserta Event Teknologi Informasi	142
LAMPIRAN c HASIL EVALUASI PROTOTYPE	146
C.1 Hasil Evaluasi Prototype Pertama Kepada Pencari Event.....	146
C.2 Hasil Evaluasi Prototype Pertama Penyelenggara Event.....	151
C.3 Hasil Evaluasi Prototype Kedua Pencari Event	152
LAMPIRAN D PENGUJIAN USABILITY.....	157
D.1 Pengujian Usability Pencari Event.....	157

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini sangatlah pesat. Berbagai sektor kehidupan mulai membutuhkan banyak sumber daya manusia dibidang teknologi informasi, penelitian yang dilakukan oleh Pusat Penelitian dan Pengembangan Aplikasi Informatika dan Informasi dan Komunikasi publik memprediksi bahwa pada tahun 2020 dibutuhkan sumber daya manusia di bidang TIK sebanyak 129.465 (Maulidia,2019).

Event adalah kegiatan yang dilakukan dengan bertujuan untuk memberikan informasi dan tujuan lain dari penyelenggara dengan cara mendatangkan masyarakat ke suatu tempat (Riadi,2019). Begitu pula *Event* di bidang teknologi informasi seperti seminar, lokakarya dan perlomba di bidang teknologi informasi, *Event* di bidang teknologi informasi memiliki peran dalam mengembangkan kuantitas dan kualitas dari sumber daya manusia di bidang teknologi informasi. Namun tidak menutup kemungkinan terdapat permasalahan permasalahan yang dihadapi ketika menyelenggarakan *event* di bidang teknologi informasi.

Berdasarkan hasil survei yang dilakukan kepada 103 masyarakat yang pernah mengikuti *event* di bidang teknologi informasi dan memiliki umur diatas 17 tahun didapatkan hasil bahwa 71,8 persen responden pernah mengalami kesulitan untuk mendapatkan informasi *event* di bidang teknologi informasi dengan topik yang diinginkan. 78,6 persen responden pernah melewatkhan *event* yang ingin anda datangi sebelumnya karena terlupa. 92,2 persen responden merasa diperlukan sistem informasi *event* teknologi informasi yang terdapat pembagian kategori dan pengingat *event* yang akan didatangi.

Wawancara juga dilakukan kepada salah satu pelaksana *event* HOLOGY 3.0, salah satu dari *event* dibidang teknologi informasi dengan skala acara nasional yang berisi seminar, perlomba dan *academy*. Ketua pelaksana dari HOLOGY 3.0 dalam wawancaranya menyampaikan mengalami kendala kurangnya jumlah peserta dalam beberapa *event* yang diselenggarakannya dikarenakan kurangnya informasi yang tersebar dengan target yang berhubungan dengan *event* yang diselenggarakan.

Sebenarnya penyebaran informasi *event* sudah dilakukan menggunakan sosial media, website resmi penyelenggara dan media partner dalam menyebarkan informasi tentang *event* yang diselenggarakan. Namun informasi tentang *event* yang diselenggarakan kerap tidak sesuai dengan target penerimanya. Hal tersebut membuat *event* yang diselenggarakan tidak sesuai dengan target penyelenggara dan masyarakat yang memiliki minat pada bidang teknologi informasi kurang mendapatkan informasi *event* yang sesuai dengan minat mereka.

Penjabaran diatas merupakan dorongan untuk melakukan penelitian guna menyelesaikan masalah tersebut dengan mengembangkan perangkat lunak bergerak berbasis Android untuk menghubungkan penyelenggara acara dan masyarakat yang berminat mengikuti *event* di bidang teknologi informasi. Bentuk perangkat lunak bergerak berbasis Android dipilih karena aplikasi dapat memberikan *interactive engagement* lebih kepada pengguna dan dapat memaksimalkan fitur bawaan dari Android seperti notifikasi yang dapat mendukung fitur yang dibutuhkan oleh pengguna. Selain itu, perangkat bergerak Android didasari karena jumlah penggunaan sistem operasi perangkat bergerak Android di indonesia mencapai 91,6% (Statcounter, 2021), sehingga perangkat lunak dapat menjangkau lebih banyak pengguna

Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi *Event* di bidang teknologi informasi diharapkan mampu menyelesaikan permasalahan yang dialami penyelenggara *event* di bidang teknologi informasi maupun masyarakat yang memiliki minat pada *event* di bidang teknologi informasi.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi sebagai berikut:

1. Apa saja kebutuhan fungsional dan non fungsional pada pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi?
2. Bagaimana hasil pengujian dari aplikasi sistem informasi *event* di bidang teknologi informasi yang telah dikembangkan?

1.3 Tujuan

Tujuan dari penelitian pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi sebagai berikut:

1. Mengetahui kebutuhan fungsional dan non fungsional pada pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi.
2. Mengetahui hasil pengujian dari aplikasi sistem informasi *event* di bidang teknologi informasi yang telah dikembangkan

1.4 Manfaat

Manfaat dari penelitian pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi adalah sebagai berikut:

1. Bagi pengguna
 - a. Masyarakat : mempermudah masyarakat untuk mendapatkan informasi terkait *event* di bidang teknologi informasi sesuai dengan topik yang diinginkan dan pengingat akan *event* yang akan di hadiri.

- b. Penyelenggara *event* : mempermudah menyebarkan informasi terkait *event* yang diadakan dengan menjangkau peserta yang sesuai dengan topik dibidangnya.
2. Bagi Peneliti
 - a. Mampu menerapkan ilmu yang didapatkan pada masa perkuliahan
 - b. Menambah wawasan dalam bidang pengembangan perangkat lunak bergerak
 - c. Memenuhi salah satu persyaratan dalam mendapatkan gelar sarjana komputer

1.5 Batasan Masalah

Batasan masalah dari penelitian pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi adalah sebagai berikut:

1. Perangkat lunak dapat digunakan pada platform sistem operasi Android dengan API 23 sampai dengan API 30.
2. Penggunaan perangkat lunak memerlukan koneksi internet.
3. Pengembangan perangkat lunak dibuat dan dijalankan menggunakan Android Studio.
4. Target pengguna dari perangkat lunak adalah masyarakat yang memiliki minat pada bidang teknologi informasi.

1.6 Sistematika Pembahasan

Sistematika Pembahasan berisi struktur dari dokumen penelitian dari Bab pendahuluan sampai dengan Bab Penutup dan berisi penjelasan singkat dari tiap Bab.

1. BAB 1 – PENDAHULUAN

Bab pendahuluan berisi tentang latar belakang dari penelitian, rumusan masalah, batasan masalah, tujuan dari penelitian, manfaat dari penelitian, dan sistematika pembahasan dari penelitian.

2. BAB 2 – LANDASAN KEPUSTAKAAN

Bab landasan kepustakaan berisi tentang teori yang berkaitan dengan penelitian, teknologi pendukung yang digunakan pada penelitian dan juga penelitian terdahulu yang berkaitan dengan penelitian.

3. BAB 3 – METODOLOGI

Bab metodologi berisi tentang strategi penelitian, subjek penelitian, lokasi penelitian, metode pengumpulan data, peralatan pendukung dan metode pengembangan yang digunakan pada penelitian.

4. BAB 4 – ANALISIS KEBUTUHAN

Bab analisis kebutuhan berisi tentang deskripsi sistem dan analisis kebutuhan dari pengembangan perangkat lunak pada penelitian.

5. BAB 5 – PERANCANGAN

Bab perancangan berisi tentang hasil perancangan dari pengembangan perangkat lunak pada penelitian.

6. BAB 6 – IMPLEMENTASI

Bab implementasi berisi tentang hasil implementasi dari pengembangan perangkat lunak pada penelitian.

7. BAB 7 – PENGUJIAN

Bab pengujian berisi tentang proses dan hasil pengujian dari pengembangan perangkat lunak pada penelitian.

8. BAB 8 – PENUTUP

Bab penutup berisi tentang kesimpulan dan saran terhadap hasil dari penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Langkah awal yang dilakukan peneliti untuk menyelesaikan permasalahan adalah mencari tahu tentang proses dan tahapan yang diperlukan dengan menggunakan beberapa penelitian terdahulu yang berkaitan dengan penelitian ini.

Penelitian pertama yang berkaitan dengan publikasi *event* adalah penelitian dengan judul “Rancangan Bangun Sistem Informasi Manajemen Publikasi *Event* Berbasis Android”. Penelitian tersebut dikembangkan dengan metode *waterfall* dan dibangun dengan bahasa pemrograman PHP, HTML, dan menggunakan *database* MySQL. Penelitian tersebut menjelaskan bahwa aplikasi *event* berhasil mempermudah memberikan informasi tentang sebuah acara kepada peserta (Martoko,2019).

Selanjutnya penelitian yang berkaitan dengan publikasi informasi *event* adalah penelitian dengan judul “Pengembangan Sistem Informasi *Event* Seminar dan Workshop (*Eventify*) Berbasis Android”. Penelitian tersebut dikembangkan dengan metode scrum dan menggunakan bahasa pemrograman Java. Pada penelitian tersebut aplikasi *Eventify* berbasis Android dapat membantu penyelenggara dan peserta *event* (Lingga, 2020).

Perbedaan yang terdapat pada dua penelitian sebelumnya dengan penelitian ini terdapat pada jenis *event* yang diselenggarakan pada penelitian ini lebih berfokus pada *event* pada bidang teknologi informasi. Selain itu, pada penelitian ini pengembangan dilakukan dengan menggunakan metode *prototyping* dan menggunakan bahasa pemrograman Kotlin.

Selanjutnya terdapat penelitian yang berkaitan tentang pengelolaan informasi dan promosi *event* yaitu penelitian dengan judul “Model Sistem Informasi Promosi Dan Management *Event* Berbasis Web”. Penelitian tersebut dikembangkan dengan menggunakan platform *website*. Penelitian tersebut menjelaskan bahwa perangkat lunak *event* mempermudah dalam mengelola *event* dan media penyebaran informasi untuk membantu PT. Prima Multi Artha dalam mengelola pencatatan *event* dengan menggunakan website (Firmansyah dan Arnie, 2017).

Terdapat juga penelitian yang berkaitan dengan penggunaan dua aktor utama dalam aplikasi dengan judul “Pengembangan Aplikasi Perangkat Bergerak Layanan Berbasis Lokasi Penghubung Sukarelawan Dengan Kegiatan Sukarela”. Penelitian tersebut dikembangkan dengan bahasa pemrograman Kotlin dan metode pengembangan *prototyping*. Penelitian tersebut menjelaskan bahwa pada penelitian ini dikembangkan dua aplikasi yang mewakili dua perilaku aktor dan dapat menjadi penghubung antara sukarelawan dan organisasi sosial (Rahmatsyah, 2019).

2.2 Event

Event adalah kegiatan di waktu tertentu yang untuk memperingati hal-hal penting yang dialami oleh individu ataupun kelompok yang diselenggarakan dengan suatu tujuan dan melibatkan masyarakat. *Event* memiliki beberapa jenis yaitu *leisure event*, *cultural event*, *personal event*, dan *organizational event*. *Leisure event* merupakan *event* yang terdapat unsur kompetisi didalamnya. *Cultural event* merupakan *event* yang memiliki unsur budaya dan terdapat nilai sosial. *Personal event* merupakan *event* perseorangan yang melibatkan keluarga dan teman. *Organizational event* merupakan *event* yang diadakan oleh suatu organisasi dan memiliki tujuan sesuai dengan tujuan organisasi tersebut (Noor,2009)

2.2.1 Event Teknologi Informasi

Teknologi informasi adalah semua jenis teknologi yang digunakan untuk membuat, mengubah dan menyebarkan informasi dengan tujuan mempermudah dan membantu kehidupan manusia. Teknologi informasi mengelola data dan informasi dengan komputasi dan komunikasi. Penerapan dari teknologi informasi dapat berupa perangkat lunak maupun perangkat keras (Irwansyah & Moniaga,2014).

Dalam disiplin ilmu teknologi informasi terdapat pula *event* yang dapat dalam mengembangkan kuantitas dan kualitas dari sumber daya manusia di bidang teknologi informasi. *Event* teknologi informasi merupakan kegiatan yang diselenggarakan oleh suatu kelompok yang berfokus pada keilmuan di bidang teknologi informasi. Beberapa *event* yang terdapat dalam *event* teknologi informasi yaitu seminar, lokakarya dan perlombaan di bidang teknologi informasi.

2.3 Analisis Kebutuhan Perangkat Lunak

2.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan jenis kebutuhan yang dibutuhkan dalam perangkat lunak, kebutuhan fungsional berisi tentang proses - proses yang dijalankan didalam sistem dan hasil dari setiap proses proses tersebut (Setiyani dan Tjandra, 2021).

2.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah jenis kebutuhan yang berisi tentang karakteristik dan keadaan dari sistem kedepannya. Pada kebutuhan non fungsional juga diuraikan batasan - batasan dari sistem (Aulia Aziiza dan Nur Fadhilah, 2020).

2.4 Aplikasi Perangkat Bergerak

Aplikasi perangkat bergerak merupakan perangkat lunak yang dapat dijalankan pada perangkat bergerak seperti *smartphone*, tablet dan perangkat bergerak sejenis lainnya. Pada perangkat bergerak dapat memberikan informasi dan

layanan secara praktis dan efisien. Karakteristik pada aplikasi perangkat bergerak adalah kemampuan komputasi yang relatif rendah daripada desktop, memiliki kemampuan komunikasi nirkabel dan ukuran dari perangkat bergerak relatif kecil, dan menggunakan baterai sebagai daya utama (Tolle et al., 2017).

Aplikasi perangkat bergerak memiliki beberapa jenis antara lain *web mobile*, *native*, *hybrid*. Aplikasi *web mobile* merupakan aplikasi yang dijalankan pada halaman web namun ukuran dari aplikasi menyesuaikan ukuran layar dari perangkat keras. Aplikasi *native* merupakan aplikasi yang dikembangkan pada platform sistem operasi secara khusus. Aplikasi *hybrid* merupakan aplikasi yang dikembangkan dan dapat digunakan pada beberapa platform sekaligus (Tolle, et al., 2017).

Aplikasi perangkat bergerak yang akan dikembangkan pada penelitian ini merupakan aplikasi perangkat bergerak *native*. Platform yang dipilih pada pengembangan aplikasi perangkat bergerak adalah Android dikarenakan jumlah pengguna dari platform Android di Indonesia merupakan yang terbesar yaitu mencapai 91,6% pengguna (Statcounter, 2021). Perangkat bergerak *native* dipilih karena memiliki fitur yang lebih banyak daripada *web mobile* dan memiliki performa lebih baik daripada *web mobile* dan aplikasi *hybrid*.

2.5 Android

Android adalah sistem operasi yang bersifat *open source* yang dirancangan dan dikembangkan untuk perangkat bergerak. Android dikembangkan oleh Android, Inc.. Pada platform pengembangan yang terbuka dan bebas membuat pengembangan Android dapat dilakukan secara inovatif maupun kaya (Effendi, 2013).

Dalam arsitektur Android, Android memiliki beberapa lapisan komponen penyusun yaitu lapisan *kernel Linux* yang berisi *drivers* dan *power management*, lalu pada lapisan *hardware* berisi perangkat keras pendukung Android, lalu pada bagian *Android Runtime* berisi *ART* dan *Core Libraries*, pada lapisan *Native C/C++ Libraries* berisi *library* yang dibutuhkan dalam sistem operasi Android. Lapisan *Java API Framework* berisi *library* Java yang digunakan sebagai pendukung sistem operasi dan pada lapisan sistem aplikasi berisi aplikasi aplikasi yang dapat digunakan secara langsung oleh pengguna (Google Android Developer, 2021). Gambar dari arsitektur Android dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Aplikasi Android

Sumber: Google Android Developer (2021)

2.6 Android Studio

Platform resmi dalam mengembangkan perangkat lunak bergerak berbasis Android adalah Android Studio. Android Studio adalah sebuah perangkat lunak yang digunakan untuk mengembangkan perangkat lunak yang didasarkan pada IntelliJ IDEA. Android Studio memiliki banyak fitur yang dapat digunakan dalam meningkatkan produktivitas dalam mengembangkan aplikasi berbasis Android (Google Android Developer, 2021).

2.7 Firebase

Firebase merupakan produk dari google berupa layanan *cloud* dan *backend* yang digunakan untuk mempermudah pengembangan perangkat lunak berbasis Android, IOS, dan Website (Stonehem,2016).

Terdapat 2 segi layanan dari Firebase yaitu *Spark* dan *Blaze*. *Spark* merupakan layanan secara gratis dengan layanan terbatas dan *Blaze* merupakan layanan yang biayanya tergantung dari jumlah pemakaian layanan. Firebase memiliki beberapa fasilitas antara lain *Authentication*, *Cloud Firestore*, *Cloud Storage*, *Cloud Messaging*, *Realtime Database* dan *Firebase Hosting* (Juliarto, 2020).

Ketika akan menerapkan fitur - fitur Firebase diperlukan *library* Firebase *Bill of Materials (BoM)* untuk mengelola versi dari fitur Firebase yang akan dipakai. Tabel 2.1 merupakan *library* dari Firebase BoM.

Tabel 2.1 Library Firebase BoM

```
dependencies{
    implementation platform('com.google.firebaseio:firebase-bom:28.3.0')
```

2.7.1 Firebase Authentication

Firebase Authentication merupakan layanan backend untuk mengautentikasi pengguna ke dalam perangkat lunak dan dapat terintegrasi dengan layanan Firebase lainnya. Pada *Firebase Authentication* terdapat autentikasi dengan nomor telepon, sandi dan autentikasi dengan penyedia identitas gabungan seperti Google, Facebook, Twitter dan lain-lain (Firebase, 2021). Tabel 2.2 merupakan *library* dari *Firebase Authentication*.

Tabel 2.2 Library Firebase Authentication

```
dependencies{
    implementation 'com.google.firebaseio:firebase-auth-ktx'
```

2.7.2 Firebase Cloud Firestore

Firebase Cloud Firestore merupakan *database cloud* NoSQL yang digunakan dalam pengembangan aplikasi perangkat bergerak, *web*, *server* untuk menyimpan data dari sisi klien dan *server*. *Firebase Cloud Firestore* dapat terintegrasi dengan layanan Firebase dan Google Cloud (Firebase, 2021). Tabel 2.3 merupakan *library* dari *Firebase Cloud Firestore*.

Tabel 2.3 Library Firebase Cloud Firestore

```
dependencies{
    implementation 'com.google.firebaseio:firebase-firebase-ktx'
```

2.7.3 Firebase Cloud Storage

Firebase Cloud Storage adalah layanan Firebase yang digunakan untuk menyimpan objek berupa gambar, audio, video dan konten lainnya. *Firebase Cloud Storage* digunakan untuk mengunggah dan mengunduh objek dari sisi klien yang dapat digunakan pada saat kualitas baik maupun buruk (Firebase, 2021). Tabel 2.4 merupakan *library* dari *Firebase Cloud Storage*.

Tabel 2.4 Library Firebase Cloud Storage

```
dependencies{
    implementation 'com.google.firebaseio:firebase-storage-ktx'
```

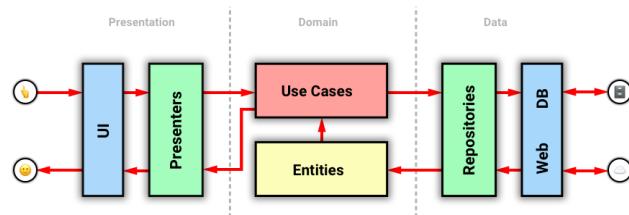
2.8 Arsitektur Pengembangan Aplikasi Android

Pada pengembangan perangkat lunak terdapat *architectural patterns* yang digunakan sebagai pengelolaan terhadap keseluruhan komponen yang ada dalam

perangkat lunak. Beberapa *architectural patterns* yang umum ada pada pengembangan Android adalah MVx *pattern* yaitu Model View Controller, Model View Presenter, dan Model View ViewModel. Setiap arsitektur tersebut memiliki kelemahan dan kelebihannya masing-masing baik dalam tahap pengembangan maupun pengujian (Sondha et al., 2020)

Dalam mengembangkan perangkat lunak yang memiliki sifat *maintainable, independent, testable*, dan *robust* salah satu *architectural patterns* yang dapat digunakan adalah *clean architecture*. *Clean architecture* merupakan *architectural patterns* yang bertindak sebagai abstraksi lebih tinggi daripada MVx *pattern*, *clean architecture* lebih berfokus pada pemisahan *layer*. Dengan memisahkan perangkat lunak menjadi beberapa *layer* dan sesuai dengan *dependency rule* akan membuat perangkat lunak menjadi lebih mudah diuji dan *maintainable* (Martin, 2017).

Pada penelitian ini pengembangan perangkat lunak dilakukan dengan menggunakan *clean architecture* dengan membagi menjadi 3 bagian yaitu presentation, domain, dan data. *Presentation* akan berisi *Presenter* dan *UI*, *Domain* akan berisi *Use Case* dan *Entities*, *Data* akan berisi *Repository* dan *Data Source*. Gambaran dari alur *data flow clean architecture* Android dapat dilihat pada Gambar 2.2.



Gambar 2.2 Data Flow Clean Architecture Android

Sumber: Jeevuz (2017)

2.9 Pengembangan Perangkat Lunak

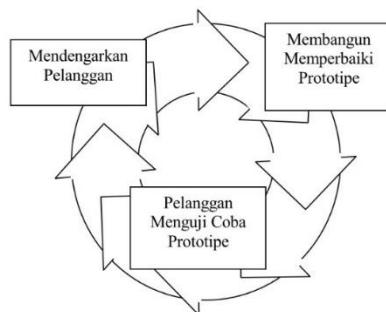
2.9.1 Software Development Life Cycle (SDLC)

Pengembangan perangkat lunak yang baik memerlukan sebuah *Software Development Life Cycle* untuk menghindari adanya *software crisis* atau pengembangan perangkat lunak yang buruk dan dapat menyebabkan kegagalan dalam proyek perangkat lunak tersebut. *Software Development Life Cycles* adalah metode dalam pembangunan perangkat lunak yang berisi analisis, desain, *testing* dan perawatan sistem (Dwanoko, 2016).

2.9.2 Prototype

Prototype merupakan *software development life cycle* yang digunakan untuk mengetahui lebih dalam spesifikasi dari kebutuhan perangkat lunak kepada pengguna dengan cara memberikan gambaran *prototype* beserta alur penggunaan perangkat lunak kepada pengguna (Ariani S. & Shalahuddin, 2018). Alur dari

software development life cycle prototype dimulai dengan menggali kebutuhan perangkat lunak kepada pelanggan, dilanjutkan dengan membuat *prototype* yang memiliki simulasi alur perangkat lunak dan *prototype* dievaluasi oleh pengguna sampai menemukan spesifikasi kebutuhan yang diinginkan oleh pengguna (Ariani S. & Shalahuddin, 2018). Ilustrasi dari model *prototype* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi Model Prototype

Sumber: Ariani S. & Shalahuddin (2018)

2.10 Pengujian Perangkat Lunak

Pengembangan perangkat lunak memiliki sebuah pengujian untuk menilai tingkat dari keberhasilan pengembangan perangkat lunak. Pada penelitian ini memiliki dua jenis pengujian yaitu pengujian validasi dan pengujian *usability*.

2.10.1 Validasi

Validasi adalah pengujian untuk memastikan aktifitas atau hasil produk yang berada pada perangkat lunak sudah sesuai dengan kebutuhan dari pengguna. Pengujian validasi dapat menggunakan *black box testing* sebagai pendekatan dalam melakukan pengujian. Pada *black box testing* pengujian dilakukan dengan membuat kasus uji pada setiap fungsi yang ada untuk memastikan semua alur dari perangkat lunak sudah selaras dengan kebutuhan dari pengguna (Ariani S. & Shalahuddin, 2018).

2.10.2 Usability

Usability adalah pengujian untuk mengetahui permasalahan dari *usability* dari perangkat lunak, mengumpulkan data kuantitatif dan kualitatif serta kepuasan pengguna dengan perangkat lunak (Farouqi, Aknuranda & Herlambang, 2018). Pengujian *usability* dapat menggunakan *System Usability Scale* dalam melakukan pengujian. *System Usability Scale* adalah pengujian yang dilakukan dengan memberikan kuesioner untuk mengukur tingkat *usability* dari sistem berdasarkan subjektif pengguna (H.N, Nugroho & Ferdiana, 2015).

Daftar pertanyaan dari *System Usability Scale* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Daftar Pertanyaan System Usability Scale

No	Pertanyaan
----	------------

1	Saya berpikir akan menggunakan aplikasi ini lagi
2	Saya merasa aplikasi ini rumit untuk digunakan
3	Saya merasa aplikasi ini mudah digunakan
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan aplikasi ini
5	Saya merasa fitur-fitur aplikasi ini berjalan dengan semestinya
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada aplikasi ini)
7	Saya merasa orang lain akan memahami cara menggunakan aplikasi ini dengan cepat
8	Saya merasa sistem ini membingungkan
9	Saya merasa tidak ada hambatan dalam menggunakan aplikasi ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan aplikasi ini

Sumber: Adaptasi dari Hasibuan et al., (2020)

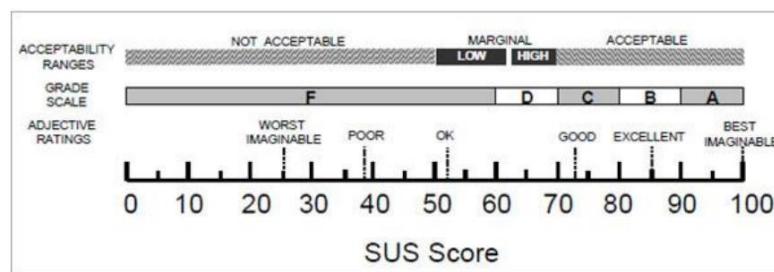
Skor dari jawaban pertanyaan *System Usability Scale* dapat dilihat pada Tabel 2.6.

Tabel 2.6 Skor Pertanyaan *System Usability Scale*

Jawaban	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Sumber: Adaptasi dari Hasibuan et al.(2020)

Representasi penilaian dari *System Usability Scale* dapat dilihat pada Gambar 2.4.



Gambar 2.4 System Usability Score Grade

Sumber: Bangor et al.(2018)

BAB 3 METODOLOGI

Bab metodologi berisi tentang uraian metode yang digunakan pada penelitian implementatif. Bab ini terdiri dari strategi penelitian, subjek penelitian, lokasi penelitian, metode pengumpulan data, peralatan pendukung, dan metode pengembangan

3.1 Strategi Penelitian

Strategi penelitian dalam penyusunan laporan tugas akhir adalah dengan studi literatur. Studi literatur bertujuan untuk mencari informasi terkait permasalahan penelitian dengan menelusuri sumber-sumber yang pernah dibuat sebelumnya melalui penelitian sebelumnya, jurnal ilmiah, buku yang berhubungan dengan permasalahan penelitian maupun landasan dasar yang dapat digunakan sebagai dasar penelitian. Adapun teori yang berkaitan sebagai berikut:

1. Kajian Pustaka
2. *Event*
3. Teknologi Informasi
4. Analisis Kebutuhan Perangkat Lunak
 - a. Kebutuhan Fungsional
 - b. Kebutuhan Non Fungsional
5. Aplikasi Perangkat Bergerak
6. Android
7. Android Studio
8. Firebase
 - a. *Firebase Authentication*
 - b. *Firebase Cloud Firestore*
 - c. *Firebase Cloud Storage*
9. Arsitektur Pengembangan Perangkat Lunak
10. Pengembangan Perangkat Lunak
 - a. *Software Development Life Cycle (SDLC)*
 - b. *Prototyping*
11. Pengujian Perangkat Lunak
 - a. Validasi
 - b. *Usability*

3.2 Subjek Penelitian

Subjek penelitian untuk pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi berbasis Android adalah masyarakat yang aktif mengikuti *event* di bidang teknologi informasi dan salah satu penyelenggara *event* di bidang teknologi informasi yaitu ketua pelaksana Hology 3.0.

3.3 Lokasi Penelitian

Lokasi penelitian pada penelitian ini dilakukan di Kota Madiun dan Kota.

3.4 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian "Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi *Event* di bidang Teknologi Informasi berbasis Android" adalah kuesioner dan wawancara. Pengumpulan data kuesioner dilakukan kepada 103 masyarakat yang pernah mengikuti *event* di bidang teknologi informasi dan wawancara dilakukan kepada salah satu penyelenggara *event* di bidang teknologi informasi yaitu ketua pelaksana dari Hology 3.0.

Daftar pertanyaan dari kuesioner dapat dilihat pada Tabel 3.1

Tabel 3.1 Daftar Pertanyaan Kuesioner

No	Pertanyaan
1	Nama Anda?
2	Usia Anda?
3	Apa pekerjaan Anda?
4	Seberapa sering anda mengikuti <i>Event</i> di bidang teknologi? (Seminar/Workshop/Lomba di bidang IT)
5	Darimana anda mendapatkan informasi <i>Event</i> IT yang ingin anda ikuti?
6	Pernahkah anda kesulitan mendapatkan informasi <i>Event</i> IT dengan topik yang ingin anda ikuti?
7	Pernahkan anda melewatkkan <i>Event</i> IT yang ingin anda datangi sebelumnya karena terlupa?
8	Apabila ada aplikasi untuk memberikan informasi <i>Event</i> IT (Workshop/Seminar/Lomba) berdasarkan kategori kategori yang terdapat fitur Calender Daftar <i>Event</i> yang anda inginkan dan Reminder ketika sudah mendekati <i>Event</i> , akan berguna bagi anda ?
9	Apakah Ada Permasalahan lainnya yang anda alami ketika hendak mengikuti <i>Event</i> IT?

Daftar pertanyaan dari wawancara dapat dilihat pada Tabel 3.2

Tabel 3.2 Daftar Pertanyaan Wawancara

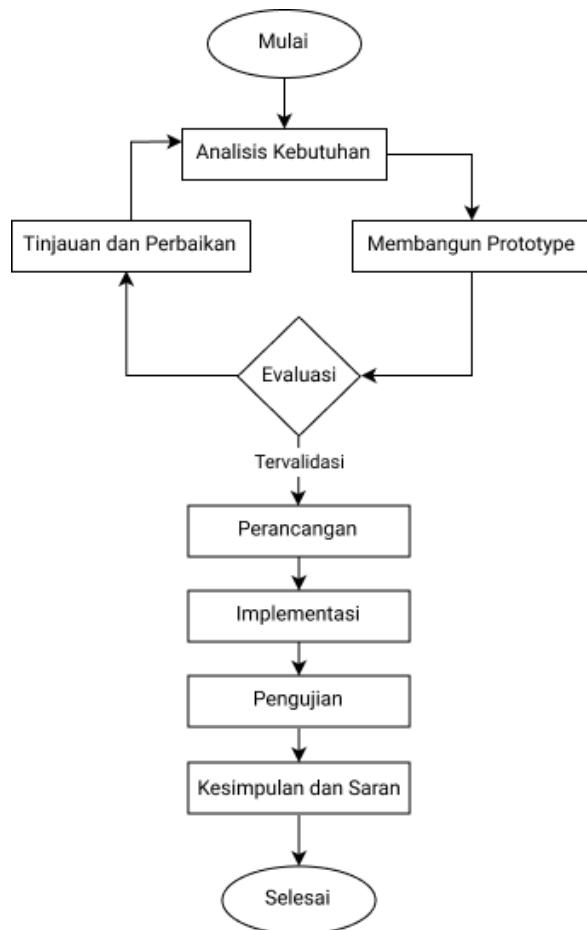
No	Pertanyaan
1	Boleh memperkenalkan diri dan peran dalam <i>event</i> dibidang IT?
2	<i>Event</i> yang ada di Hology ada apa saja?
3	Apakah ada kendala dalam penyebaran Informasi tentang <i>Event</i> yang anda adakan?
4	Bagaimana cara anda menyebarkan informasi <i>Event</i> tersebut?
5	Permasalahan yang dihadapi selama sebelum acara dan saat acara?
6	Apakah terdapat peserta yang mendaftar namun tidak mendatangi <i>eventnya</i> ?
7	Menurut anda selaku orang yang pernah menyelenggarakan <i>Event</i> IT Apabila ada aplikasi untuk memberikan informasi <i>Event</i> IT (Workshop/Seminar/Lomba) berdasarkan kategori kategori yang terdapat fitur Calender Daftar <i>Event</i> yang anda inginkan dan Reminder ketika sudah mendekati <i>Event</i> , akan berguna bagi peserta dan juga penyelenggara?
8	Apakah ada kritik dan saran untuk pengembangan aplikasi saya?

3.5 Peralatan Pendukung

Peralatan pendukung yang digunakan dalam penelitian ini terbagi menjadi dua peralatan pendukung yaitu perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam penelitian adalah laptop dan perangkat bergerak berbasis Android. Perangkat lunak yang digunakan dalam penelitian ini adalah Android Studio sebagai peralatan pengembangan perangkat lunak dan Figma sebagai peralatan untuk membuat desain antarmuka dari perangkat lunak.

3.6 Metode Pengembangan

Metode pengembangan perangkat lunak pada penelitian ini adalah *software development life cycle prototyping*. *Software development life cycle prototyping* digunakan pada penelitian ini dikarenakan lebih fleksibel dalam pengembangan dan kebutuhan perangkat lunak dapat ditambah ataupun dikurangi sesuai dengan kebutuhan pengguna dalam pengembangan perangkat lunak. Gambar dari alur metode pengembangan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Metode Pengembangan

3.6.1 Analisis Kebutuhan

Pada tahap analisis kebutuhan, analisis kebutuhan dilakukan dengan melakukan survei kepada 103 masyarakat yang aktif mengikuti *event* dibidang teknologi informasi dan wawancara kepada salah satu penyelenggara *event* dibidang teknologi informasi yaitu ketua pelaksana Hology 3.0. Tahap ini bertujuan untuk mengetahui kebutuhan yang dibutuhkan dalam perancangan maupun pengembangan perangkat lunak.

Informasi yang didapat dari survei dan wawancara akan dianalisis untuk spesifikasi kebutuhan dari perangkat lunak. Hasil dari analisis akan digunakan untuk kebutuhan yang dibutuhkan dalam tahap perancangan baik kebutuhan fungsional maupun kebutuhan non fungsional yang akan diuraikan ke dalam bentuk *use case diagram* dan *use case scenario*.

3.6.2 Membangun Prototype

Pada tahap membangun *prototype* peneliti membangun gambaran dari aplikasi berupa *prototype* yang akan digunakan oleh pengguna melalui tampilan desain antarmuka *high fidelity* dengan menggunakan perangkat lunak Figma. Tahap ini bertujuan untuk membuat gambaran dari perangkat lunak dan alur dari

penggunaan perangkat lunak kepada pengguna untuk mendapatkan *feedback* dan saran atas perangkat lunak yang dikembangkan.

3.6.3 Evaluasi

Berdasarkan *prototype* yang telah dibuat pada tahap membangun *prototype*, *prototype* diberikan kepada lima masyarakat yang memiliki minat pada bidang teknologi informasi dan juga kepada salah satu penyelenggara *event* dibidang teknologi informasi untuk mendapatkan *feedback* berupa kritik dan saran yang akan digunakan untuk mendapatkan hasil evaluasi. Hasil dari evaluasi akan digunakan sebagai pertimbangan untuk melakukan iterasi untuk pembaruan kebutuhan atau tetap melanjutkan ke tahap selanjutnya.

Iterasi pada pengembangan perangkat lunak dibatasi dengan maksimal tiga kali iterasi dikarenakan keterbatasan waktu yang ada dan apabila pada iterasi ke dua sudah sesuai dengan kebutuhan pengguna maka akan langsung dilanjutkan pada tahap selanjutnya.

3.6.4 Tinjauan dan Perbaikan

Feedback yang didapatkan dari tahap evaluasi sebelumnya dipergunakan oleh peneliti untuk ditinjau dan dilakukan perbaikan terhadap kebutuhan dari perangkat lunak yang dikembangkan. Tidak semua *feedback* akan dipergunakan untuk perbaikan dan pembaruan, *feedback* yang dirasa perlu akan dipergunakan sebagai acuan perbaikan dan penambahan dari penggalian kebutuhan.

3.6.5 Perancangan

Pada tahap perancangan, perancangan menggunakan metode pendekatan *object oriented*. Metode *object oriented* melakukan pendekatan dengan menggambarkan sistem sebagai objek - objek. Hasil dari evaluasi yang telah tervalidasi akan digunakan sebagai panduan dalam melakukan perancangan. Tahap perancangan berisi beberapa perancangan antara lain perancangan arsitektur perangkat lunak, perancangan *sequence diagram*, perancangan *class diagram*, perancangan *basis data* dan *pseudocode*.

3.6.6 Implementasi

Setelah perancangan pada tahap sebelumnya telah selesai. Hasil dari perancangan digunakan sebagai acuan dalam mengimplementasikan kode program menjadi perangkat lunak menjadi sebuah aplikasi Android dengan menggunakan kotlin sebagai bahasa pemrograman dan Android Studio sebagai *Integrated Development Environment* (IDE) pengembangan perangkat lunak. Selain itu, hasil dari *prototype* yang sudah tervalidasi digunakan sebagai acuan dalam implementasi antarmuka pada perangkat lunak.

3.6.7 Pengujian

Perangkat lunak yang telah selesai di implementasikan akan diuji pada tahap pengujian. Pengujian pada penelitian ini dilakukan dengan 3 jenis pengujian yaitu

pengujian validasi, *usability* dan *compatibility*. Pengujian bertujuan untuk mengetahui tingkat kelayakan dan ketepatan fungsi dari perangkat lunak.

3.6.8 Kesimpulan dan Saran

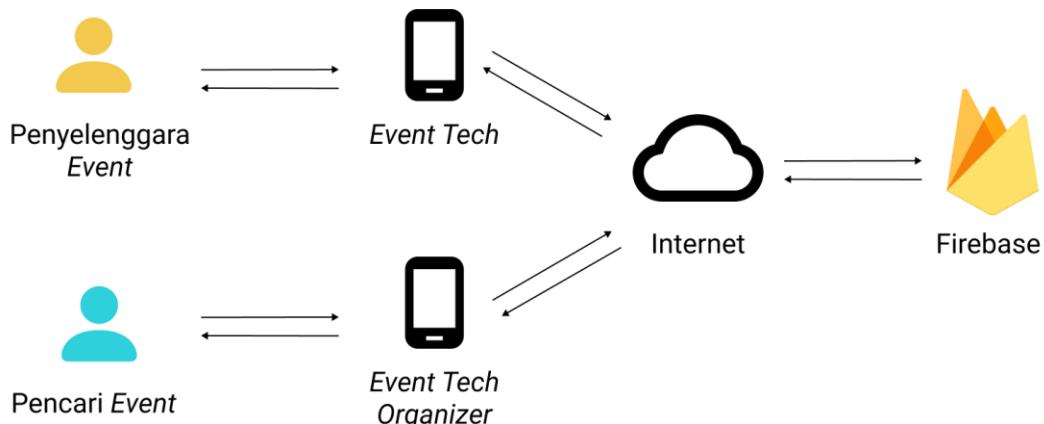
Setelah pengujian pada tahap sebelumnya telah selesai. Hasil dari tahap pengujian sebelumnya akan dijadikan sebagai acuan untuk penarikan kesimpulan pada tahap ini. Selain itu, saran juga dituliskan pada tahap ini untuk peneliti lain yang ingin mengembangkan penelitian ini.

BAB 4 ANALISIS KEBUTUHAN

Bab analisis kebutuhan berisi tentang hasil analisis kebutuhan. Bab ini memuat penjabaran gambaran umum sistem, identifikasi pengguna, kebutuhan fungsional, kebutuhan non fungsional, *use case* diagram.

4.1 Gambaran Umum Sistem

Pada penelitian ini sistem yang dikembangkan adalah aplikasi informasi *event* di bidang teknologi informasi berbasis Android. Aplikasi pada penelitian ini bernama *Event Tech*, *Event Tech* menghubungkan penyelenggara *event* di bidang teknologi dengan pencari *event* yang merupakan masyarakat yang memiliki minat pada bidang teknologi informasi. Penyelenggara *event* dapat mempublikasikan *event* yang mereka adakan kepada pengguna aplikasi ini. Masyarakat yang menggunakan aplikasi ini dapat mencari *event* di bidang teknologi informasi sesuai kategori yang diminati dan *event* tersebut merupakan *event* yang telah dipublikasikan oleh penyelenggara *event*. Aplikasi ini menggunakan Firebase *Firestore* untuk mengunggah data dan mengunduh data. Gambaran umum dari sistem dapat dilihat pada Gambar 4.1.



Gambar 4.1 Gambaran Umum Sistem

4.2 Identifikasi Pengguna

Pada aplikasi informasi *event* di bidang teknologi informasi ini terdapat pengguna yang akan menjalankan sistem. Terdapat dua jenis pengguna dalam sistem ini, jenis pengguna dan deskripsi tentang pengguna tersebut diuraikan pada Tabel 4.1.

Tabel 4.1 Identifikasi Pengguna

Jenis Pengguna	Deskripsi
Tamu	Tamu merupakan pengguna yang belum memasuki sistem. Tamu dapat melakukan

	login dan registrasi untuk bisa menggunakan fitur yang tersedia di aplikasi.
Penyelenggara <i>Event</i>	Penyelenggara <i>Event</i> merupakan pengguna yang telah terdaftar dalam sistem sebagai penyelenggara <i>event</i> . Penyelenggara <i>event</i> dapat mempublikasikan <i>event</i> , mengelola <i>event</i> yang dipublikasikannya serta melihat jumlah orang yang memfavoritkan <i>eventnya</i> .
Pencari <i>Event</i>	Pencari <i>Event</i> merupakan pengguna yang telah terdaftar dalam sistem sebagai pencari <i>event</i> . Pencari <i>event</i> dapat melihat daftar <i>event</i> dibidang teknologi informasi sesuai kategori yang di inginkan, menambahkan pengingat <i>event</i> dan mendaftar <i>event</i> .

4.3 Prototype Iterasi Pertama

Dalam membuat *prototype* pada iterasi pertama peneliti menggunakan hasil analisis kebutuhan yang didapatkan dari survei dan wawancara kepada pengguna. Pada tahap iterasi pertama dirancang kebutuhan fungsional, kebutuhan non fungsional, *use case* diagram dan *prototype* dari aplikasi dalam bentuk tampilan antarmuka *high-fidelity*.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi kebutuhan yang dibutuhkan oleh pengguna pada sistem dan dapat dikendalikan oleh pengguna. Setiap jenis pengguna memiliki kebutuhan fungsionalnya masing-masing sesuai dengan Tabel 4.1 terdapat tiga jenis pengguna yaitu tamu, penyelenggara *event*, dan pencari *event*.

4.3.1.1 Tamu

Jenis pengguna tamu sesuai dengan Tabel 4.1 merupakan pengguna yang belum memasuki sistem. Jenis pengguna tamu dibagi menjadi dua, yaitu tamu dari sisi penyelenggara *event* dan tamu dari sisi pencari *event*. Kebutuhan fungsional untuk tamu penyelenggara *event* tertera pada Tabel 4.2 dan tamu pencari *event* pada Tabel 4.3.

Tabel 4.2 Kebutuhan Fungsional Tamu Penyelenggara *Event*

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	ETO-F-001	Registrasi	Tamu dapat melakukan pendaftaran pada sistem sebagai penyelenggara <i>event</i> .

2.	ETO-F-002	<i>Login</i>	Tamu dapat masuk ke dalam sistem sebagai penyelenggara <i>event</i> .
----	-----------	--------------	---

Tabel 4.3 Kebutuhan Fungsional Tamu Pencari *Event*

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	ET-F-001	Registrasi	Tamu dapat melakukan pendaftaran pada sistem sebagai pencari <i>event</i> .
2.	ET-F-002	<i>Login</i>	Tamu dapat masuk ke dalam sistem sebagai pencari <i>event</i> .

4.3.1.2 Penyelenggara *Event*

Jenis pengguna penyelenggara *event* sesuai dengan Tabel 4.1 merupakan pengguna yang telah terdaftar dalam sistem sebagai penyelenggara *event*. Kebutuhan fungsional untuk penyelenggara *event* tertera pada Tabel 4.4

Tabel 4.4 Kebutuhan Fungsional Penyelenggara *Event*

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	ETO-F-003	Membuat <i>event</i>	Penyelenggara <i>event</i> dapat membuat <i>event</i> yang akan dipublikasikan.
2.	ETO-F-004	Menyunting <i>event</i>	Penyelenggara <i>event</i> dapat mengubah keterangan yang terdapat dalam <i>event</i> yang telah di tambahkan oleh penyelenggara <i>event</i> .
3.	ETO-F-005	Menghapus <i>event</i>	Penyelenggara <i>event</i> dapat menghapus <i>event</i> yang telah di tambahkan oleh penyelenggara <i>event</i> .
4.	ETO-F-006	Melihat <i>insight</i> dari <i>event</i>	Penyelenggara <i>event</i> dapat melihat <i>insight</i> dari <i>event</i> yang telah di tambahkan berupa jumlah <i>event</i> dilihat dan jumlah halaman pendaftaran dibuka.
5.	ETO-F-007	Melihat daftar <i>event</i>	Penyelenggara <i>event</i> dapat melihat daftar <i>event</i> dari <i>event</i> yang telah di tambahkan oleh penyelenggara <i>event</i> .

6.	ETO-F-008	Mengubah nama penyelenggara	Penyelenggara <i>event</i> dapat mengubah nama akun dari akun penyelenggara <i>event</i> .
7.	ETO-F-009	Mengubah <i>password</i>	Penyelenggara <i>event</i> dapat mengubah <i>password</i> dari akun penyelenggara.
8.	ETO-F-010	<i>Logout</i>	Penyelenggara <i>event</i> dapat keluar dari sistem.

4.3.1.3 Pencari Event

Jenis pengguna pencari *event* sesuai dengan Tabel 4.1 merupakan pengguna yang telah terdaftar dalam sistem sebagai pencari *event*. Kebutuhan fungsional untuk pencari *event* tertera pada Tabel 4.5

Tabel 4.5 Kebutuhan Fungsional Pencari Event

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	ET-F-003	Melihat daftar <i>event conference</i>	Pencari <i>event</i> dapat melihat <i>event conference</i> berdasarkan kategori kategori yang tersedia di sistem, dan terdapat menu untuk penyortiran.
2.	ET-F-004	Melihat daftar <i>event kompetisi IT</i>	Pencari <i>event</i> dapat melihat <i>event conference</i> berdasarkan kategori kategori yang tersedia di sistem, dan terdapat menu untuk penyortiran.
3.	ET-F-005	Melihat detail <i>event conference</i>	Pencari <i>event</i> dapat melihat detail informasi <i>event conference</i> pada halaman detail <i>event conference</i> .
4.	ET-F-006	Melihat detail <i>event kompetisi IT</i>	Pencari <i>event</i> dapat melihat detail informasi <i>event</i> pada halaman detail <i>event kompetisi IT</i> .
5.	ET-F-007	Menambahkan notifikasi pengingat	Pencari <i>event</i> dapat menambahkan notifikasi pengingat pada <i>event</i> yang telah ditambahkan pengingat.
6.	ET-F-008	Menghapus notifikasi pengingat	Pencari <i>event</i> dapat menghapus notifikasi pengingat pada <i>event</i> yang telah ditambahkan pengingat.

7.	ET-F-009	Mendaftar <i>event</i>	Pencari <i>event</i> dapat mendaftar <i>event</i> pada halaman detail <i>event</i> .
8.	ET-F-010	Melihat jadwal <i>event</i>	Pencari <i>event</i> dapat melihat jadwal <i>event</i> yang telah ditambahkan pengingat pada halaman detail <i>event</i> sebelumnya.
9.	ET-F-011	Menambahkan favorit	Pencari <i>event</i> dapat menambahkan favorit pada <i>event</i> pada halaman detail <i>event</i> .
10.	ET-F-012	Melihat favorit <i>event</i>	Pencari <i>event</i> dapat melihat daftar <i>event</i> yang telah ditambahkan favorit pada halaman detail <i>event</i> sebelumnya.
11.	ET-F-013	Menghapus favorit <i>event</i>	Pencari <i>event</i> dapat menghapus favorit <i>event</i>
12.	ET-F-014	Melihat daftar favorit <i>event</i>	Pencari <i>event</i> dapat melihat daftar favorit <i>event</i>
13.	ET-F-015	Mengubah nama pengguna	Pencari <i>event</i> dapat mengubah nama pengguna.
14.	ET-F-016	Mengubah <i>password</i>	Pencari <i>event</i> dapat mengubah <i>password</i> dari akun pencari <i>event</i> .
15.	ET-F-017	<i>Logout</i>	Pencari <i>event</i> dapat keluar dari sistem.

4.3.2 Kebutuhan Non Fungsional

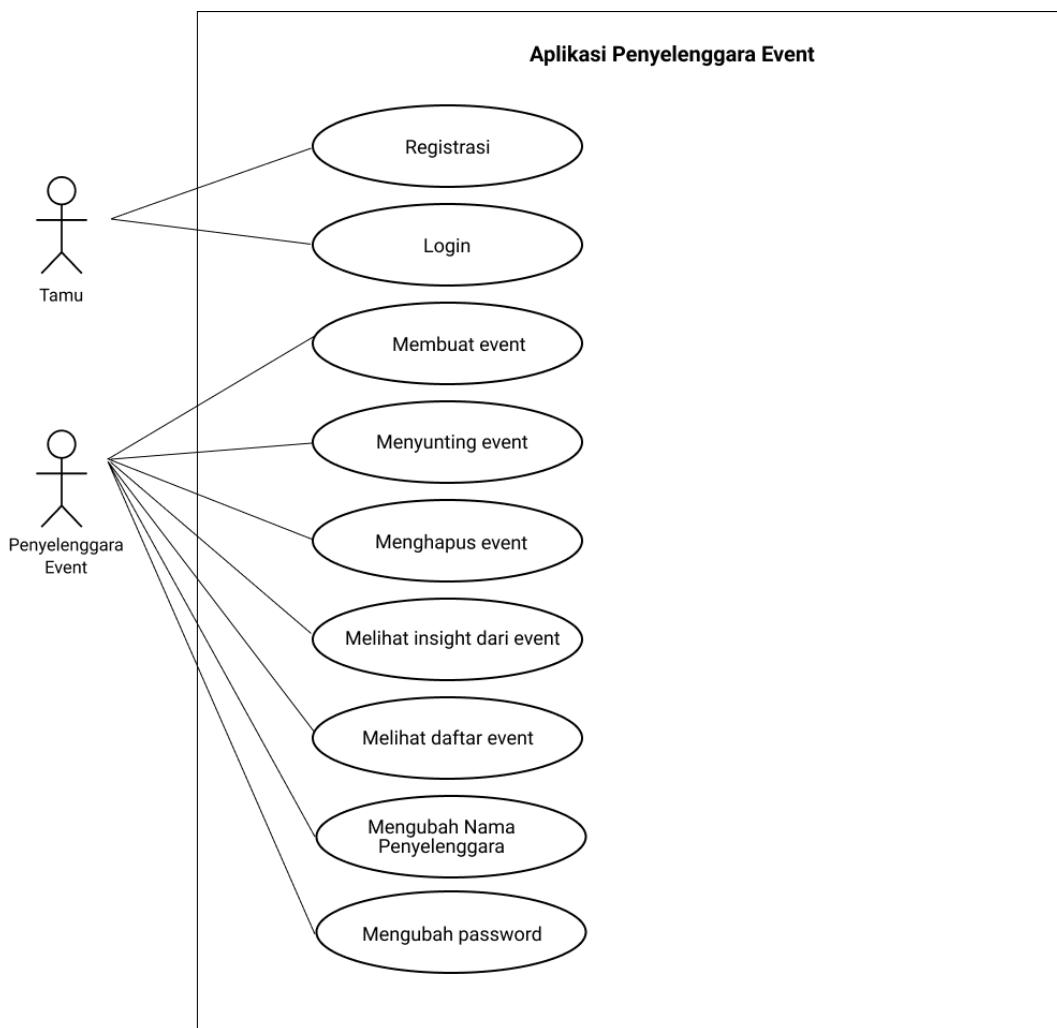
Kebutuhan non fungsional berisi tentang karakteristik sistem kedepannya dan batasan dari sistem. Kebutuhan non fungsional tertera pada Tabel 4.1

Tabel 4.6 Kebutuhan Non Fungsional

No	Kode Fungsi	Parameter	Deskripsi
1.	ET-NF-001	<i>Compatibility</i>	Sistem dapat berjalan pada sistem operasi Android SDK 23 sampai SDK 30.
2.	ET-NF-002	<i>Usability</i>	Sistem dapat memenuhi target pengujian <i>system usability scale</i> dengan rata rata <i>score</i> minimal 70.

4.3.3 Use Case Diagram

Use case diagram digunakan untuk memberikan gambaran dari sistem yang dapat dirasakan oleh pengguna. Pada pengembangan perangkat lunak ini terdapat dua *use case diagram* yaitu *use case diagram* aplikasi penyelenggara *event* pada Gambar 4.2 dan *use case diagram* aplikasi pencari *event* pada Gambar 4.3.



Gambar 4.2 Use Case Diagram Aplikasi Penyelenggara Event

Pada Gambar 4.2 *use case diagram* pada aplikasi penyelenggara *event* terdiri dari dua aktor yaitu tamu dan penyelenggara *event*. Aktor tamu dapat melakukan fungsional *login* dan registrasi dan aktor penyelenggara *event* memiliki beberapa fungsionalitas, yaitu membuat *event*, menyunting *event*, menghapus *event*, melihat *insight* dari *event*, melihat daftar *event*, mengubah nama penyelenggara, mengubah *password*.



Gambar 4.3 Use Case Diagram Aplikasi Pencari Event

Pada Gambar 4.3 *use case* diagram pada aplikasi pencari *event* terdiri dari dua aktor yaitu tamu dan pencari *event*. Aktor tamu memiliki *use case* registrasi dan *login*. Aktor pencari *event* memiliki beberapa *use case* antara lain melihat daftar *event conference*, melihat daftar *event* kompetisi IT, melihat detail *event conference*, melihat detail *event* kompetisi IT, menambahkan notifikasi pengingat, menghapus notifikasi pengingat, mendaftar *event*, melihat jadwal *event*, menambahkan favorit *event*, menghapus favorit *event*, melihat favorit *event*, mengubah nama pengguna, mengubah *password* dan *logout*.

4.3.4 Use Case Scenario

Use case scenario pada iterasi pertama merupakan penjabaran lebih lanjut dari *use case diagram* pada sub bab sebelumnya. Pada *use case skenario* dijelaskan tentang objektif, aktor, prasyarat, skenario utama, scenario alternatif, dan kondisi sesudah *use case* dijalankan.

Tabel 4.7 Use Case Scenario Registrasi Penyelenggara Event

Registrasi	
Objektif	Aktor dapat mendaftar ke dalam sistem
Aktor	Tamu
Prasyarat	Aktor telah berada pada halaman registrasi
Skenario utama	<ol style="list-style-type: none"> 1. Aktor mengisi nama penyelenggara, <i>email</i> dan <i>password</i> 2. Aktor menekan tombol daftar 3. Sistem memproses pendaftaran 4. Akun penyelenggara di tambahkan ke dalam <i>database</i> 5. Sistem menampilkan halaman beranda
Skenario alternatif	<ol style="list-style-type: none"> 1a. Ketika aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan maka akan terdapat pesan untuk mengisi data 3a. Apabila <i>email</i> yang didaftarkan telah terdaftar maka sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan
Kondisi sesudah	Aktor masuk ke dalam sistem menjadi penyelenggara <i>event</i>

Tabel 4.8 Use Case Scenario Login Penyelenggara Event

Login	
Objektif	Aktor dapat masuk ke dalam sistem
Aktor	Tamu
Prasyarat	Aktor telah berada pada halaman login
Skenario utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>email</i> dan <i>password</i> 2. Aktor menekan tombol masuk 3. Sistem melakukan validasi <i>email</i> dan <i>password</i> 4. Sistem menampilkan halaman beranda

Skenario alternatif	<p>1a. Ketika aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan maka akan terdapat pesan untuk mengisi data</p> <p>3a. Apabila <i>email</i> dan <i>password</i> tidak berhasil diverifikasi maka akan menampilkan pesan masuk gagal</p>
Kondisi sesudah	Aktor masuk ke dalam sistem sebagai penyelenggara <i>event</i>

Tabel 4.9 Use Case Scenario Registrasi Pencari Event

Registrasi	
Objektif	Aktor dapat mendaftar ke dalam sistem
Aktor	Tamu
Prasyarat	Aktor telah berada pada halaman registrasi
Skenario utama	<ol style="list-style-type: none"> 1. Aktor mengisi nama penyelenggara, <i>email</i> dan <i>password</i> 2. Aktor menekan tombol daftar 3. Sistem memproses pendaftaran 4. Akun penyelenggara di tambahkan ke dalam <i>database</i> 5. Sistem menampilkan halaman beranda
Skenario alternatif	<p>1a. Ketika aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan maka akan terdapat pesan untuk mengisi data</p> <p>3a. Apabila <i>email</i> yang didaftarkan telah terdaftar maka sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan</p>
Kondisi sesudah	Aktor masuk ke dalam sistem sebagai pencari <i>event</i>

Tabel 4.10 Use Case Scenario Scenario Login Pencari Event

<i>Login</i>	
Objektif	Aktor dapat masuk ke dalam sistem
Aktor	Tamu
Prasyarat	Aktor telah berada pada halaman login
Skenario utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>email</i> dan <i>password</i> 2. Aktor menekan tombol masuk

	3. Sistem melakukan validasi <i>email</i> dan <i>password</i> 4. Sistem menampilkan halaman beranda
Skenario alternatif	1a. Ketika aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan maka akan terdapat pesan untuk mengisi data 3a. Apabila <i>email</i> dan <i>password</i> tidak berhasil diverifikasi maka akan menampilkan pesan masuk gagal
Kondisi sesudah	Aktor masuk ke dalam sistem sebagai pencari <i>event</i>

Tabel 4.11 Use Case Membuat Event

Membuat Event	
Objektif	Aktor dapat membuat <i>event</i>
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor berada pada halaman tambahkan <i>event</i>
Skenario utama	1. Aktor memilih jenis <i>event</i> yang mau di tambahkan dengan menekan tombol tambah <i>event</i> 2. Aktor mengisi kelengkapan informasi dari <i>event</i> yang akan di tambahkan 3. Aktor menekan tombol publikasikan <i>event</i> 4. Sistem memproses penambahan <i>event</i> 5. Sistem menampilkan halaman tambah <i>event</i>
Skenario alternatif	1a. Ketika aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan maka akan terdapat pesan untuk mengisi data
Kondisi sesudah	Aktor berhasil membuat <i>event</i>

Tabel 4.12 Use Case Menyunting Event

Menyunting event	
Objektif	Aktor dapat menyunting <i>event</i> miliknya
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman beranda
Skenario utama	1. Aktor menekan tombol menu pada <i>item event</i> yang ingin di sunting 2. Aktor memilih opsi sunting

	<ul style="list-style-type: none"> 3. Aktor mengganti <i>field</i> data yang ingin diubah 4. Aktor menekan tombol perbarui perubahan 5. Sistem menampilkan pesan perubahan <i>event</i> berhasil 6. Sistem menampilkan halaman beranda
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil menyunting <i>event</i>

Tabel 4.13 Use Case Scenario Menghapus Event

Menghapus Event	
Objektif	Aktor dapat menghapus <i>event</i> miliknya
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman beranda
Skenario utama	<ul style="list-style-type: none"> 1. Aktor menekan tombol menu pada <i>item event</i> yang ingin dihapus 2. Aktor memilih opsi hapus 3. Sistem menampilkan dialog konfirmasi hapus <i>event</i> 4. Aktor memilih tombol hapus pada dialog yang muncul untuk konfirmasi
Skenario alternatif	4a. Aktor memilih tombol batalkan pada dialog yang muncul untuk konfirmasi
Kondisi sesudah	Aktor berhasil menghapus <i>event</i>

Tabel 4.14 Use Case Scenario Melihat Insight Dari Event

Melihat <i>insight</i> dari <i>event</i>	
Objektif	Aktor dapat melihat <i>insight</i> dari <i>event</i> yang telah ditambahkan sebelumnya
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman <i>event insight</i>
Skenario utama	<ul style="list-style-type: none"> 1. Sistem menampilkan <i>insight</i> dari semua <i>event</i> milik aktor 2. Aktor melihat <i>insight</i> dari semua <i>event</i> milik aktor
Skenario alternatif	-

Kondisi sesudah	Aktor berhasil melihat <i>insight</i> dari <i>event</i> milik aktor
-----------------	---

Tabel 4.15 Use Case Scenario Melihat Daftar Event

Melihat daftar event	
Objektif	Aktor dapat melihat daftar <i>event</i> yang telah dipublikasikannya
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman beranda
Skenario utama	<ol style="list-style-type: none"> 1. Sistem menampilkan semua <i>event</i> milik aktor 2. Aktor melihat semua <i>event</i> milik aktor
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat daftar <i>event</i> milik aktor

Tabel 4.16 Use Case Scenario Mengubah Nama Penyelenggara

Mengubah nama penyelenggara	
Objektif	Aktor dapat mengubah nama penyelenggara <i>event</i>
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman profil
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan ubah nama pengguna 2. Sistem menampilkan halaman ubah nama 3. Aktor mengisi nama pada <i>field</i> nama penyelenggara 4. Aktor menekan tombol simpan perubahan 5. Sistem menampilkan halaman Profil
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil mengubah nama penyelenggara

Tabel 4.17 Use Case Scenario Mengubah Password Penyelenggara Event

Mengubah password	
Objektif	Aktor dapat mengubah nama penyelenggara <i>event</i>
Aktor	Penyelenggara <i>event</i>
Prasyarat	Aktor telah berada pada halaman profil
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan ubah <i>password</i>

	<ol style="list-style-type: none"> 2. Sistem menampilkan halaman ubah <i>password</i> 3. Aktor mengisi <i>email</i> pada <i>field email</i> penyelenggara 4. Aktor menekan tombol kirim <i>reset password</i> 5. Sistem menampilkan pesan tautan perubahan berhasil dikirim melalui <i>email</i> 6. Sistem menampilkan halaman Profil
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil mendapatkan tautan <i>reset password</i> pada <i>email</i> miliknya

Tabel 4.18 Use Case Scenario Logout Penyelenggara Event

Logout	
Objektif	Aktor dapat <i>logout</i> dari sistem
Aktor	Penyelenggara event
Prasyarat	Aktor telah berada pada halaman profil
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan <i>logout</i> 2. Sistem menampilkan halaman <i>login</i>
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil <i>logout</i> dari sistem

Tabel 4.19 Use Case Scenario Melihat Daftar Event Conference

Melihat daftar event conference	
Objektif	Aktor dapat melihat daftar event conference
Aktor	Pencari event
Prasyarat	Aktor berada pada halaman daftar event conference
Skenario utama	<ol style="list-style-type: none"> 1. Aktor memilih kategori event 2. Sistem menampilkan daftar event sesuai dengan kategori yang dipilih 3. Aktor melihat daftar event conference sesuai dengan kategori yang dipilih
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat daftar event conference

Tabel 4.20 Use Case Scenario Melihat Event Kompetisi IT

Melihat daftar event kompetisi IT	
Objektif	Aktor dapat melihat daftar event kompetisi IT
Aktor	Pencari event
Prasyarat	Aktor berada pada halaman daftar event kompetisi IT
Skenario utama	<ol style="list-style-type: none">1. Aktor memilih kategori event2. Sistem menampilkan daftar event sesuai dengan kategori yang dipilih3. Aktor melihat daftar event kompetisi IT sesuai dengan kategori yang dipilih
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat daftar event kompetisi IT

Tabel 4.21 Use Case Scenario Melihat Detail Event Conference

Melihat detail event conference	
Objektif	Aktor dapat melihat detail event conference
Aktor	Pencari event
Prasyarat	Aktor berada pada halaman daftar event conference
Skenario utama	<ol style="list-style-type: none">1. Aktor memilih event dari daftar event yang tersedia2. Sistem menampilkan halaman detail event dan informasi detail event3. Aktor melihat detail event dan informasi didalamnya
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat detail event conference

Tabel 4.22 Use Case Scenario Melihat Detail Event Kompetisi IT

Melihat detail event kompetisi IT	
Objektif	Aktor dapat melihat detail event kompetisi IT
Aktor	Pencari event
Prasyarat	Aktor berada pada halaman daftar event kompetisi IT
Skenario utama	<ol style="list-style-type: none">1. Aktor memilih event dari daftar event yang tersedia2. Sistem menampilkan halaman detail dan informasi detail event

	3. Aktor melihat detail <i>event</i> dan informasi didalamnya
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat detail <i>event</i> kompetisi IT

Tabel 4.23 Use Case Scenario Menambahkan Notifikasi Pengingat

Menambahkan notifikasi pengingat	
Objektif	Aktor dapat menambahkan pengingat pada <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman detail <i>event</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor melihat detail <i>event</i> dan informasi didalamnya 2. Aktor menekan tombol tambahkan pengingat 3. Sistem menampilkan dialog konfirmasi untuk menambahkan pengingat 4. Aktor menekan tombol setuju pada dialog konfirmasi 5. Sistem menambahkan pengingat pada <i>event</i> yang telah di tambahkan pengingat oleh Aktor 6. Sistem menjadwalkan notifikasi pengingat pada <i>event</i> yang telah di tambahkan pengingat 7. Aktor mendapatkan notifikasi pengingat sesuai dengan waktu pengingat pada <i>event</i> yang telah di tambahkan pengingat
Skenario alternatif	4a. Aktor menekan tombol batalkan pada dialog konfirmasi
Kondisi sesudah	Aktor berhasil menambahkan pengingat pada <i>event</i>

Tabel 4.24 Use Case Scenario Menghapus Notifikasi Pengingat

Menambahkan notifikasi pengingat	
Objektif	Aktor dapat menambahkan pengingat pada <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman detail <i>event</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor melihat detail <i>event</i> dan informasi didalamnya 2. Aktor menekan tombol diingatkan

	<ul style="list-style-type: none"> 3. Sistem menampilkan dialog konfirmasi untuk menghapus pengingat 4. Aktor menekan tombol setuju pada dialog konfirmasi 5. Sistem menghapus pengingat pada <i>event</i> yang telah di tambahkan pengingat oleh Aktor 6. Sistem menghapus jadwal notifikasi pengingat pada <i>event</i> yang telah dihapus pengingat
Skenario alternatif	<ul style="list-style-type: none"> 5a. Aktor menekan tombol batalkan pada dialog konfirmasi dan <i>event</i> tidak jadi dihapus pengingatnya
Kondisi sesudah	Aktor berhasil menghapus pengingat pada <i>event</i>

Tabel 4.25 Use Case Scenario Mendaftar Event

Mendaftar event	
Objektif	Aktor dapat mendaftar pada <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman detail <i>event</i>
Skenario utama	<ul style="list-style-type: none"> 1. Aktor melihat detail <i>event</i> dan informasi didalamnya 2. Aktor menekan tombol daftar 3. Sistem menampilkan halaman <i>webview</i> yang berisi pendaftaran yang disediakan oleh penyelenggara <i>event</i> 4. Aktor mendaftar pada halaman <i>webview</i> yang berisi pendaftaran yang disediakan oleh penyelenggara <i>event</i>
Skenario alternatif	-
Kondisi sesudah	Aktor berada pada halaman <i>webview</i> yang berisi pendaftaran yang disediakan oleh penyelenggara <i>event</i>

Tabel 4.26 Use Case Scenario Melihat Jadwal Event

Melihat jadwal event	
Objektif	Aktor dapat melihat jadwal <i>event</i> dari <i>event</i> yang di tambahkan pengingat
Aktor	Pencari <i>event</i>

Prasyarat	Aktor telah berada pada halaman jadwal <i>event</i>
Skenario utama	<ol style="list-style-type: none"> 1. Sistem menampilkan semua <i>event</i> yang telah ditambahkan pengingat 2. Aktor melihat semua jadwal <i>event</i> beserta informasi waktu pelaksanaan dari <i>event</i> yang telah ditambahkan pengingat
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat jadwal <i>event</i> dari <i>event</i> yang ditambahkan pengingat

Tabel 4.27 Use Case Scenario Menambahkan Favorit

Menambahkan favorit	
Objektif	Aktor dapat menambahkan favorit pada <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman detail <i>event</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor melihat detail <i>event</i> dan informasi didalamnya 2. Aktor menekan tombol favorit 3. Sistem menambahkan favorit pada <i>event</i> yang difavoritkan
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil menambahkan favorit pada <i>event</i>

Tabel 4.28 Use Case Scenario Menghapus Favorit Event

Menghapus favorit <i>event</i>	
Objektif	Aktor dapat menghapus favorit pada <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah menambahkan favorit pada <i>event</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol favorit <i>event</i> 2. Sistem menghapus favorit pada <i>event</i> yang difavoritkan
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil menghapus favorit pada <i>event</i>

Tabel 4.29 Use Case Scenario Melihat Daftar Favorit Event

Melihat daftar favorit event	
Objektif	Aktor dapat melihat daftar <i>event</i> yang difavoritkan
Aktor	Pencari <i>event</i>
Prasyarat	Aktor berada pada halaman favorit <i>event</i>
Skenario utama	<ol style="list-style-type: none">1. Sistem menampilkan daftar <i>event</i> yang telah difavoritkan2. Aktor melihat daftar favorit <i>event</i>
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil melihat daftar favorit <i>event</i>

Tabel 4.30 Use Case Scenario Mengubah Nama Pengguna

Mengubah nama pengguna	
Objektif	Aktor dapat mengubah nama pengguna
Aktor	Pencari <i>event</i>
Prasyarat	Aktor berada pada halaman profil
Skenario utama	<ol style="list-style-type: none">1. Aktor menekan nama pengguna2. Sistem menampilkan halaman ubah nama pengguna3. Aktor mengisi nama pada <i>field</i> nama penyelenggara4. Aktor menekan tombol simpan perubahan5. Sistem menampilkan halaman Profil
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil mengubah nama pengguna

Tabel 4.31 Use Case Scenario Mengubah Password Pencari Event

Mengubah password	
Objektif	Aktor dapat mengubah password pencari <i>event</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman profil
Skenario utama	<ol style="list-style-type: none">1. Aktor menekan ubah password2. Sistem menampilkan halaman ubah password3. Aktor mengisi <i>email</i> pada <i>field</i> <i>email</i>

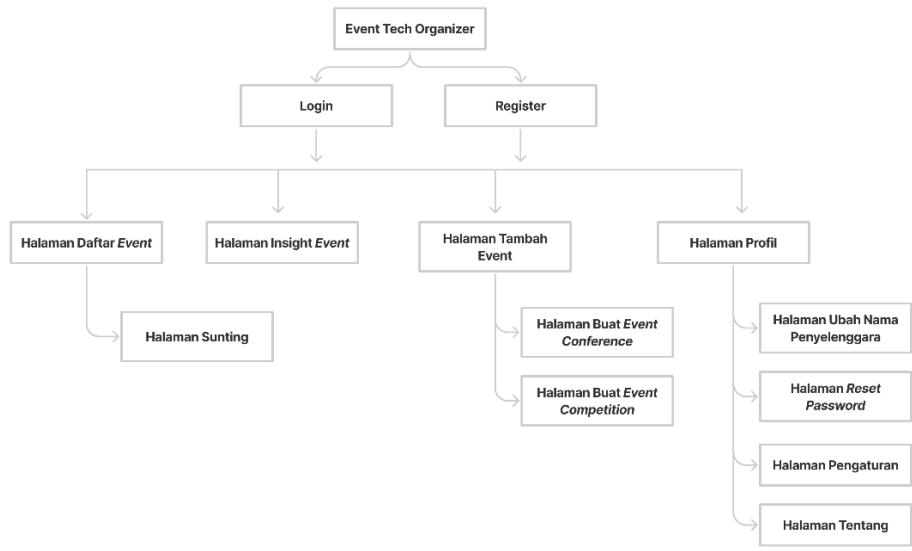
	4. Aktor menekan tombol kirim <i>reset password</i> 5. Sistem menampilkan pesan tautan perubahan berhasil dikirim melalui <i>email</i> 6. Sistem menampilkan halaman Profil
Skenario alternatif	-
Kondisi sesudah	Aktor berhasil mendapatkan tautan <i>reset password</i> pada <i>email</i> nya

Tabel 4.32 Use Case Scenario Logout Pencari Event

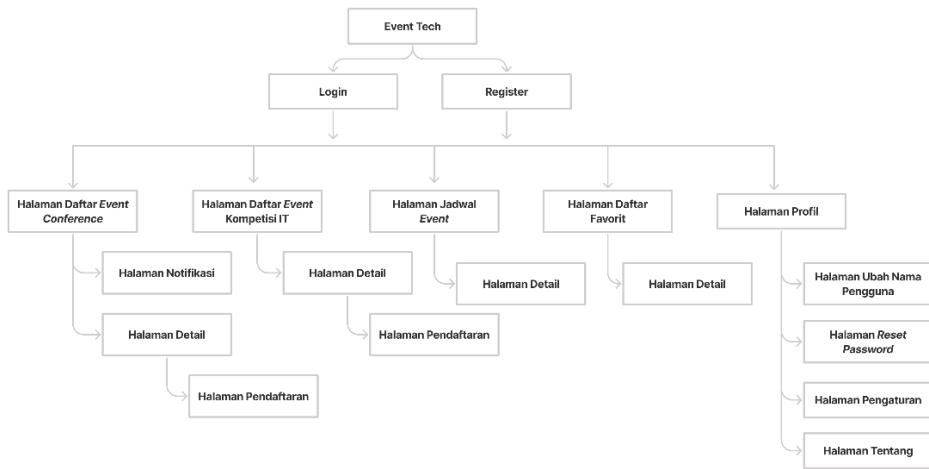
Logout	
Objektif	Aktor dapat <i>logout</i> dari sistem
Aktor	Pencari event
Prasyarat	Aktor telah berada pada halaman profil
Skenario utama	1. Aktor menekan <i>logout</i> 2. Aktor memilih tombol keluar pada dialog yang muncul untuk konfirmasi 3. Sistem menampilkan halaman <i>login</i>
Skenario alternatif	2. Aktor memilih tombol batalkan pada dialog yang muncul untuk konfirmasi
Kondisi sesudah	Aktor berhasil <i>logout</i> dari sistem

4.3.5 Mockup Aplikasi

Pada subbab ini akan dijabarkan *mockup* desain dalam bentuk *high-fidelity* yang di dapat dari hasil analisis sebelumnya. *Mockup* akan digunakan sebagai bahan evaluasi yang akan diajukan pada pengguna pada iterasi pertama *prototype*. *Information architecture* untuk aplikasi penyelenggara *event* dapat dilihat pada Gambar 4.4 dan *information architecture* untuk aplikasi pencari *event* dapat dilihat pada Gambar 4.5.



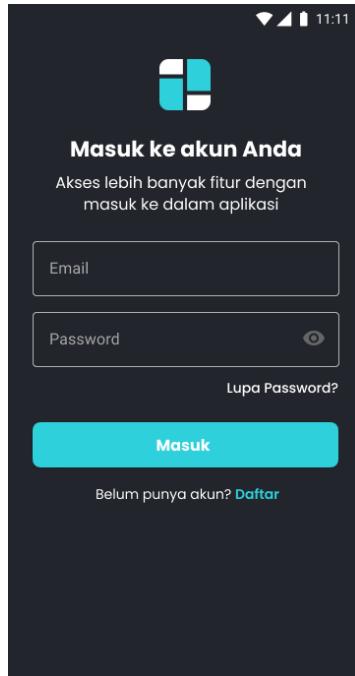
Gambar 4.4 Information Architecture Aplikasi Penyelenggara Event



Gambar 4.5 Information Architecture Aplikasi Pencari Event

4.3.5.1 Halaman Login Penyelenggara Event

Desain dalam bentuk *high-fidelity* untuk halaman *login* penyelenggara event tertera pada Gambar 4.6.

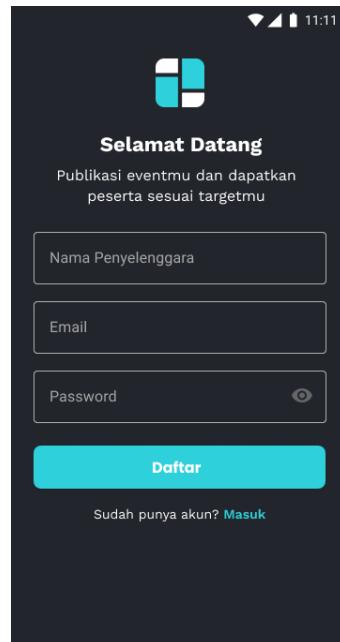


Gambar 4.6 High-Fidelity Halaman Login Penyelenggara Event

Pada halaman *login* terdapat kolom untuk mengisi *email* dan *password* yang akan diisi oleh pengguna untuk dapat divalidasi oleh sistem ketika pengguna menekan tombol masuk. Selain itu, terdapat pula sebuah teks “Daftar” untuk berpindah dari halaman *login* menuju ke halaman registrasi.

4.3.5.2 Halaman Registrasi Penyelenggara Event

Desain dalam bentuk *high-fidelity* untuk halaman registrasi penyelenggara *event* tertera pada Gambar 4.7.

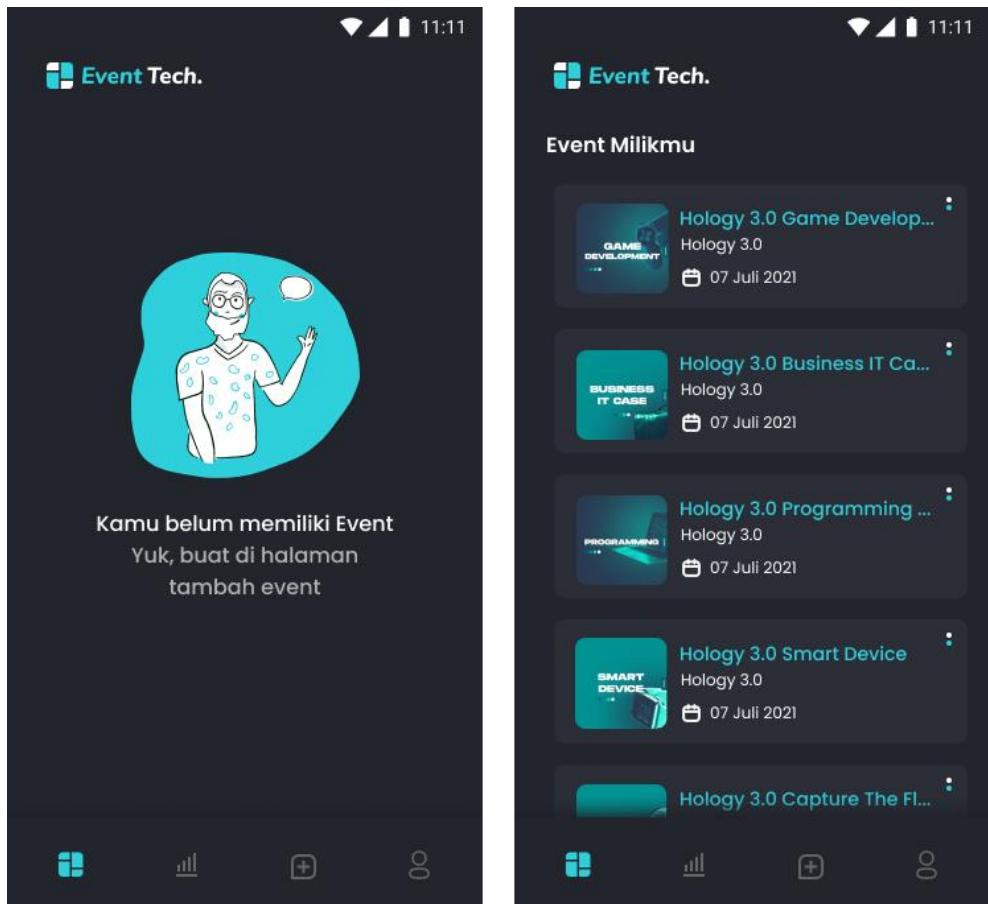


Gambar 4.7 High-Fidelity Halaman Registrasi Penyelenggara Event

Pada halaman registrasi penyelenggara *event harus mengisi* tiga kolom data yang digunakan untuk mendaftarkan akun penyelenggara yaitu, nama penyelenggara, *email*, dan *password*. Selain itu, terdapat tombol daftar yang akan memproses pendaftaran ketika tombol ditekan. Pengguna juga dapat berpindah halaman ke halaman *login* dengan menekan tulisan masuk.

4.3.5.3 Halaman Beranda Penyelenggara Event

Desain dalam bentuk *high-fidelity* untuk halaman beranda penyelenggara *event* tertera pada Gambar 4.8.

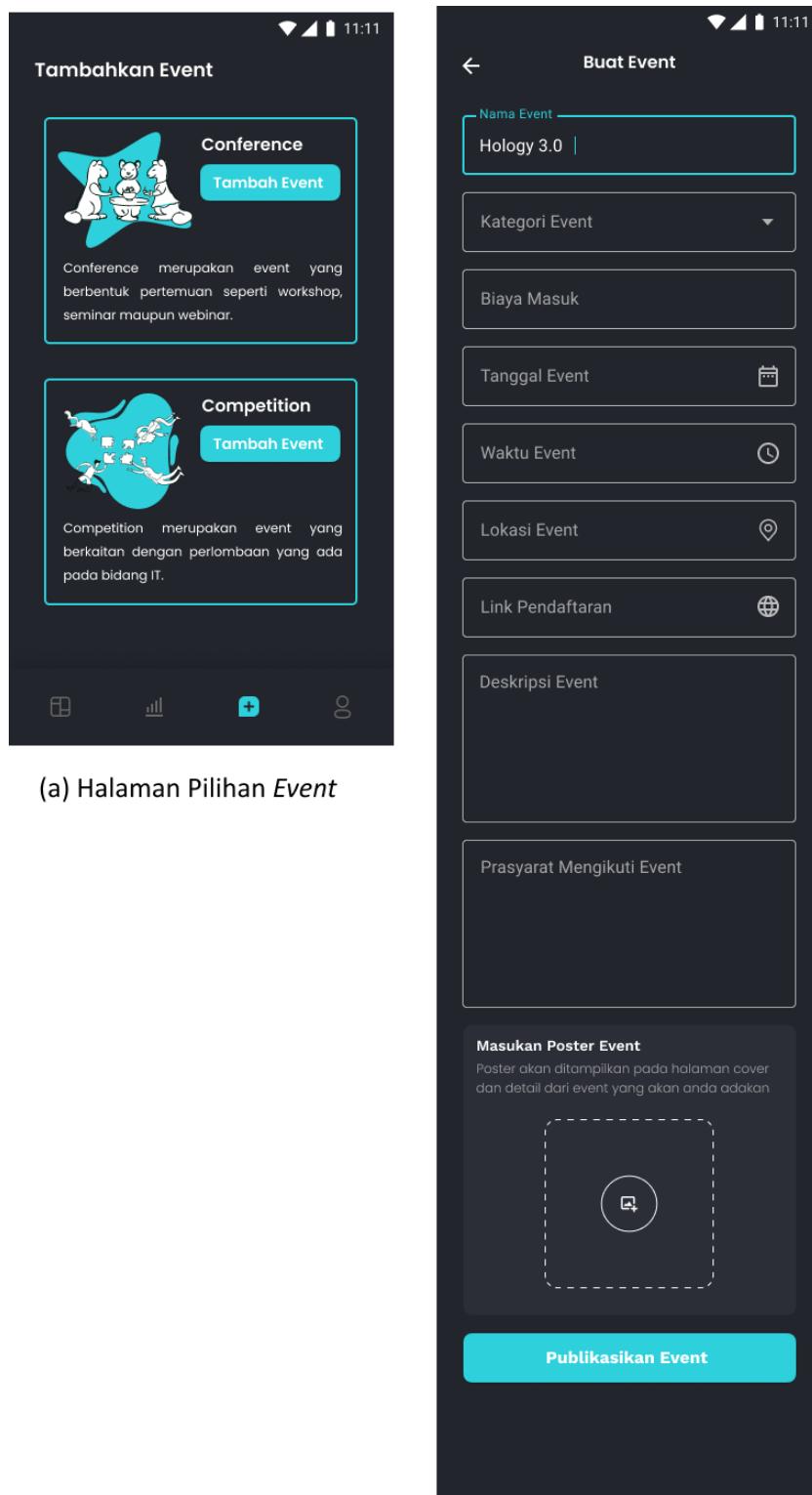


Gambar 4.8 High-Fidelity Halaman Beranda Penyelenggara Event

Pada halaman beranda apabila belum terdapat *event* yang dipublikasi oleh penyelenggara *event* akan ditampilkan pesan yang menjelaskan belum terdapat *event* yang dipublikasi. Ketika pengguna sudah memiliki *event* yang di tambahkan maka akan terdapat daftar *event* yang telah di tambahkan. Setiap *event* dari daftar *event* terdapat judul *event*, nama penyelenggara, dan tanggal pelaksanaan.

4.3.5.4 Halaman Tambahkan Event

Desain dalam bentuk *high-fidelity* untuk halaman tambahkan *event* tertera pada Gambar 4.9.



(b) Halaman Buat Event

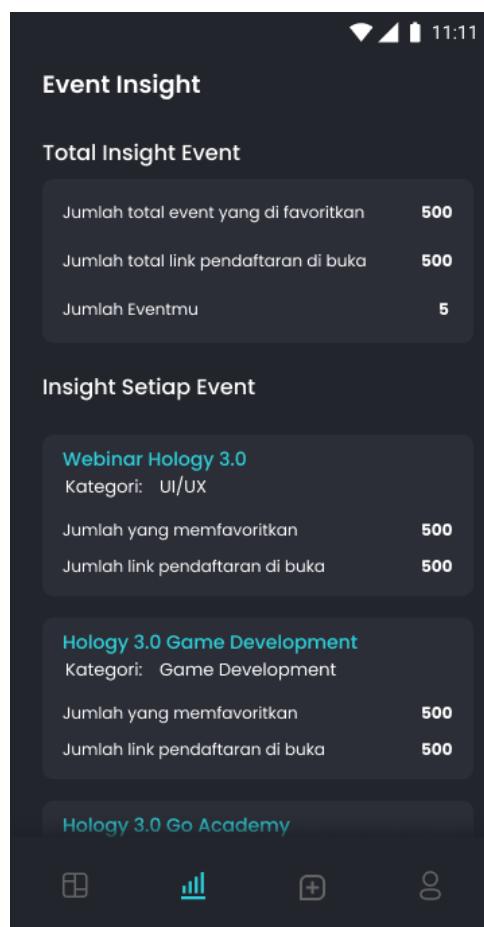
Gambar 4.9 *High-Fidelity Halaman Tambahkan Event*

Pada halaman tambahkan *event* terdapat dua bagian halaman, pada bagian a terdapat dua pilihan *event* yaitu *conference* yang berisi *event* yang berbentuk

pertemuan dan *competition* yang berisi *event* berbentuk kompetisi. Pada bagian b merupakan halaman untuk membuat *event*. Pada halaman buat *event* terdapat kolom informasi dan poster *event* yang harus dimasukan oleh penyelenggara *event* ketika ingin mempublikasikan *event*. Selain itu, terdapat tombol publikasikan *event* yang akan memproses penambahan *event*.

4.3.5.5 Halaman Insight Event

Desain dalam bentuk *high-fidelity* untuk halaman *insight event* penyelenggara *event* tertera pada Gambar 4.10.



Gambar 4.10 High-Fidelity Halaman *Insight Event*

Pada halaman *insight event* terdapat informasi mengenai total *insight* dari semua *event* yang telah di tambahkan oleh pengguna sebelumnya. Selain itu, terdapat *insight* dari setiap *event* yang berisikan jumlah yang memfavoritkan *event*, dan jumlah link pendaftaran di buka.

4.3.5.6 Halaman Sunting Event

Desain dalam bentuk *high-fidelity* untuk halaman sunting *event* tertera pada Gambar 4.11.

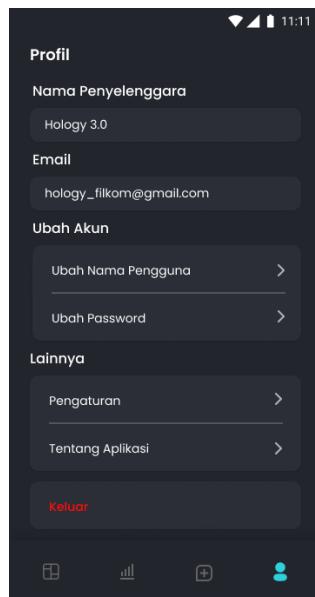


Gambar 4.11 High-Fidelity Halaman Sunting Event

Pada halaman sunting *event* terdapat kolom yang berisi data dari *event* yang ingin disunting. Pada halaman sunting *event* terdapat tombol perbarui perubahan untuk menyimpan perubahan. Selain itu terdapat pula tombol kembali untuk kembali ke halaman beranda.

4.3.5.7 Halaman Profil Penyelenggara Event

Desain dalam bentuk *high-fidelity* untuk halaman profil penyelenggara *event* tertera pada Gambar 4.12.

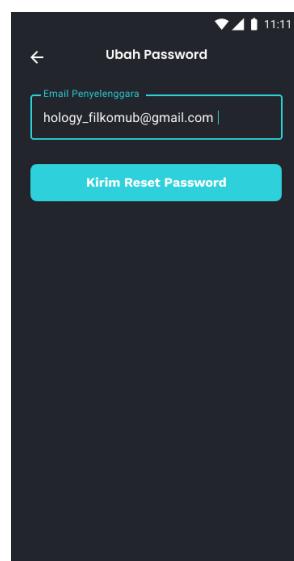


Gambar 4.12 High-Fidelity Halaman Profil Penyelenggara

Pada halaman profil penyelenggara terdapat keterangan nama dan *email* yang digunakan oleh pemilik akun, dan terdapat pula menu ubah nama pengguna, ubah *password*, pengaturan, tentang aplikasi dan keluar yang memiliki peran untuk berpindah halaman sesuai dengan fungsinya masing-masing.

4.3.5.8 Halaman Ubah Password Penyelenggara

Desain dalam bentuk *high-fidelity* untuk ubah *password* penyelenggara tertera pada Gambar 4.13.

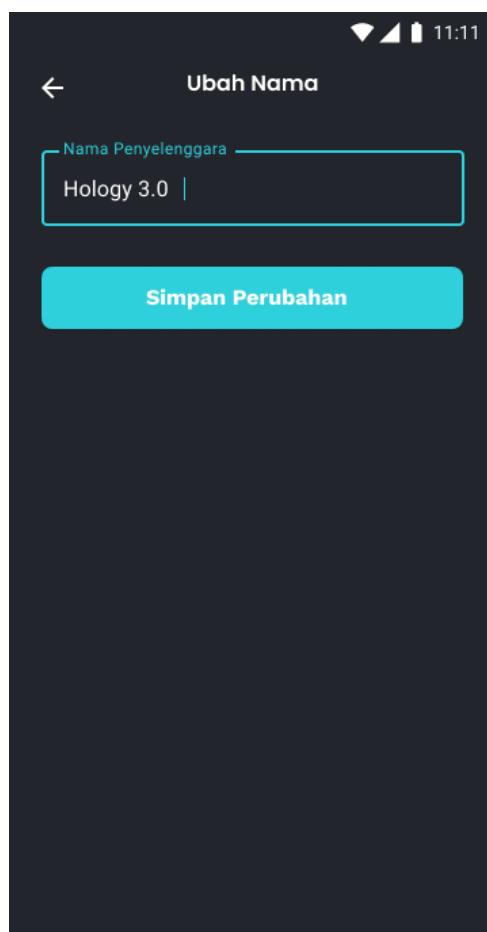


Gambar 4.13 High-Fidelity Halaman Ubah Password Penyelenggara

Pada halaman ubah *password* terdapat kolom untuk mengisi *email* penyelenggara. Pada halaman ubah *password* terdapat tombol kembali yang akan menuju ke halaman profil. Selain itu, juga terdapat tombol kirim *reset password* penyelenggara untuk mengirimkan tautan untuk memperbarui *password* penyelenggara.

4.3.5.9 Halaman Ubah Nama Penyelenggara

Desain dalam bentuk *high-fidelity* untuk ubah nama penyelenggara tertera pada Gambar 4.14.

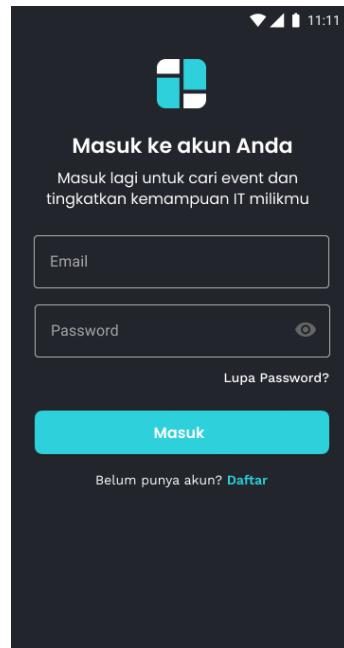


Gambar 4.14 High-Fidelity Halaman Ubah Nama Penyelenggara

Pada halaman ubah nama penyelenggara terdapat kolom untuk mengubah nama penyelenggara dengan nama yang baru. Pada halaman ubah nama penyelenggara terdapat tombol kembali yang akan menuju ke halaman profil. Selain itu, juga terdapat tombol simpan perubahan untuk menyimpan nama penyelenggara yang baru.

4.3.5.10 Halaman Login Pencari Event

Desain dalam bentuk *high-fidelity* untuk halaman login pencari *event* tertera pada Gambar 4.15.

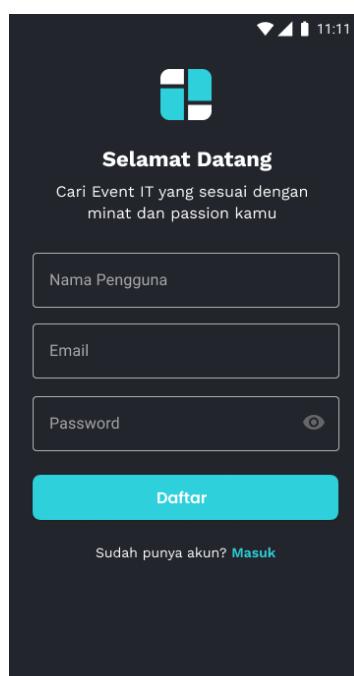


Gambar 4.15 High-Fidelity Halaman Login Pencari Event

Pada halaman *login* terdapat kolom untuk mengisi *email* dan *password* yang akan diisi oleh pengguna untuk dapat divalidasi oleh sistem ketika pengguna menekan tombol masuk. Selain itu, terdapat pula sebuah teks “Daftar” untuk berpindah dari halaman *login* menuju ke halaman registrasi.

4.3.5.11 Halaman Registrasi Pencari Event

Desain dalam bentuk *high-fidelity* untuk halaman registrasi pencari *event* tertera pada Gambar 4.16.

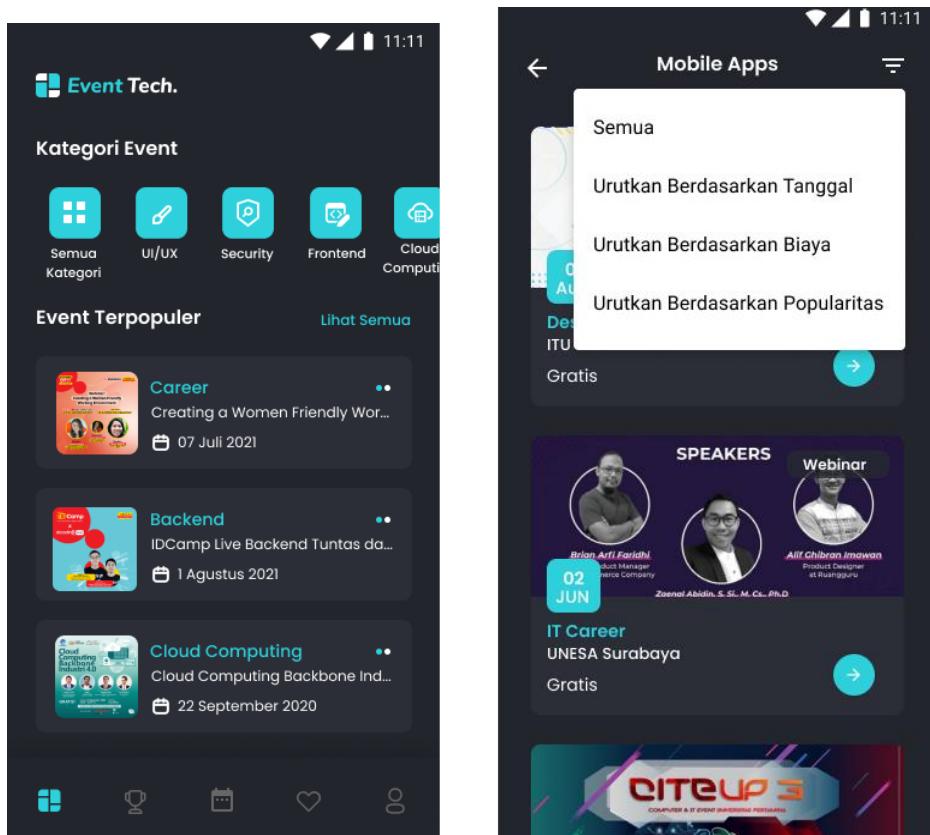


Gambar 4.16 High-Fidelity Halaman Registrasi Pencari Event

Pada halaman registrasi pencari *event harus mengisi* tiga kolom data yang digunakan untuk mendaftarkan akun pengguna yaitu, nama pengguna, *email*, dan *password*. Selain itu, terdapat tombol daftar yang akan memproses pendaftaran ketika tombol ditekan. Pengguna juga dapat berpindah halaman ke halaman *login* dengan menekan tulisan masuk.

4.3.5.12 Halaman Daftar Event Conference

Desain dalam bentuk *high-fidelity* untuk halaman daftar *event conference* tertera pada Gambar 4.17.



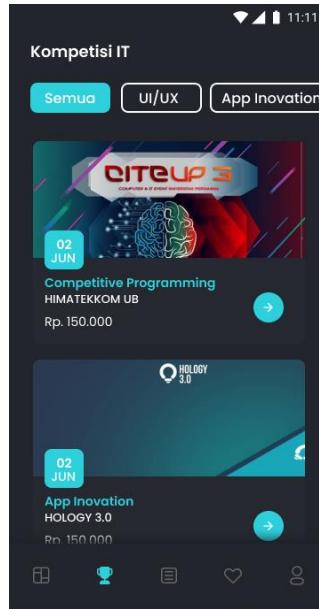
Gambar 4.17 High-Fidelity Halaman Daftar Event Conference

Pada halaman daftar *event conference* terdapat *list* kategori *event conference*, terdapat pula kumpulan *event conference* terpopuler. Selain itu, terdapat *icon* notifikasi yang berisi notifikasi ketika terdapat notifikasi pengingat.

Pada halaman *list event* berisi kumpulan *event* yang sesuai dengan kategori yang dipilih di halaman daftar *event conference*. Pada halaman ini terdapat sejumlah *event* dengan informasi tanggal pelaksanaan, nama penyelenggara dan biaya pendaftaran. Selain itu, terdapat pula menu yang digunakan untuk sorting *event-event* dan terdapat tombol kembali yang mengarah ke halaman daftar *event conference*.

4.3.5.13 Halaman Daftar Event Kompetisi IT

Desain dalam bentuk *high-fidelity* untuk halaman daftar *event* kompetisi IT tertera pada Gambar 4.18.

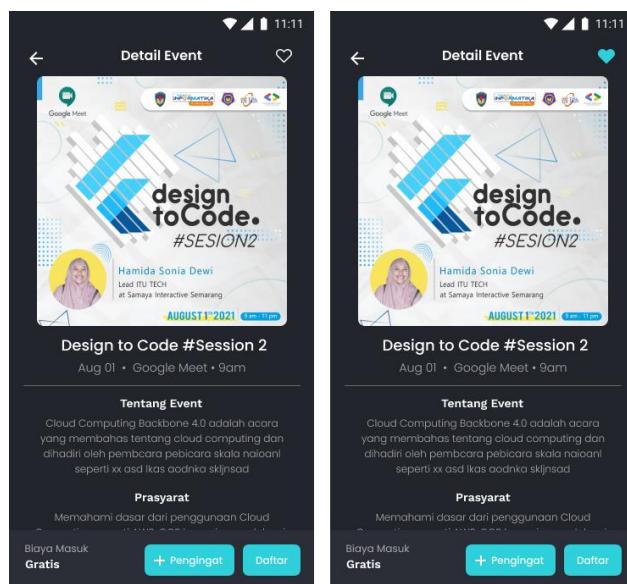


Gambar 4.18 High-Fidelity Halaman Daftar Event Kompetisi IT

Pada halaman daftar *event* kompetisi IT terdapat kategori-kategori dari *event* kompetisi IT dan terdapat daftar *event* yang sesuai dengan kategori yang sedang aktif, pada setiap *event* terdapat cover, waktu pelaksanaan, nama penyelenggara dan nominal biaya pendaftaran kompetisi.

4.3.5.14 Halaman Detail Event Conference

Desain dalam bentuk *high-fidelity* untuk halaman detail *event conference* tertera pada Gambar 4.19.

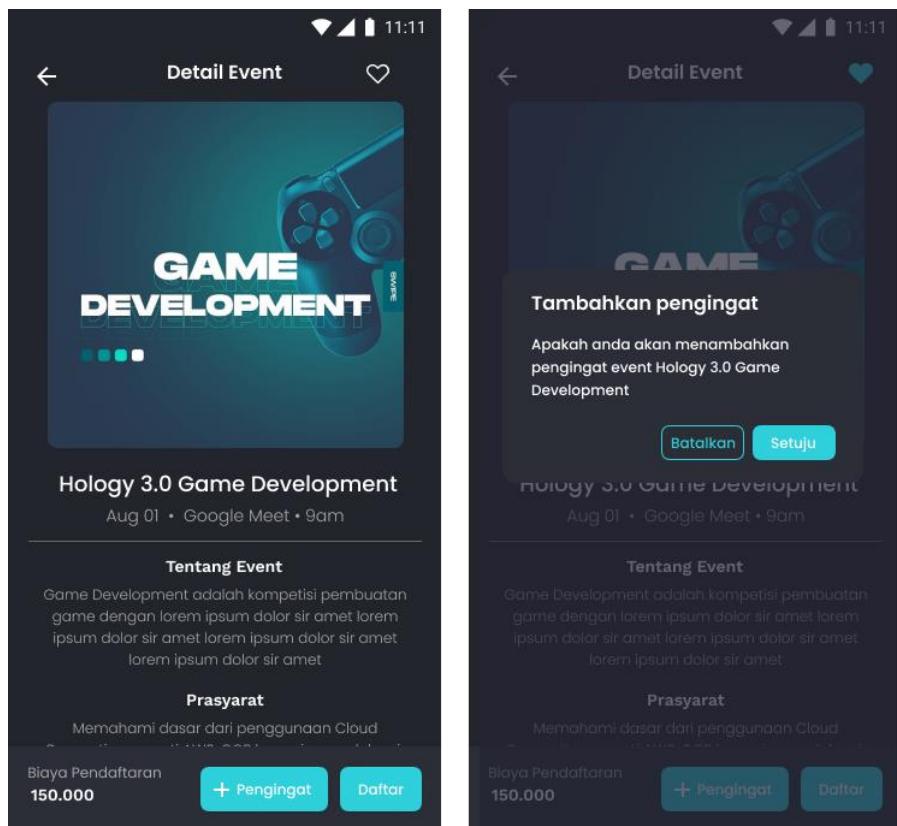


Gambar 4.19 High-Fidelity Halaman Detail Event

Pada halaman detail *event conference* terdapat informasi tentang *event* yang meliputi *cover gambar event*, jenis *event*, nama *event*, tentang *event*, prasyarat mengikuti *event*, informasi biaya masuk, tombol pengingat, tombol kembali, tombol favorit dan tombol pendaftaran.

4.3.5.15 Halaman Detail Event Kompetisi IT

Desain dalam bentuk *high-fidelity* untuk halaman detail *event* tertera pada Gambar 4.20.

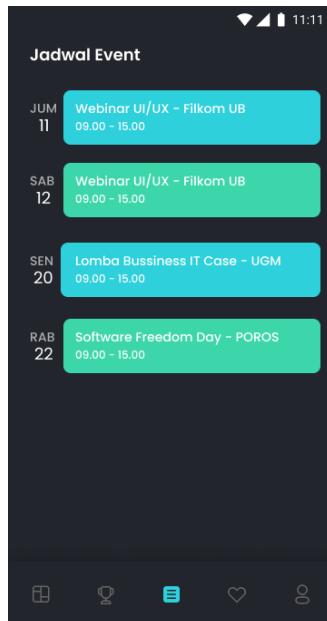


Gambar 4.20 High-Fidelity Halaman Detail Event

Pada halaman detail *event* terdapat informasi tentang *event* yang meliputi *cover gambar event*, nama *event*, tentang *event*, prasyarat mengikuti *event*, informasi biaya masuk, tombol pengingat, tombol kembali, tombol favorit dan tombol pendaftaran.

4.3.5.16 Halaman Jadwal Event

Desain dalam bentuk *high-fidelity* untuk halaman jadwal *event* tertera pada Gambar 4.21.

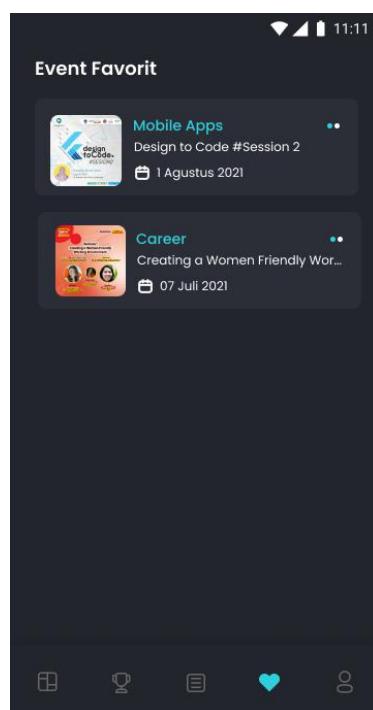


Gambar 4.21 High-Fidelity Halaman Jadwal Event

Pada halaman jadwal *event* terdiri dari kumpulan informasi dari *event* yang telah di tambahkan pengingat oleh pencari *event*. Setiap *event* yang tertera pada jadwal *event* memiliki informasi tanggal pelaksanaan, hari pelaksanaan, informasi waktu pelaksanaan, nama *event* dan nama penyelenggara *event*.

4.3.5.17 Halaman Event Favorit

Desain dalam bentuk *high-fidelity* untuk halaman *event* favorit tertera pada Gambar 4.22.

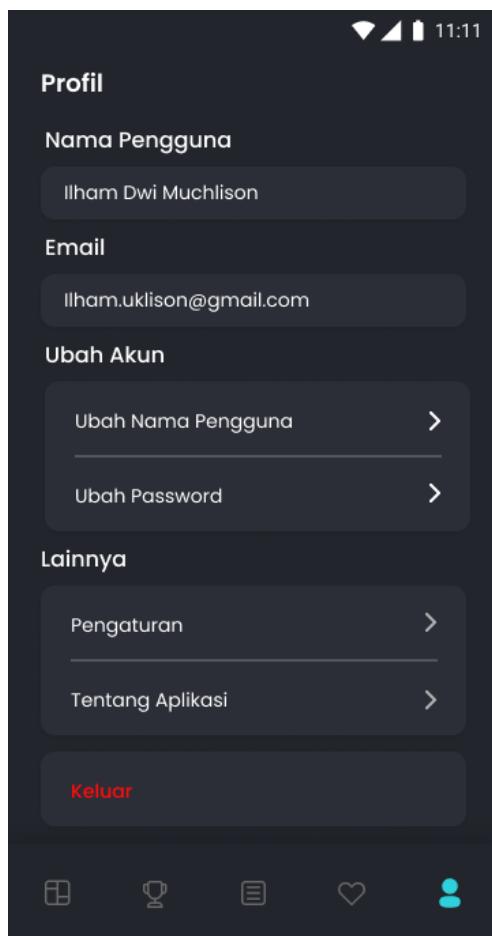


Gambar 4.22 High-Fidelity Halaman Event Favorit

Pada halaman *event* favorit terdapat kumpulan *event* yang telah ditambahkan favorit oleh pencari *event*. Setiap *event* yang terdapat di favorit memiliki informasi kategori *event*, nama *event* dan tanggal pelaksanaan *event*.

4.3.5.18 Halaman Profil Pencari Event

Desain dalam bentuk *high-fidelity* untuk halaman profil pencari *event* tertera pada Gambar 4.23.

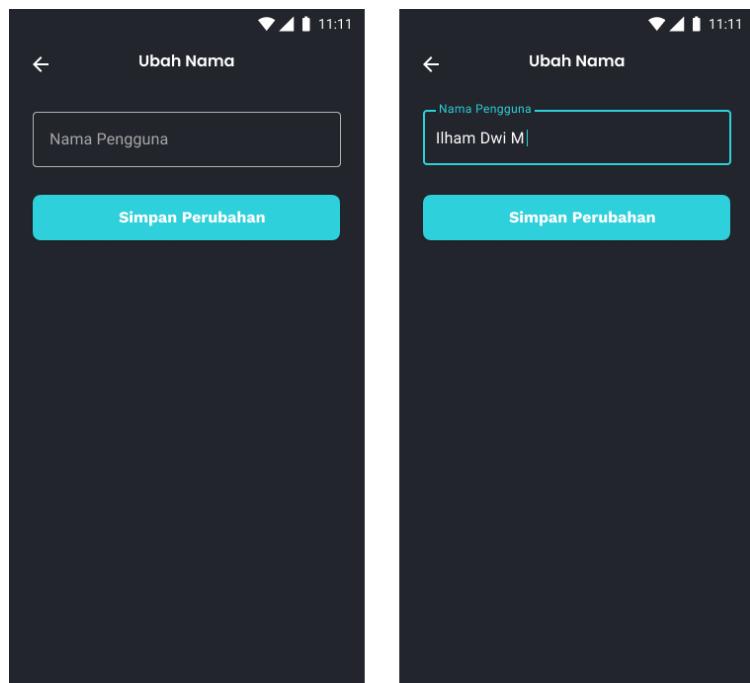


Gambar 4.23 High-Fidelity Halaman Profil Pencari Event

Pada halaman profil pencari *event* terdapat keterangan nama dan *email* yang digunakan oleh pemilik akun, dan terdapat pula menu ubah nama pengguna, ubah *password*, pengaturan, tentang aplikasi dan keluar yang memiliki peran untuk berpindah halaman sesuai dengan fungsinya masing-masing.

4.3.5.19 Halaman Ubah Nama Pengguna

Desain dalam bentuk *high-fidelity* untuk ubah nama pengguna tertera pada Gambar 4.24.

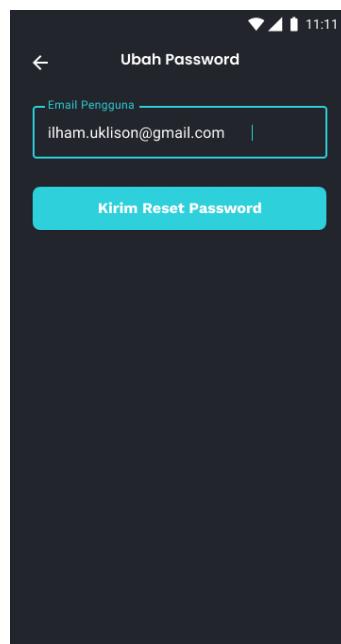


Gambar 4.24 High-Fidelity Halaman Ubah Nama Pengguna

Pada halaman ubah nama pengguna terdapat kolom untuk mengubah nama pengguna dengan nama yang baru. Pada halaman ubah nama pengguna terdapat tombol kembali yang akan menuju ke halaman profil. Selain itu, juga terdapat tombol simpan perubahan untuk menyimpan nama pengguna yang baru.

4.3.5.20 Halaman Ubah Password

Desain dalam bentuk *high-fidelity* untuk ubah *password* tertera pada Gambar 4.25.

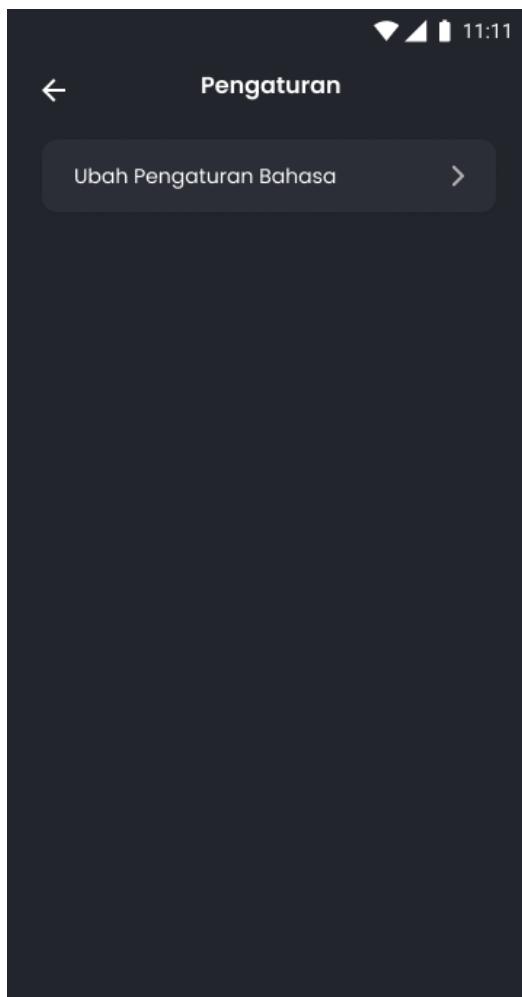


Gambar 4.25 High-Fidelity Halaman Ubah Password Pengguna

Pada halaman ubah *password* terdapat kolom untuk mengisi *email* pengguna. Pada halaman ubah nama *password* terdapat tombol kembali yang akan menuju ke halaman profil. Selain itu, juga terdapat tombol kirim *reset password* untuk mengirimkan tautan untuk memperbarui *password* pengguna.

4.3.5.21 Halaman Pengaturan

Desain dalam bentuk *high-fidelity* untuk halaman pengaturan tertera pada Gambar 4.26.



Gambar 4.26 High-Fidelity Halaman Pengaturan

Pada halaman pengaturan terdapat menu preferensi bahasa yang akan menyesuaikan bahasa yang digunakan sesuai dengan bahasa yang terdapat di perangkat Android milik pengguna.

4.3.5.22 Halaman Tentang

Desain dalam bentuk *high-fidelity* untuk halaman tentang tertera pada Gambar 4.27.



Gambar 4.27 High-Fidelity Halaman Tentang

Pada halaman tentang terdapat informasi dari aplikasi yang sedang digunakan dan fungsi pada aplikasi digunakan.

4.3.6 Evaluasi

Pada iterasi *prototype* tahap pertama setelah dilakukan evaluasi *prototype* kepada pengguna penyelenggara *event* dan pencari *event* terdapat beberapa hasil evaluasi yang telah ditinjau. Berdasarkan hasil dari evaluasi yang telah ditinjau akan dilanjutkan perbaikan pada iterasi *prototype* tahap kedua. Beberapa perbaikan yang dilakukan adalah penambahan fungsionalitas fitur pada aplikasi pencari *event* yaitu fitur melakukan pencarian *event conference* dan pencarian *event* kompetisi IT. Selain itu, terdapat pula perbaikan pada desain *mockup* yang akan dijelaskan lebih lanjut pada iterasi *prototype* tahap kedua.

4.4 Prototype Iterasi Kedua

Prototype pada iterasi kedua didasarkan pada hasil evaluasi kepada pengguna pada *prototype* iterasi pertama. Pada *prototype* iterasi kedua di tambahkan

beberapa fungsional dan dilakukan perubahan pada beberapa desain *mockup* yang dibuat pada iterasi pertama. Selain itu, terdapat juga perubahan pada *use case diagram* dan *use case scenario*.

4.4.1 Kebutuhan Fungsional

Pada *prototype* iterasi kedua terdapat beberapa perubahan kebutuhan fungsional sesuai dengan hasil evaluasi yang melalui tinjauan. Perubahan tersebut hanya terdapat pada aktor pencari *event*.

4.4.1.1 Pencari Event

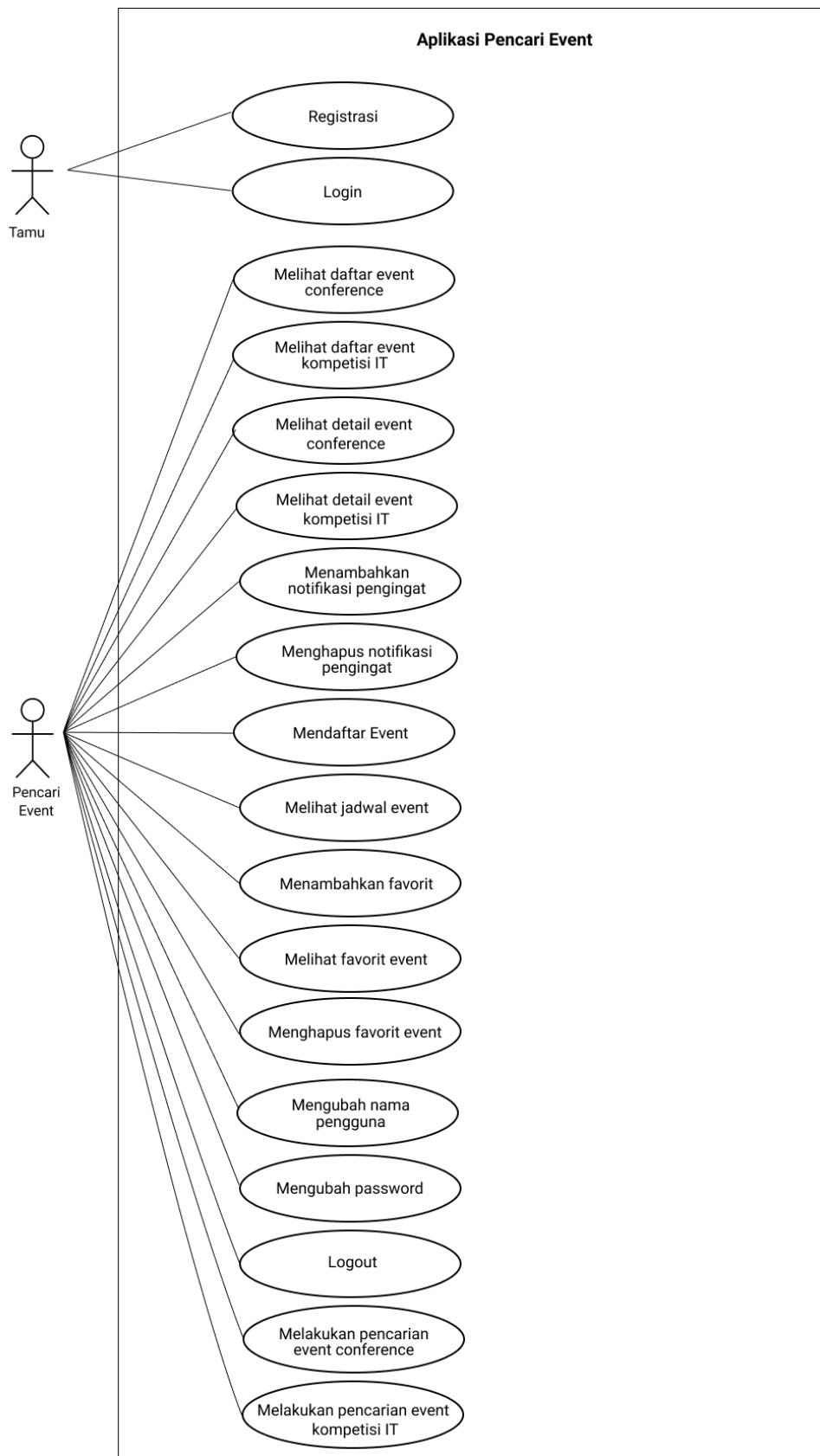
Kebutuhan fungsional pada aktor pencari *event* terdapat beberapa perubahan. Perubahan yang ditambahkan pada aktor pencari *event* adalah fungsional pencarian *event* pada daftar *event conference* dan daftar *event kompetisi IT*. Kebutuhan fungsional untuk pencari *event* tertera pada Tabel 4.4.

Tabel 4.33 Kebutuhan Fungsional Pencari Event

No	Kode Fungsi	Nama Fungsi	Deskripsi
1.	ET-F-018	Melakukan pencarian <i>event conference</i>	Pencari <i>event</i> dapat melakukan pencarian <i>event conference</i> dengan menggunakan judul <i>event</i> sebagai kata kunci pencarian.
2.	ET-F-019	Melakukan pencarian <i>event kompetisi IT</i>	Pencari <i>event</i> dapat melakukan pencarian <i>event kompetisi IT</i> dengan menggunakan judul <i>event</i> sebagai kata kunci pencarian.

4.4.2 Use Case Diagram

Use case diagram pada iterasi kedua terdapat penambahan pada *use case* aplikasi penyelenggara *event*. Penambahan terdapat pada *use case* melakukan pencarian *event conference* dan melakukan pencarian *event kompetisi IT*. *Use case diagram* dapat dilihat pada Gambar 4.28.



Gambar 4.28 Use Case Diagram Aplikasi Pencari Event Event Conference

4.4.3 Use Case Scenario

Use case scenario juga mendapatkan penambahan pada iterasi kedua dikarenakan terdapat perubahan pada *use case* diagram terutama untuk aktor pencari *event*. Penambahan dapat dilihat pada Tabel 4.34 dan Tabel 4.35.

Tabel 4.34 Use Case Scenario Melakukan Pencarian Event Conference

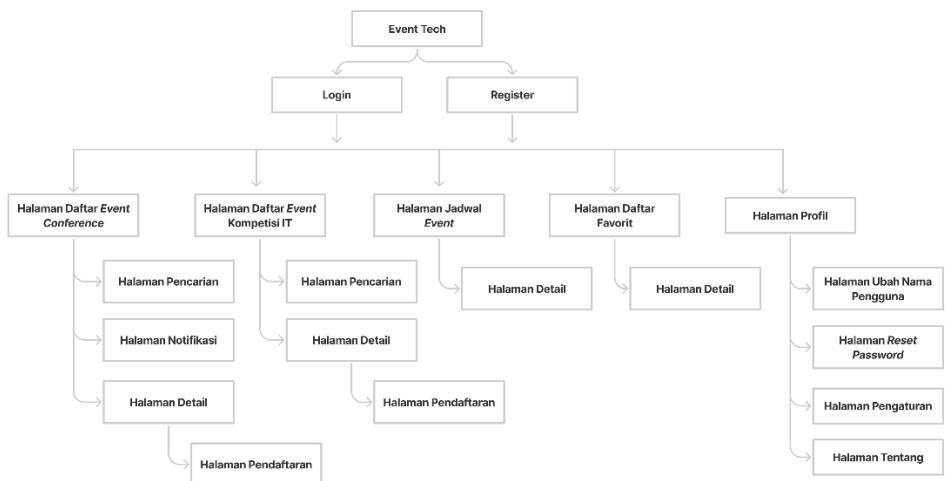
Melakukan pencarian <i>event conference</i>	
Objektif	Aktor dapat melakukan pencarian pada daftar <i>event conference</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman daftar <i>event conference</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol pencarian <i>event</i> 2. Aktor mengisi kata kunci pencarian 3. Sistem menampilkan <i>event</i> yang sesuai dengan kata kunci pencarian yang dimasukan aktor
Skenario alternatif	<ol style="list-style-type: none"> 3. Sistem tidak menemukan <i>event</i> yang sesuai dengan kata kunci pencarian yang dimasukan aktor dan menampilkan pencarian tidak ditemukan
Kondisi sesudah	Aktor berhasil melakukan pencarian <i>event</i> .

Tabel 4.35 Use Case Scenario Melakukan Pencarian Event Kompetisi IT

Melakukan pencarian <i>event kompetisi IT</i>	
Objektif	Aktor dapat melakukan pencarian pada daftar <i>event conference</i>
Aktor	Pencari <i>event</i>
Prasyarat	Aktor telah berada pada halaman daftar <i>event kompetisi IT</i>
Skenario utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol pencarian <i>event</i> 2. Aktor mengisi kata kunci pencarian 3. Sistem menampilkan <i>event</i> yang sesuai dengan kata kunci pencarian yang dimasukan aktor
Skenario alternatif	<ol style="list-style-type: none"> 3. Sistem tidak menemukan <i>event</i> yang sesuai dengan kata kunci pencarian yang dimasukan aktor dan menampilkan pencarian tidak ditemukan
Kondisi sesudah	Aktor berhasil melakukan pencarian <i>event</i> .

4.4.4 Mockup Aplikasi

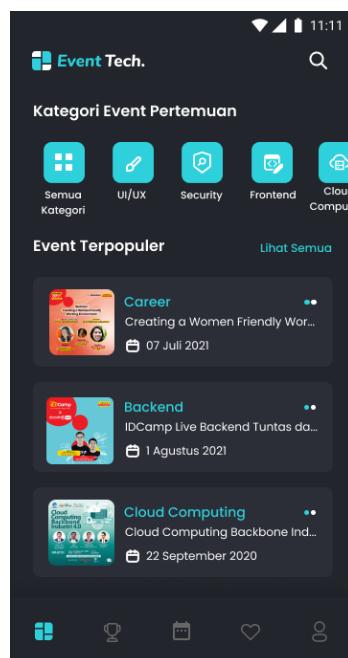
Pada iterasi kedua terdapat beberapa perubahan dan penambahan pada *mockup* yang didapatkan dari hasil evaluasi pada tahap iterasi pertama. Mockup digunakan sebagai bahan evaluasi kepada pengguna pada iterasi kedua. *Information architecture* untuk aplikasi penyelenggara *event* pada iterasi kedua dapat dilihat pada Gambar 4.29.



Gambar 4.29 *Information Architecture* Aplikasi Pencari *Event* Iterasi kedua

4.4.4.1 Perbaikan Halaman Daftar Event Conference

Perbaikan desain dalam bentuk *high-fidelity* untuk halaman daftar *event conference* tertera pada Gambar 4.30.

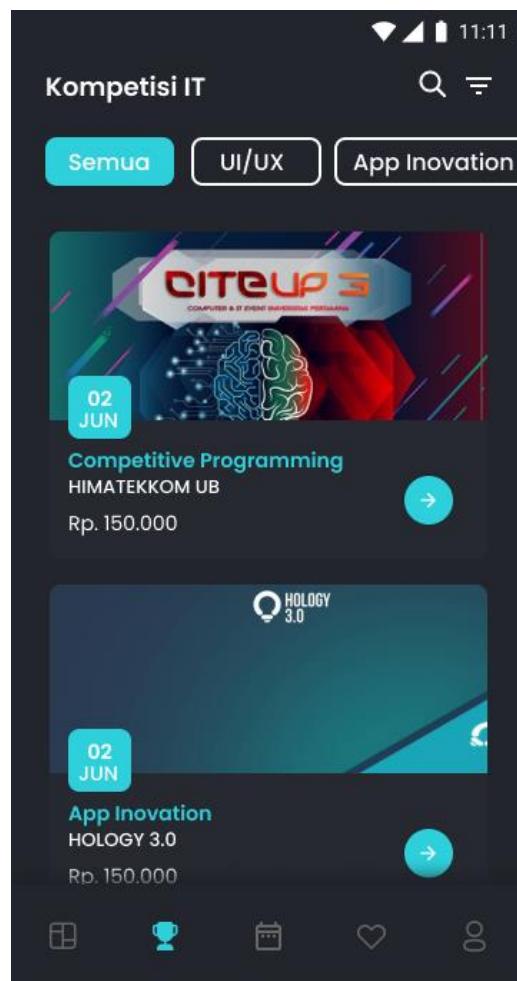


Gambar 4.30 Perbaikan *High-Fidelity* Halaman Daftar *Event Conference*

Pada halaman daftar *event conference* terdapat perbaikan pada pemberian nama kategori *event* menjadi kategori *event* pertemuan. Pada halaman ini di tambahkan juga fungsional pencarian di sebelah fungsional notifikasi.

4.4.4.2 Perbaikan Halaman Daftar Event Kompetisi IT

Perbaikan desain dalam bentuk *high-fidelity* untuk halaman daftar *event* kompetisi IT tertera pada Gambar 4.31.



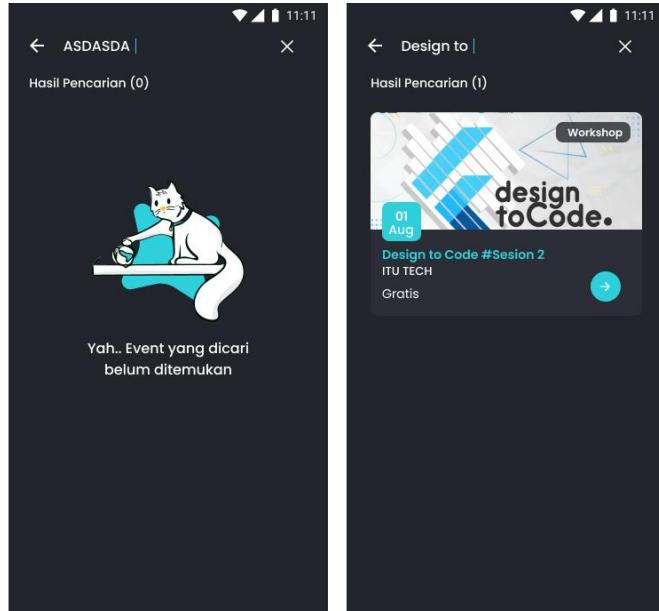
Gambar 4.31 Perbaikan High-Fidelity Halaman Daftar Event Kompetisi IT

Pada halaman daftar *event* kompetisi IT terdapat penambahan fungsional pencarian dan fitur *sorting* untuk mempermudah pengguna dalam mencari *event* yang dicari oleh pencari *event*.

4.4.4.3 Penambahan Halaman Pencarian

Halaman pencarian *event* digunakan pada fungsionalitas pencarian *event conference* dan pencarian *event conference*. Pada halaman pencarian ini terdapat kolom pencarian dan tombol kembali. Ketika pengguna tidak menemukan *event* yang sesuai dengan kata kunci pencarian maka akan ditampilkan ilustrasi beserta pesan bahwa *event* tidak ditemukan. Ketika pengguna menemukan kata kunci yang sesuai dengan *event* maka akan ditampilkan *event* beserta informasinya yang

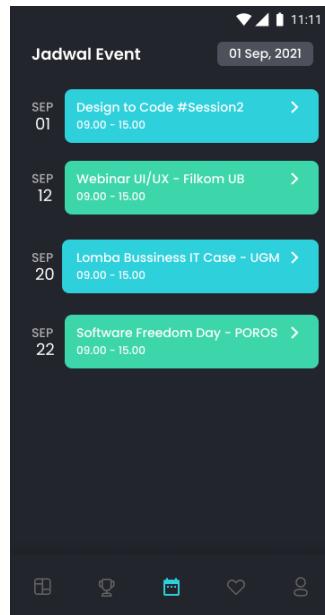
ketika di tekan mengarah ke halaman detail *event*. Penambahan desain dalam bentuk *high-fidelity* untuk halaman pencarian *event* tertera pada Gambar 4.32.



Gambar 4.32 High-Fidelity Halaman Pencarian Event

4.4.4.4 Perbaikan Halaman Jadwal Event

Perbaikan desain dalam bentuk *high-fidelity* untuk halaman jadwal *event* tertera pada Gambar 4.33.



Gambar 4.33 Perbaikan High-Fidelity Halaman Jadwal Event

Pada halaman jadwal *event* terdapat beberapa perubahan pada desain antara lain penunjuk waktu untuk *event* yang sebelumnya ditulis tanggal dan hari diubah menjadi tanggal dan bulan, terdapat juga penambahan hari saat pengguna

memakai aplikasi. Selain itu, ketika jadwal *event* ditekan akan mengarah langsung pada halaman detail *event* tersebut.

4.4.5 Evaluasi

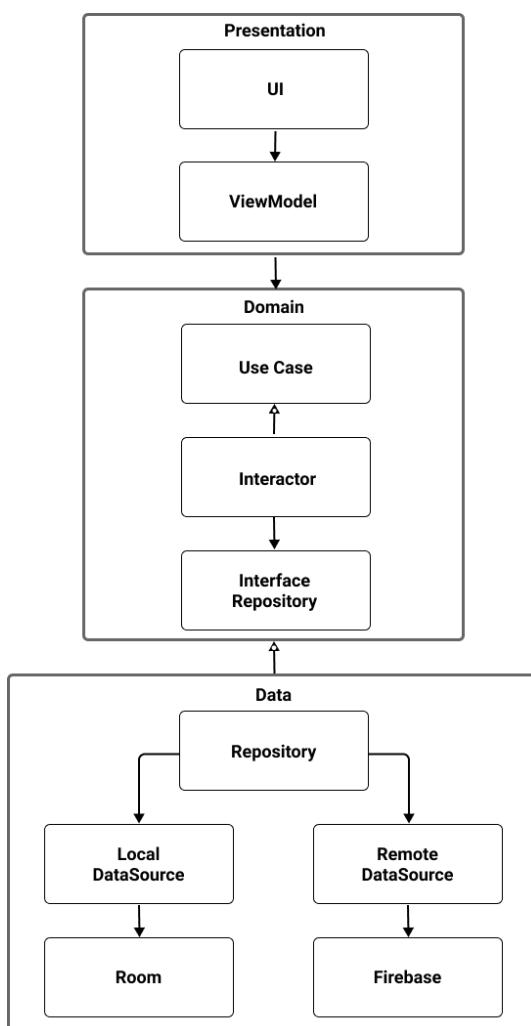
Pada iterasi kedua setelah dilakukan evaluasi kepada pengguna tidak terdapat perbaikan atau penambahan pada desain dan fungsionalitas perangkat lunak. Berdasarkan hasil tersebut, hasil analisis kebutuhan dianggap telah tervalidasi dan dapat dilanjutkan ke tahap perancangan.

BAB 5 PERANCANGAN

Bab perancangan berisi tentang hasil perancangan dari pengembangan perangkat lunak pada penelitian ini. Bab ini memuat penjabaran tentang perancangan arsitektur perangkat lunak, perancangan *sequence diagram*, perancangan *Database*, perancangan *class diagram* dan perancangan *pseudocode*.

5.1 Perancangan Arsitektur Perangkat Lunak

Pada pengembangan perangkat lunak pada penelitian ini *architectural pattern* yang digunakan mengadopsi dari *clean architecture*. Alur pengelolaan data pada *clean architecture* Android tertera pada Gambar 5.1.



Gambar 5.1 Alur Pengelolaan Data Clean Architecture Android

Sumber: Adaptasi Jeevuz (2017)

Gambar diatas merupakan penjabaran lebih mendetail dari Gambar 2.2 *Data Flow Clean Architecture* Android. Pada *layer presentation*, *presenter* yang digunakan untuk menghubungkan UI dengan *layer domain* adalah *viewmodel*.

Pada *layer domain* terdapat *use case*, *interactor* dan *interface repository*. Layer domain digunakan untuk menghubungkan antara layer data dan presentation. Layer data berperan untuk mengelola data yang berasal dari internet dan juga disimpan dalam data lokal pada perangkat pengguna. Penjelasan lebih lanjut mengenai peran setiap entitas pada Gambar 5.1 sebagai berikut:

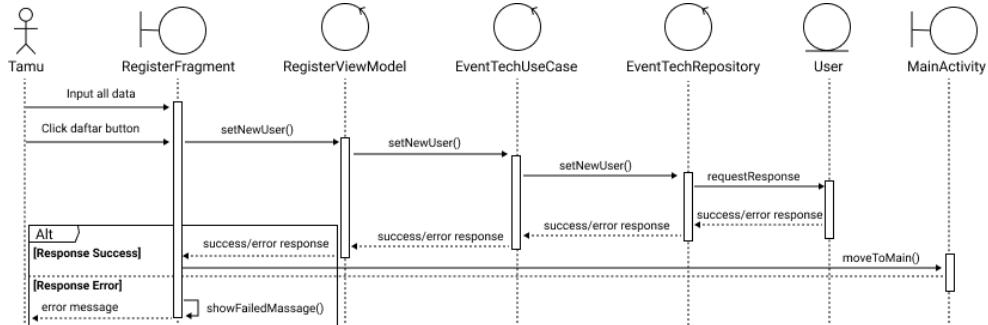
- a. UI adalah entitas yang berperan sebagai tampilan antarmuka yang langsung berhubungan dengan pengguna. Pada entitas ini berisi *activity* dan *fragment*.
- b. *Viewmodel* merupakan kelas yang digunakan untuk memberikan data dan menjaga dari suatu data ketika terjadi *configuration changes* pada perangkat Android. Pada bagan ini *viewmodel* berperan untuk menghubungkan antara *presentation layer* dan *domain*.
- c. *Use case* merupakan kelas abstraksi yang berisi fungsi - fungsi *business logic* dari aplikasi. *Use case* berperan untuk menjadi penghubung antara layer presenter dan domain.
- d. *Interactor* merupakan kelas yang mengimplementasikan *Use case* dan menjadi penghubung dengan *Interface Repository*.
- e. *Interface Repository* merupakan kelas abstraksi yang berisi fungsi - fungsi yang akan diimplementasikan oleh repository pada *data layer* dan menjadi perantara antara *layer domain* dan *layer data*.
- f. *Repository* merupakan kelas yang berisi fungsi fungsi yang digunakan untuk mengelola data yang digunakan oleh aplikasi baik data yang bersumber dari *database* lokal dan dari internet.
- g. *Local Data Source* merupakan kelas yang berisi fungsi - fungsi yang digunakan untuk mengelola database lokal pada aplikasi.
- h. *Remote Data Source* merupakan kelas yang berisi fungsi - fungsi yang digunakan untuk mengelola data yang bersumber dari internet.
- i. Room adalah library yang digunakan untuk mempermudah penggunaan SQLite yang berperan dalam pengelolaan *database* lokal pada aplikasi.
- j. Firebase merupakan layanan yang digunakan untuk menyimpan dan mengelola data pada *server* di internet.

5.2 Perancangan *Sequence Diagram*

Sequence diagram adalah diagram yang digunakan untuk menggambarkan perilaku dari objek dengan memberikan peran dan pesan yang dikirim maupun diterima oleh objek (Ariani S. & Shalahuddin, 2018). Pada bagian ini akan digambarkan beberapa *sequence diagram* yang merupakan perwakilan dari kebutuhan fungsional yang telah didapatkan pada bab sebelumnya. *Sequence diagram* pada bagian ini terdiri dari *sequence diagram* dari proses pengelolaan data berupa *create*, *read*, *update* dan *delete*.

5.2.1 Sequence Diagram Registrasi Penyelenggara Event

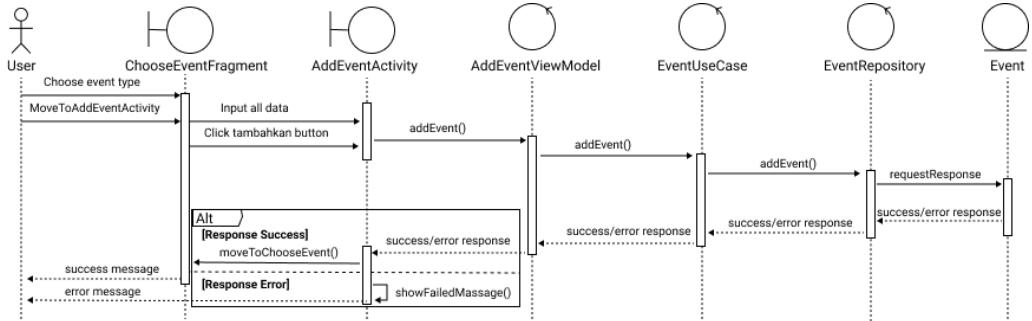
Sequence diagram untuk fungsional registrasi penyelenggara event yang menggambarkan proses perilaku objek dalam registrasi pada aplikasi penyelenggara event tertera pada Gambar 5.2.



Gambar 5.2 Sequence Diagram Registrasi Penyelenggara Event

5.2.2 Sequence Diagram Membuat Event

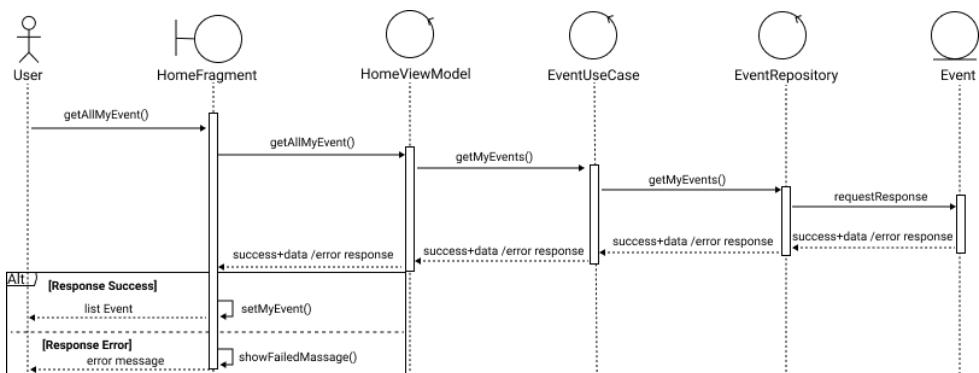
Sequence diagram untuk fungsional membuat event yang menggambarkan proses perilaku objek dalam membuat event pada aplikasi penyelenggara event tertera pada Gambar 5.3.



Gambar 5.3 Sequence Diagram Membuat Event

5.2.3 Sequence Diagram Melihat Daftar Event

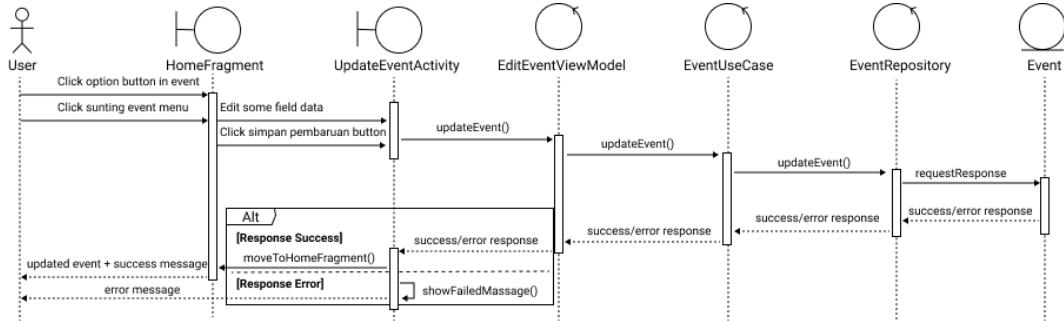
Sequence diagram untuk fungsional melihat daftar event pada aplikasi penyelenggara event tertera pada Gambar 5.4.



Gambar 5.4 Sequence Diagram Melihat Daftar Dari Event

5.2.4 Sequence Diagram Menyunting Event

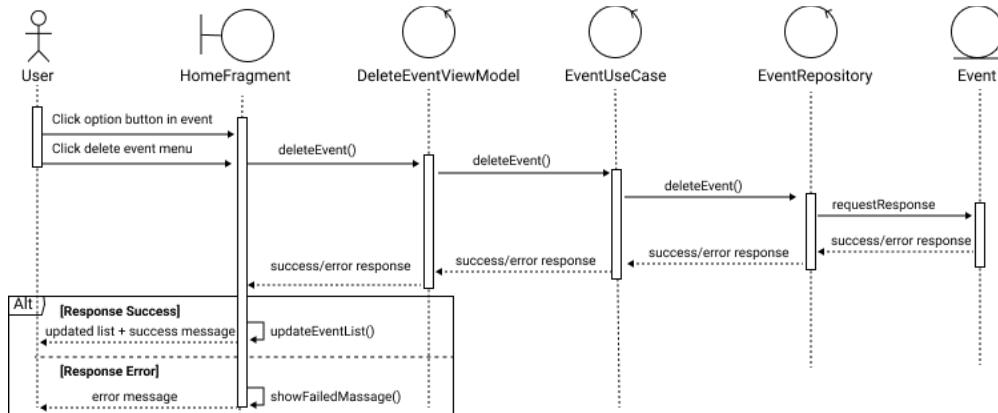
Sequence diagram untuk fungsional menyunting event yang menggambarkan proses perilaku objek dalam menyunting event pada aplikasi penyelenggara event tertera pada Gambar 5.5.



Gambar 5.5 Sequence Diagram Menyunting Event

5.2.5 Sequence Diagram Menghapus Event

Sequence diagram untuk menghapus event yang menggambarkan proses perilaku objek dalam menghapus event pada aplikasi penyelenggara event tertera pada Gambar 5.6.



Gambar 5.6 Sequence Diagram Menghapus Event

5.3 Perancangan Database

Entitas yang digunakan pada pengembangan perangkat lunak ini terdiri dari empat jenis entitas. Adapun entitas tersebut sebagai berikut:

1. *Event Organizer* adalah entitas yang digunakan untuk menyimpan informasi dari penyelenggara event.
2. *Event Hunter* adalah entitas yang digunakan untuk menyimpan informasi dari pencari event.
3. *Event* adalah entitas yang digunakan untuk menyimpan informasi dari event conference dan competition.

4. *Conference Category* adalah kategori untuk *event* yang bersifat pertemuan, entitas ini digunakan untuk membedakan antara kategori dalam *event conference*.
5. *Competition Category* adalah kategori untuk *event* yang bersifat perlombaan, entitas ini digunakan untuk membedakan antara kategori dalam *event competition*.

Tabel 5.1 Entitas *Event Organizer*

No	Nama Atribut	Tipe Data	Keterangan
1.	Uid	<i>String</i>	Identitas pembeda antara entitas <i>event organizer</i>
2.	Username	<i>String</i>	Nama pengguna dari penyelenggara <i>event</i>
3.	Email	<i>String</i>	Alamat <i>email</i> dari penyelenggara <i>event</i>
5.	myEvent	<i>String []</i>	Daftar dari semua uid <i>event</i> yang dimiliki oleh penyelenggara <i>event</i>

Tabel 5.2 Entitas *Event Hunter*

No	Nama Atribut	Tipe Data	Keterangan
1.	Uid	<i>String</i>	Identitas pembeda antara entitas <i>event hunter</i>
2.	Username	<i>String</i>	Nama pengguna dari pencari <i>event</i>
3.	Email	<i>String</i>	Alamat <i>email</i> dari pencari <i>event</i>
4.	Schedule	<i>String []</i>	Daftar uid <i>event</i> yang ditambahkan pengingat jadwal
5.	Favorite	<i>String []</i>	Daftar uid <i>event</i> yang telah ditambahkan favorit

Tabel 5.3 Entitas *Event*

No	Nama Atribut	Tipe Data	Keterangan
1.	Uid	<i>String</i>	Identitas pembeda antara entitas <i>event</i>
2.	eventName	<i>String</i>	Nama dari <i>event</i>

3.	<i>eventType</i>	<i>String</i>	Tipe dari <i>event</i> yang digunakan sebagai pembeda antara <i>event conference</i> dan <i>event competition</i>
4.	<i>eventCategory</i>	<i>String</i>	Kategori dari <i>event</i> seperti <i>backend</i> dan <i>frontend</i>
5.	Price	<i>String</i>	Harga untuk mengikuti <i>event</i>
6.	Date	<i>String</i>	Tanggal pelaksanaan <i>event</i>
7.	Time	<i>String</i>	Waktu pelaksanaan <i>event</i>
8.	Location	<i>String</i>	Lokasi <i>event</i> diselenggarakan
9.	linkRegistration	<i>String</i>	Tautan untuk mendaftar <i>event</i> yang digunakan untuk mengarah pada tampilan <i>webview</i>
10.	Description	<i>String</i>	Deskripsi tentang <i>event</i>
11.	Prerequisite	<i>String</i>	Prasyarat yang harus dimiliki sebelum mengikuti <i>event</i>
12.	<i>eventCover</i>	<i>String</i>	Poster dari <i>event</i>
13.	numbersOfView	<i>Number</i>	Jumlah <i>event</i> di lihat oleh pencari <i>event</i> yang digunakan sebagai penentu tingkat popular <i>event</i>
14.	numbersOfRegistrationClick	<i>Number</i>	Jumlah halaman pendaftaran dibuka oleh pencari <i>event</i>

Tabel 5.4 Entitas Conference Category

No	Nama Atribut	Tipe Data	Keterangan
1.	Uid	<i>String</i>	Identitas pembeda antara entitas <i>conference category</i>
2.	categoryName	<i>String</i>	Nama dari kategori

3.	categoryAvatar	<i>String</i>	Avatar dari kategori
----	----------------	---------------	----------------------

Tabel 5.5 Entitas *Competition Category*

No	Nama Atribut	Tipe Data	Keterangan
1.	Uid	<i>String</i>	Identitas pembeda antara entitas <i>competition category</i>
2.	categoryName	<i>String</i>	Nama dari kategori

5.3.2 Perancangan Database Entitas *Event Organizer*

Perancangan *database* dalam bentuk skema struktur data JSON untuk entitas *event organizer* dapat dilihat pada Gambar 5.7.

```

    {
      "event_organizer": {
        "uid": {
          "uid": "ASFasdSFqqxP8g9tQksbYXoJ1",
          "username": "Hology 4.0",
          "email": "hology@ub.ac.id",
          "myEvent": {
            "0": "T9QksbYXoJ1lVCqqxP8g",
            "1": "TSFkYXeboJ10lGSqqxP8g9",
            "2": "SoS0lGSXp8g9tSFYXe"
          }
        }
      }
    }
  
```

Gambar 5.7 Skema Struktur Data JSON Entitas *Event Organizer*

5.3.3 Perancangan Database Entitas *Event Hunter*

Perancangan *database* dalam bentuk skema struktur data JSON untuk entitas *event hunter* dapat dilihat pada Gambar 5.8.

```

    {
      "event_hunter": {
        "uid": {
          "uid": "QksbYXoJ1lVT9CqqxP8g"
          "username": "Ilham Dwi Muchlison"
          "email": "ilham.uklison@gmail.com"
        },
        "schedule": {
          "0": "Ts13kYXeboJ101GSqqxP8g9"
          "1": "TSFkYXeboJ101GSqqxP8g9"
          "2": "SoS01GSXp8g9tSFYXe"
        }
      },
      "favorite": {
        "0": "AsfS01GSXp8g9tSFYXe"
        "1": "T9QksbYXoJ1lVCqqxP8g"
        "2": "SZas01GSXp8g9tSFYXe"
      }
    }
  }
}

```

Gambar 5.8 Skema Struktur Data JSON Entitas *Event Hunter*

5.3.4 Perancangan *Database* Entitas *Event*

Perancangan *database* dalam bentuk skema struktur data JSON untuk entitas *event* dapat dilihat pada Gambar 5.9.

```

    {
      "event": {
        "uid": {
          "uid": "1VCqqxP8g9tQksbYXoJ1"
          "eventName": "Webinar Web PHP"
          "eventType": "Conference"
          "eventCategory": "Backend"
          "price": "0"
          "date": "30 Agustus 2021"
          "time": "09:00"
          "location": "Zoom"
          "linkRegistration": "0"
          "description": "Event diselenggarakan oleh POROS yang berfokus pada pengembangan WEB"
          "prerequisite": "Memahami Basic dari Pemrograman OOP"
          "eventCover": "https://repository.com/image.jpg"
          "number Of View": 10
          "number Of Registration Click": 10
        }
      }
    }

```

Gambar 5.9 Skema Struktur Data JSON Entitas *Event*

5.3.5 Perancangan Database Entitas *Conference Category*

Perancangan database dalam bentuk skema struktur data JSON untuk entitas *conference category* dapat dilihat pada Gambar 5.10.

```
{  
  "conference_category": {  
    "uid": {  
      "uid": "AsdkajsASDlkesmsdpf"  
      "categoryName": "Mobile Apps"  
      "categoryAvatar": "https://repository.com/image.jpg"  
    }  
  }  
}
```

Gambar 5.10 Skema Struktur Data JSON Entitas *Conference Category*

5.3.6 Perancangan Database Entitas *Competition Category*

Perancangan database dalam bentuk skema struktur data JSON untuk entitas *competition category* dapat dilihat pada Gambar 5.11.

```
{  
  "competition_category": {  
    "uid": {  
      "uid": "Xasr2sdXoJ11VCqqxP8g"  
      "categoryName": "Business Case"  
    }  
  }  
}
```

Gambar 5.11 Skema Struktur Data JSON Entitas *Competition Category*

5.4 Perancangan *Class Diagram*

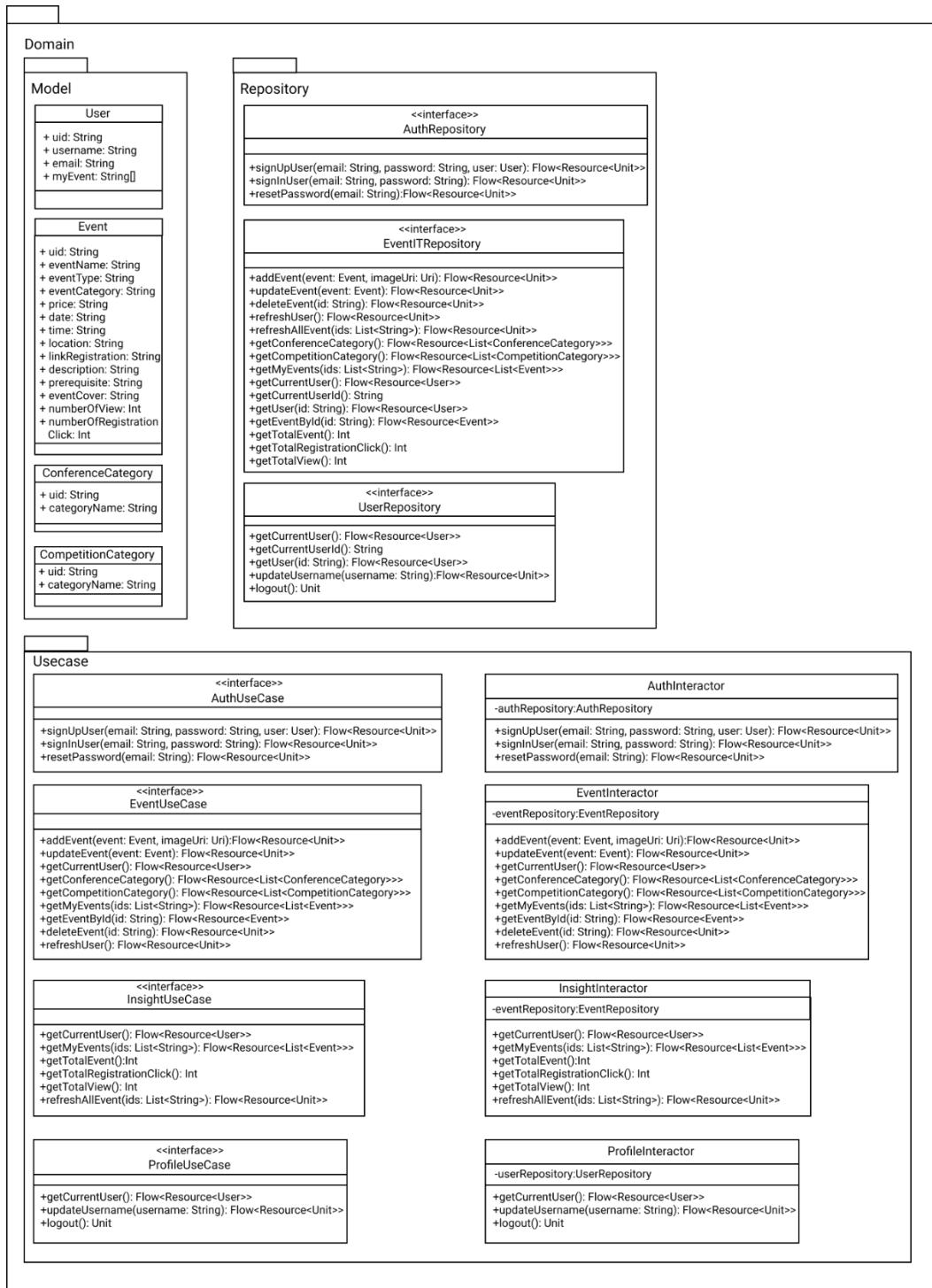
Class diagram adalah diagram yang digunakan untuk mendefinisikan kelas-kelas dan relasi antar kelas yang akan digunakan pada perangkat lunak. *Class diagram* berfungsi sebagai acuan ketika pengembang aplikasi mengembangkan aplikasi tersebut agar dokumentasi dari perangkat lunak sesuai dengan perangkat lunak ketika diimplementasikan (Ariani S. & Shalahuddin, 2018).

5.4.1 *Class Diagram* Aplikasi Penyelenggara Event

Pada aplikasi penyelenggara event *class diagram* dikelompokan menjadi tiga modul yaitu *domain*, *data* dan *features*.

5.4.1.1 Class Diagram Modul Domain Aplikasi Penyelenggara Event

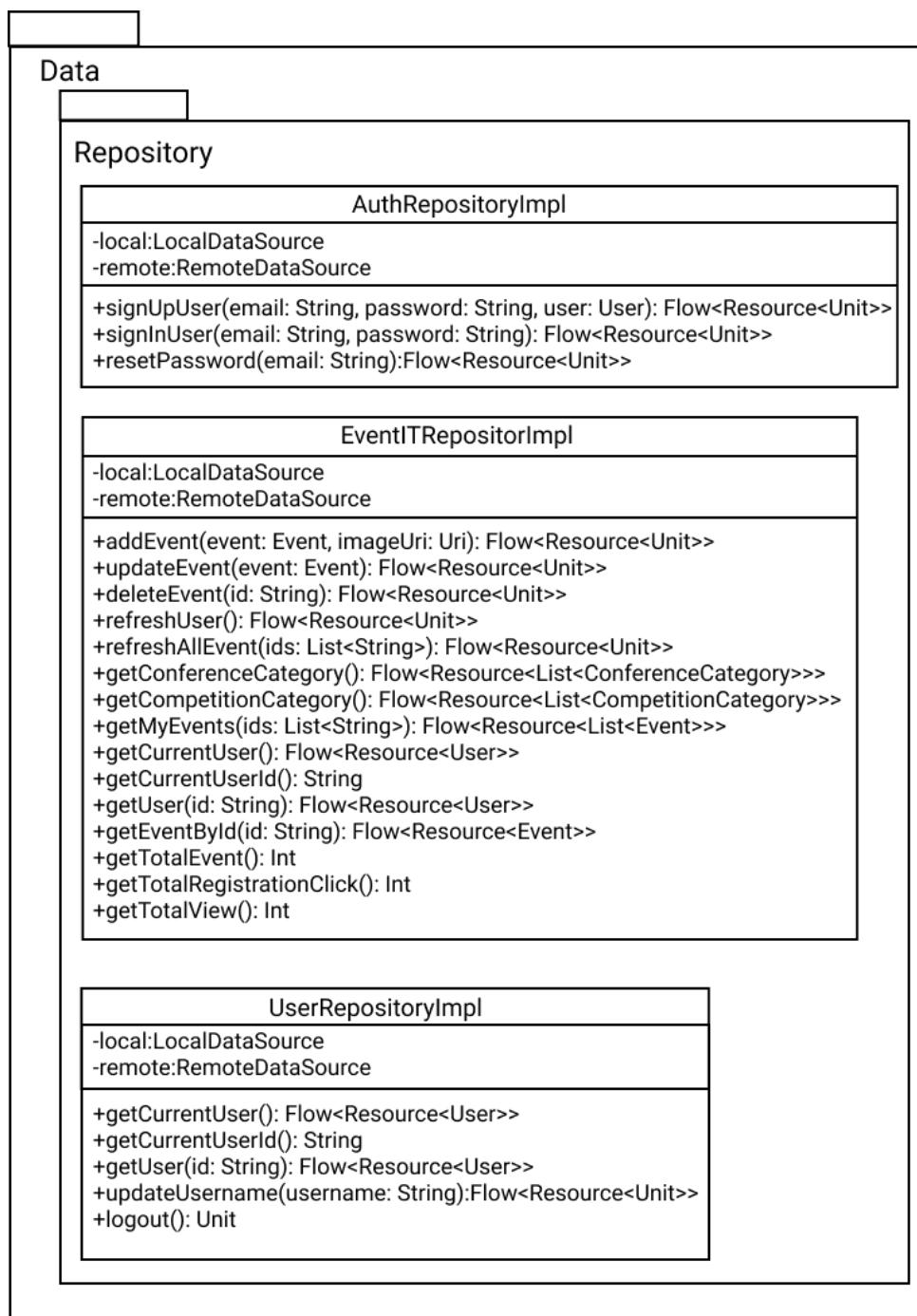
Hasil perancangan *class diagram* pada modul *domain* aplikasi penyelenggara event tertera pada Gambar 5.12.



Gambar 5.12 *Class Diagram* Modul Domain Aplikasi Penyelenggara Event

5.4.1.2 Class Diagram Modul Data Aplikasi Penyelenggara Event

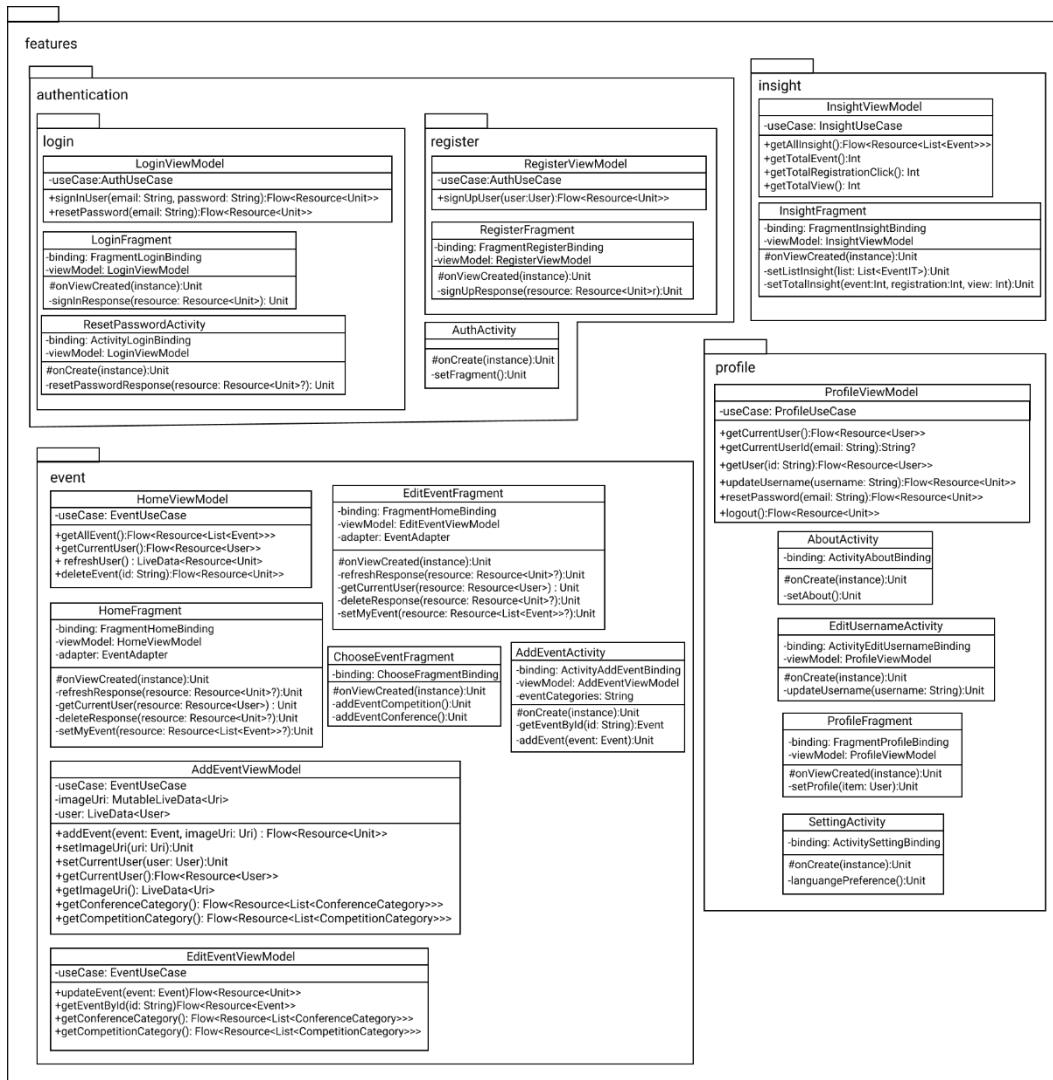
Hasil perancangan *class diagram* pada modul data aplikasi penyelenggara event tertera pada Gambar 5.13.



Gambar 5.13 Class Diagram Modul Domain Aplikasi Penyelenggara Event

5.4.1.3 Class Diagram Modul Features Aplikasi Penyelenggara Event

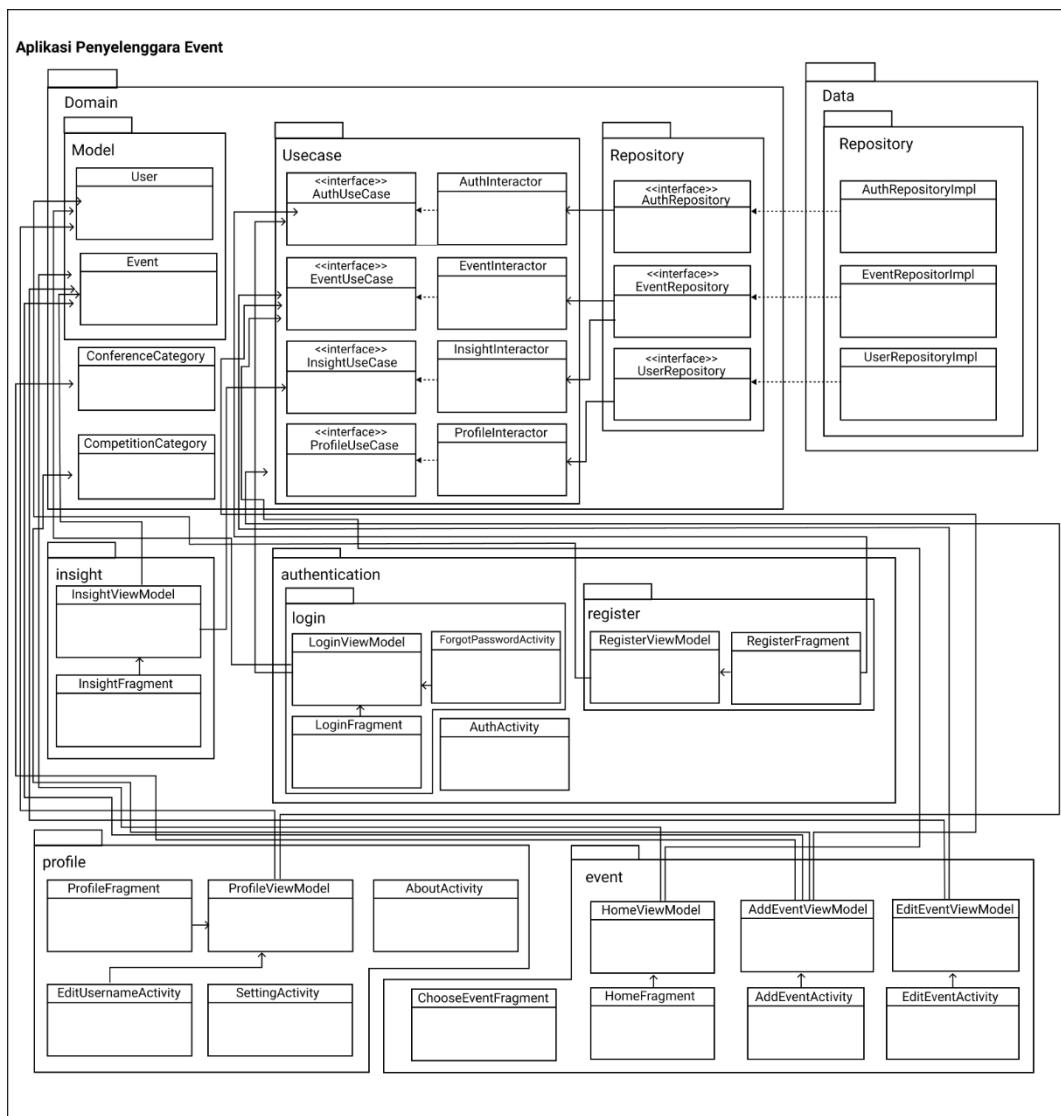
Hasil perancangan *class diagram* pada modul data aplikasi penyelenggara event tertera pada Gambar 5.14.



Gambar 5.14 Class Diagram Modul Features Aplikasi Penyelenggara Event

5.4.1.4 Relasi Class Diagram Aplikasi Penyelenggara Event

Hasil perancangan *class diagram* pada aplikasi penyelenggara event tertera pada Gambar 5.15.



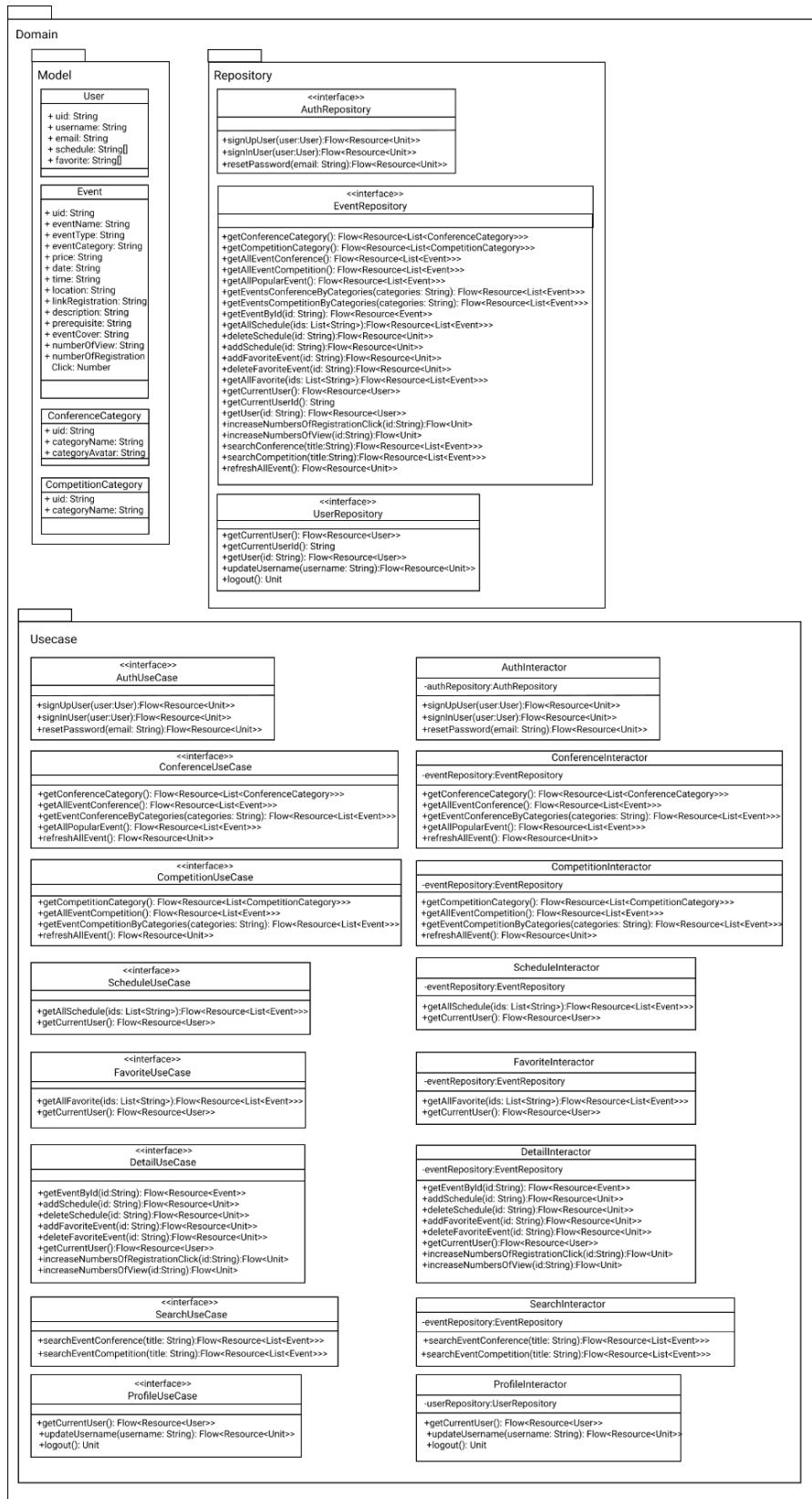
Gambar 5.15 Relasi *Class Diagram* Aplikasi Penyelenggara Event

5.4.2 *Class Diagram* Aplikasi Pencari Event

Pada aplikasi pencari event *class diagram* dikelompokan menjadi tiga modul yaitu domain, data dan *features*.

5.4.2.1 *Class Diagram* Modul Domain Aplikasi Pencari Event

Hasil perancangan *class diagram* pada modul domain aplikasi pencari event tertera pada Gambar 5.16.



Gambar 5.16 Class Diagram Modul Features Aplikasi Pencari Event

5.4.2.2 Class Diagram Modul Data Aplikasi Pencari Event

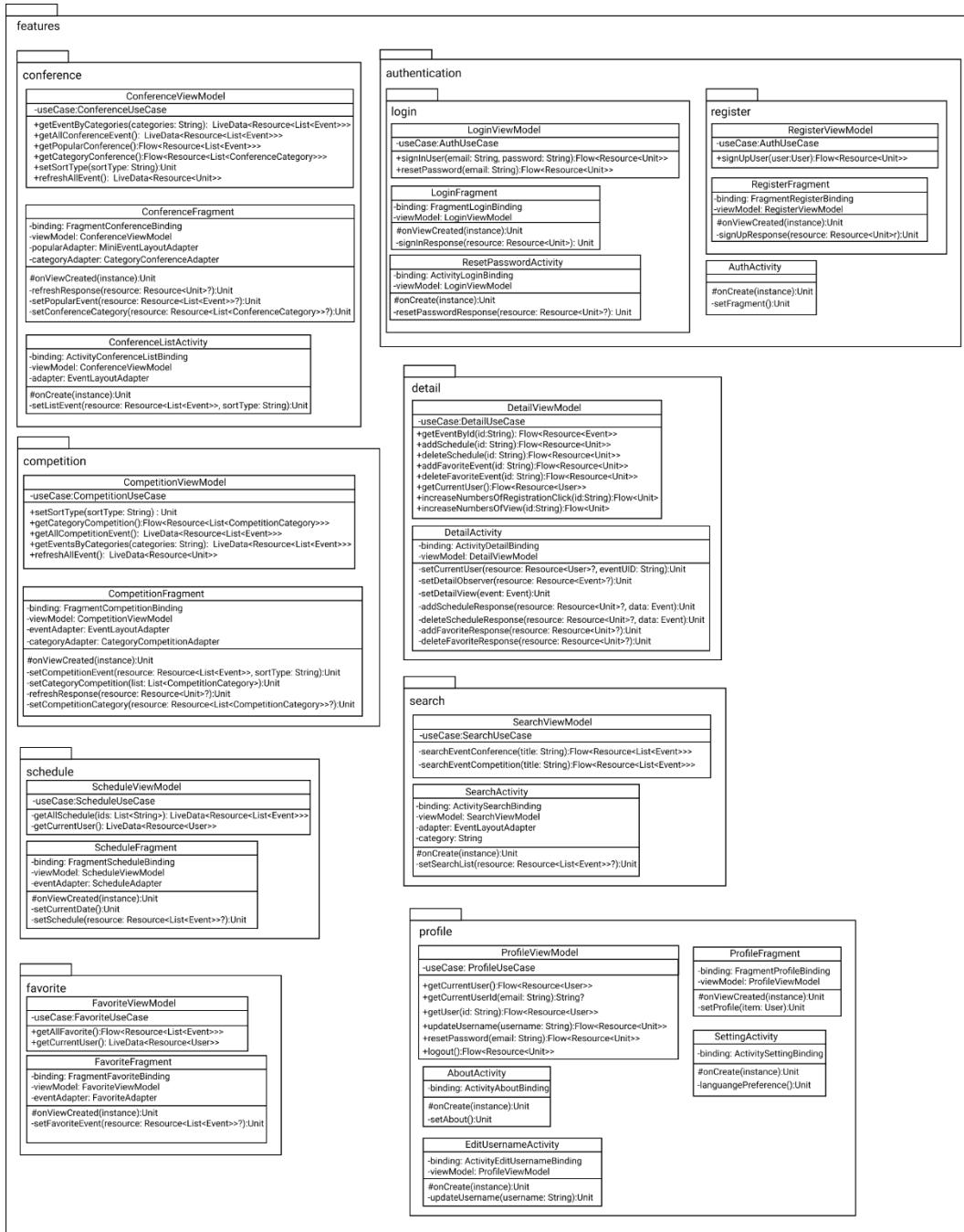
Hasil perancangan *class diagram* pada modul data aplikasi pencari event tertera pada Gambar 5.17.



Gambar 5.17 Class Diagram Modul Features Aplikasi Pencari Event

5.4.2.3 Class Diagram Modul Features Aplikasi Pencari Event

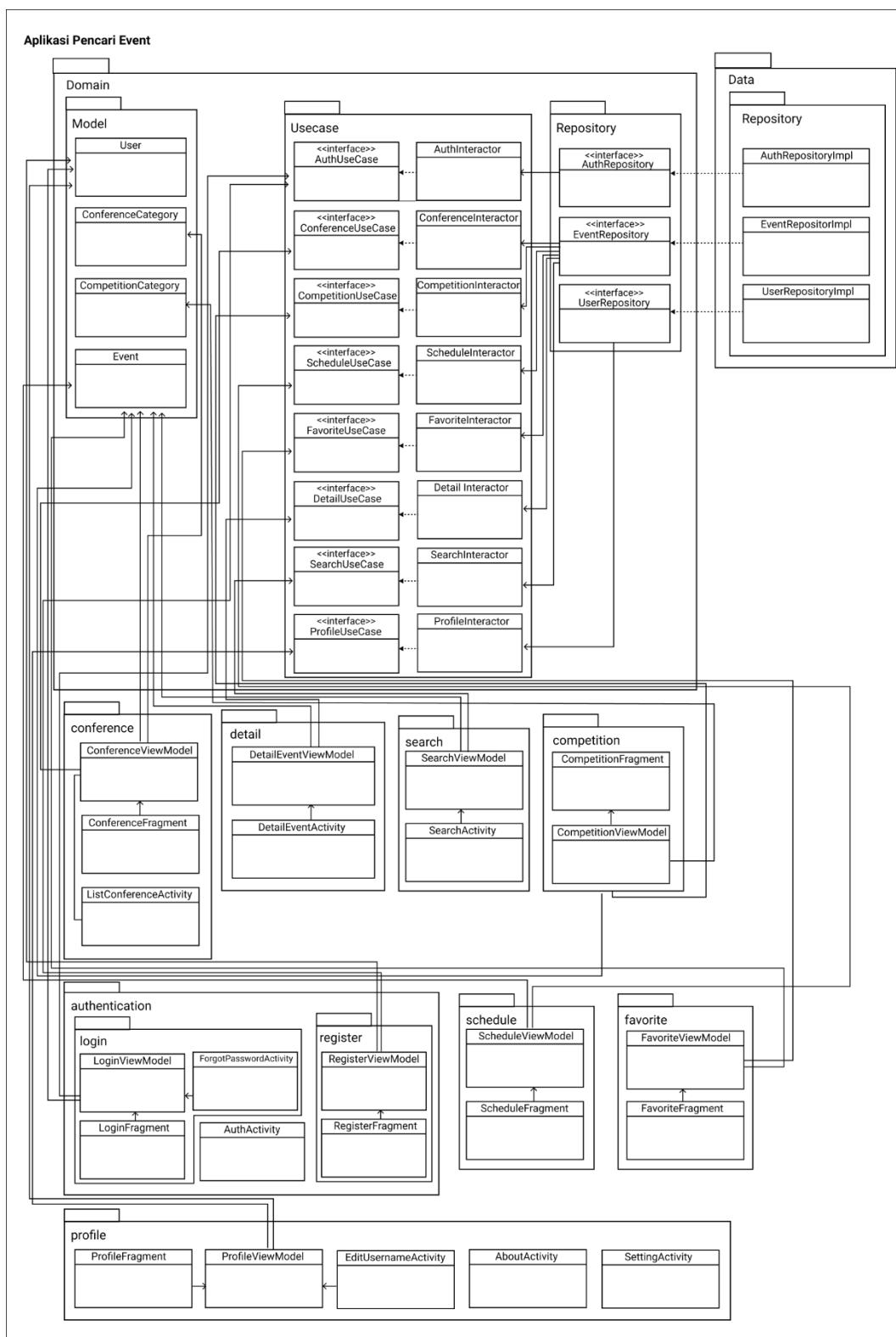
Hasil perancangan *class diagram* pada modul *features* aplikasi pencari event tertera pada Gambar 5.18.



Gambar 5.18 Class Diagram Modul Features Aplikasi Pencari Event

5.4.2.4 Relasi Class Diagram Aplikasi Pencari Event

Hasil perancangan *class diagram* pada aplikasi pencari event tertera pada Gambar 5.19.



Gambar 5.19 Relasi Class Diagram Aplikasi Pencari Event

5.5 Perancangan *Pseudocode*

Pseudocode adalah bahasa yang digunakan untuk menggambarkan algoritma ketika menyelesaikan suatu permasalahan tanpa harus memikirkan bahasa pemrograman dan *syntax* yang dipakai. *Pseudocode* berfungsi untuk mempermudah seorang programmer menuliskan kodennya dan memahami algoritma dari suatu kode tersebut (Nilakandi, 2020).

5.5.1 Perancangan *Pseudocode Registrasi Penyelenggara Event*

Pseudocode untuk fungsional registrasi penyelenggara *event* yang menggambarkan algoritma dalam registrasi pada aplikasi penyelenggara *event* tertera pada Tabel 5.6.

Tabel 5.6 Pseudocode Registrasi Penyelenggara Event

```
Input: email, password, user data
output: UserResponse

Start
    set function to Asynchronous function
    call createUserWithEmailAndPassword function from Firebase
    input email and password to createUserWithEmailAndPassword
    function
    when createUserWithEmailAndPassword response
        is Success
            set response data with type UserResponse to user
            data
            emit user data to Firestore
        is Error
            emit error message response to FirebaseResponse
        is Empty
            emit FirebaseResponse with Empty
End
```

5.5.2 Perancangan *Pseudocode Membuat Event*

Pseudocode untuk fungsional membuat *event* yang menggambarkan algoritma dalam membuat *event* pada aplikasi penyelenggara *event* tertera pada Tabel 5.7.

Tabel 5.7 Pseudocode Membuat Event

```
Input: event data, posterUri
output: EventResponse

Start
    set function to Asynchronous function
```

```

    generateEventUID from firestore
        input event collection, eventUID, posterUri to
        uploadPicture function

        when uploadPictureResponse
            is Success
                set response data with type EventResponse to event
                data
                emit event data to Firestore
            is Error
                emit response message to FirebaseResponse
            is Empty
                emit FirebaseResponse with Empty
    End

```

5.5.3 Perancangan Pseudocode Melihat Daftar Event

Pseudocode untuk fungsional melihat daftar *event* yang menggambarkan algoritma dalam melihat daftar *event* pada aplikasi penyelenggara *event* tertera pada Tabel 5.8.

Tabel 5.8 Pseudocode Melihat Daftar Event

```

Input: -
Output: listEvent, Response message

Start
    set function to Asynchronous function
        input event collection, userUID to getDocument function
    when Response
        is Success
            emit response data to FirebaseResponse Success
        is Error
            emit response message to FirebaseResponse
        is Empty
            emit FirebaseResponse with Empty
    End

```

5.5.4 Perancangan Pseudocode Menyunting Event

Pseudocode untuk fungsional menyunting *event* yang menggambarkan algoritma dalam menyunting *event* pada aplikasi penyelenggara *event* tertera pada Tabel 5.9.

Tabel 5.9 Pseudocode Menyunting Event

Input: event data

```

output: -
Start
    set function to Asynchronous function
    input event collection, eventUID to updateDocument function
    try
        set collection with event collection
        get eventUID
        set document id with event UID
        call update method
        emit FirebaseResponse with Success response
    catch
        emit error message response to FirebaseResponse
End

```

5.5.5 Perancangan Pseudocode Menghapus Event

Pseudocode untuk fungsional menghapus *event* yang menggambarkan algoritma dalam menghapus *event* pada aplikasi penyelenggara *event* tertera pada Tabel 5.10.

Tabel 5.10 Pseudocode Menghapus Event

Input: eventUID output: Unit
Start set function to Asynchronous function input event collection, eventUID to deleteDocument function try remove event id from user myEvent field call delete file method emit FirebaseResponse with Success response catch emit error message response to FirebaseResponse End

BAB 6 IMPLEMENTASI

Bab implementasi berisi tentang hasil implementasi dari pengembangan perangkat lunak pada penelitian ini. Bab ini memuat penjabaran tentang spesifikasi lingkungan pengembangan perangkat lunak, implementasi *database*, implementasi kode program, dan implementasi antarmuka.

6.1 Spesifikasi Lingkungan Pengembangan Perangkat Lunak

Pada bagian ini dijelaskan spesifikasi lingkungan yang digunakan dalam pengembangan perangkat lunak. Spesifikasi lingkungan pengembangan dibagi menjadi tiga yaitu spesifikasi perangkat keras pengembang perangkat lunak yang dapat dilihat pada Tabel 6.1, spesifikasi perangkat keras perangkat bergerak yang dapat dilihat pada Tabel 6.2, dan spesifikasi perangkat lunak yang dapat dilihat pada Tabel 6.3.

Tabel 6.1 Spesifikasi Perangkat Keras Pengembang Perangkat Lunak

Komponen	Spesifikasi
Model Sistem	Acer Aspire E 14
Processor	Intel Core i5-8250U 1.6Ghz
Memory	8 GB DDR4
Storage	1000 GB

Tabel 6.2 Spesifikasi Perangkat Keras Perangkat Bergerak

Komponen	Spesifikasi
Model Sistem	Redmi Note 7
Processor	Octa-core Max 2.20GHz
Memory	4.00 GB
Storage	64 GB

Tabel 6.3 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi	Linux Ubuntu 21.10
Bahasa Pemrograman	Kotlin
Development Tools	Android Studio 2020.3.1

6.2 Implementasi *Database*

Pada bagian ini berisi hasil implementasi dari perancangan *database* pada bab sebelumnya. Implementasi *database* pada perangkat lunak dapat dilihat pada Gambar 6.1 sampai Gambar 6.5.

6.2.1 Implementasi *Database Entitas Event Organizer*

Implementasi *database* pada Cloud Firestore untuk entitas *event organizer* dapat dilihat pada Gambar 6.1.

The screenshot shows the Cloud Firestore interface with the path `event_organizer > AyWZW1e81VcoLCG52NgghAHv59c2`. The left sidebar lists collections: competition_category, conference_category, event, event_hunter, and event_organizer. The event_organizer collection is expanded, showing a list of documents. One document is selected, displaying its details:

- email:** "dscub@gmail.com"
- myEvent** (array):
 - 0: "6b3LMQV6lPyMV8UB5lcJ"
 - 1: "AHgKSAGHS6yvwCrSgrH"
 - 2: "ubnYzil10dYmanhOjddC"
 - 3: "lldGcLFHMWNPSEEwq3Ck"
 - 4: "rdnK3HlXaRgpOLXIBJIS"
 - 5: "M3r33DJu3z88SsGlshd"
 - 6: "QM1PAirh0h78P1VAbKhU"
 - 7: "6ps2mpLqpF6efdUGK2qV"
- uid:** "AyWZW1e81VcoLCG52NgghAHv59c2"
- username:** "DSC Indonesia"

Gambar 6.1 Implementasi *Database Entitas Event Organizer*

6.2.2 Implementasi *Database Entitas Event Hunter*

Implementasi *database* pada Cloud Firestore untuk entitas *event hunter* dapat dilihat pada Gambar 6.2.

The screenshot shows the Cloud Firestore interface with the path `event_hunter > 8Biw0iKwMhbKv6SSgzoOs2sAAQ2`. The left sidebar lists collections: competition_category, conference_category, event, event_hunter, and event_organizer. The event_hunter collection is expanded, showing a list of documents. One document is selected, displaying its details:

- favorite** (array):
 - 0: "0gKNIPWLExSeVW54Qvh"
 - 1: "QM1PAirh0h78P1VAbKhU"
 - 2: "RU00zgp24W20QNQSQMeB"
- schedule** (array):
 - 0: "BkL5lErnZNiuv4PAjSWk"
 - 1: "2EoeQjtmjsm1OnvkPtPV"
 - 2: "M3r33DJu3z88SsGlshd"
 - 3: "QM1PAirh0h78P1VAbKhU"
 - 4: "GFUnHgoU8mFvGfkszFkx"
 - 5: "RY4T9yjG7x3CXEwuiLAB"
- uid:** "8Biw0iKwMhbKv6SSgzoOs2sAAQ2"
- username:** "suhaili99"

Gambar 6.2 Implementasi *Database Entitas Event Hunter*

6.2.3 Implementasi Database Entitas Event

Implementasi *database* pada Cloud Firestore untuk entitas *event* dapat dilihat pada Gambar 6.3.

Gambar 6.3 Implementasi Database Entitas Event

6.2.4 Implementasi Database Entitas Conference Category

Implementasi *database* pada Cloud Firestore untuk entitas *conference category* dapat dilihat pada Gambar 6.4.

Gambar 6.4 Implementasi Database Entitas Conference Category

6.2.5 Implementasi Database Entitas Competition Category

Implementasi *database* pada Cloud Firestore untuk entitas *competition category* dapat dilihat pada Gambar 6.5.

event-tech-9ef5c	competition_category		375PGjGPtrcKFpTxrrTa
Start collection		Add document	Start collection
competition_category >		375PGjGPtrcKFpTxrrTa >	Add field
conference_category	SIA7fdkdGvdJ6b1nMcr1		categoryName: "Business IT Case"
event	V0t0sdP9LkOwba0UgAqQ		uid: "375PGjGPtrcKFpTxrrTa"
event_hunter	h0b8PDkLFu4eP0Guag1Y		
event_organizer	kb9HArLXkTvPswY0VzWX		
	lnwnmECQj2I8r8KwB87W		
	y39PIHJsy0ENTH6CwjJ		

Gambar 6.5 Implementasi Database Entitas Entitas Competition Category

6.3 Implementasi Kode Program

Pada bagian ini berisi hasil implementasi kode program dari rancangan *pseudocode* pada bab sebelumnya. Implementasi kode program pada perangkat lunak dapat dilihat pada Tabel 6.4 sampai Tabel 6.10.

6.3.1 Implementasi Kode Program Registrasi Penyelenggara Event

Implementasi kode program untuk registrasi penyelenggara event yang dalam aplikasi penyelenggara event tertera pada Tabel 6.4.

Tabel 6.4 Kode Program Registrasi Penyelenggara Event

```

1  fun signUp(email: String, password: String, user: User): Flow<FirebaseResponse<UserResponse>> =
2      flow {
3          createUserWithEmailAndPassword(email, password).collect{response->
4              when (response) {
5                  is FirebaseResponse.Success -> {
6                      emitAll(
7                          setDocument<User, UserResponse>(
8                              FirebaseConstant.FirebaseCollection.USER,
9                              response.data,
10                             user.copy(
11                                 uid = response.data
12                                     )
13                             )
14                         )
15                     }
16                 is FirebaseResponse.Error -> {
17                     emit(FirebaseResponse.Error(response.errorMessage))
18                 }
19                 FirebaseResponse.Empty -> {
20                     emit(FirebaseResponse.Empty)
21                 }
22             }
23         }
}

```

Pada kode program diatas baris 1 berisi parameter fungsi yang digunakan sebagai nilai masukan pada fungsi `signUp` dan nilai return dari fungsi. Baris 3 memanggil fungsi `createUserWithEmailAndPassword` dan memasukan nilai yang didapat dari parameter fungsi `signUp` untuk digunakan pada fungsi `create`. Pada

baris 4 menentukan tindakan selanjutnya dari fungsi `signUp` ketika telah mendapatkan `response` setelah memanggil fungsi `createUserWithEmailAndPassword`. Baris 5 sampai dengan baris 15 membuat sebuah objek baru bernama `user` dan diberikan sebuah id yang didapat dari data `response` pada saat memanggil fungsi `createUserWithEmailAndPassword` sebelumnya. Pada baris 16 sampai 18 ketika membuat `user` mengalami error maka fungsi akan melakukan `emit` kepada fungsi `signUp` dengan nilai pesan error. Pada baris 19 sampai 21 ketika `response` kosong maka fungsi `signUp` akan melakukan `emit` nilai kosong.

6.3.2 Implementasi Kode Program Membuat Event

Implementasi kode program untuk membuat `event` pada aplikasi penyelenggara `event` tertera pada Tabel 6.5.

Tabel 6.5 Kode Program Membuat Event

```

1  fun insertEvent(event: Event, imageUri: Uri, userId: String): 
2      Flow<FirebaseResponse<EventResponse>> = 
3      flow { 
4          val eventUid = generateDocumentId(EVENT_COLLECTION) 
5          uploadPicture( 
6              EVENT_COLLECTION, 
7              eventUid, 
8              imageUri 
9          ).first { response -> 
10             when (response) { 
11                 is FirebaseResponse.Success -> { 
12                     val imageDownloadUrl = response.data 
13                     val updateEventData = event.copy( 
14                         uid = eventUid, 
15                         eventCover = imageDownloadUrl 
16                     ) 
17                     addArrayStringValueAtDocField( 
18                         FirebaseFirestore.USER, 
19                         userId, 
20                         FirebaseConstant.Field.MY_EVENT, 
21                         eventUid 
22                     ) 
23                     emitAll( 
24                         setDocument<Event, EventResponse>( 
25                             EVENT_COLLECTION, 
26                             eventUid, 
27                             updateEventData 
28                         ) 
29                     ) 
30                     is FirebaseResponse.Error -> { 
31                         emit(FirebaseResponse.Error(response.errorMessage)) 
32                     } 
33                     FirebaseResponse.Empty -> { 
34                         emit(FirebaseResponse.Empty) 
35                     } 
36                     true 
37                 } 
38             } 
39         } 
40     }

```

Pada kode program di atas baris 1 berisi parameter yang terdiri dari data dari entitas *event*, Uri dari gambar poster *event*, dan juga identitas dari pengguna. Pada baris ke 3 dibuat sebuah uid yang nantinya akan digunakan sebagai identitas pada *event*. Pada baris ke 5 sampai 9 gambar dari poster dalam bentuk Uri dimasukan ke dalam *firebase storage* dan didapatkan tautan gambar dari *firebase storage* yang akan dimasukan ke dalam entitas *event*. Pada baris 11 sampai dengan baris 29 data *event* di-*emit* ke dalam firestore ketika *response* yang didapatkan dari unggahan poster bernilai *success*. Pada baris 30 sampai dengan 31 ketika data error maka akan di-*emit* pesan dari error. Pada baris 32 sampai dengan 34 ketika *response* yang didapat kosong maka akan di-*emit* nilai kosong.

6.3.3 Implementasi Kode Program Melihat Daftar Event

Implementasi kode program untuk melihat daftar *event* pada aplikasi penyelenggara *event* tertera pada Tabel 6.6 dan Tabel 6.7.

Tabel 6.6 Kode Program Fungsi GetMyEvents

1	fun getMyEvents(ids: List<String>):
2	Flow<FirebaseResponse<List<EventResponse>>> =
3	getDocumentsWhereIds(
4	FirebaseConstant.FirebaseCollection.EVENT_COLLECTION,
5	FirebaseConstant.Field.UID,
6	ids
)

Tabel 6.7 Kode Program GetDocumentsWhereIds

1	inline fun <reified ResponseType> getDocumentsWhereIds(
2	collection: String,
3	fieldName: String,
4	value: List<String>
5): Flow<FirebaseResponse<List<ResponseType>>> =
6	flow {
7	val result = firestore
8	.collection(collection)
9	.whereIn(fieldName,value)
10	.get()
11	.await()
12	if (result.isEmpty) {
13	emit(FirebaseResponse.Empty)
14	} else {
15	emit(FirebaseResponse.Success(result.toObjects(ResponseType::class.java)))
16	}
17	}.catch {
18	emit(FirebaseResponse.Error(it.message.toString()))
19	}.flowOn(Dispatchers.IO)
20	
21	

Pada kelas fungsi *getMyEvents* yang memiliki parameter masukan data berupa daftar id dari *event* yang dimiliki oleh pengguna. Pada baris 2 sampai 6 dipanggil sebuah fungsi *getDocumentsWhereIds*. Fungsi *getDocumentsWhereIds* memiliki

parameter masukan `collection` yang merujuk lokasi dari `collection` pada `firestore`, `fieldname` sebagai penanda lokasi `field` dan `value` berupa daftar id yang akan diambil. Pada fungsi tersebut data yang berhasil diambil akan di-`emit` dalam bentuk `object`. Ketika data dipanggil terdapat data kosong maka akan di-`emit` data `empty`. Dan ketika terdapat `error` akan catch dan di-`emit` pesan `error` dari response.

6.3.4 Implementasi Kode Program Menyunting Event

Implementasi kode program untuk menyunting `event` pada aplikasi penyelenggara `event` tertera pada Tabel 6.8 dan Tabel 6.9.

Tabel 6.8 Kode Program Fungsi UpdateEvent

1	<code>fun updateEvent(event: Event): Flow<FirebaseResponse<EventResponse>> =</code>
2	<code> setDocument(</code>
3	<code> FirebaseConstant.FirebaseCollection.EVENT_COLLECTION,</code>
4	<code> event.uid,</code>
5	<code> event</code>
6	<code>)</code>

Tabel 6.9 Kode Program Fungsi SetDocument

1	<code>inline fun <RequestType, reified ResponseType> setDocument(</code>
2	<code> collection: String,</code>
3	<code> docId: String,</code>
4	<code> document: RequestType</code>
5	<code>): Flow<FirebaseResponse<ResponseType>> =</code>
6	<code> flow {</code>
7	<code> firestore</code>
8	<code> .collection(collection)</code>
9	<code> .document(docId)</code>
10	<code> .set(document as Any)</code>
11	<code> .await()</code>
12	<code> }</code>
13	<code> emitAll(getDocument<ResponseType>(collection, docId))</code>
14	<code> } .catch {</code>
15	<code> emit(FirebaseResponse.Error(it.message.toString()))</code>
16	<code> } .flowOn(Dispatchers.IO)</code>
17	<code></code>

Pada fungsi `updateEvent` terdapat parameter data masukan berupa data dari `object event` dan dipanggil fungsi `setDocument`. Pada fungsi `setDocument` dipanggil fungsi `set` dari `firestore`, ketika data dipanggil terdapat data kosong maka akan di-`emit` data `empty`. Dan ketika terdapat `error` akan catch dan di-`emit` pesan `error` dari response.

6.3.5 Implementasi Kode Program Menghapus Event

Implementasi kode program untuk fungsionalitas menghapus `event` pada aplikasi penyelenggara `event` tertera pada Tabel 6.10.

Tabel 6.10 Kode Program Menghapus Event

1	<code>fun deleteEvent(id: String, userId: String): Flow<FirebaseResponse<Unit>> =</code>
2	<code> flow {</code>

```

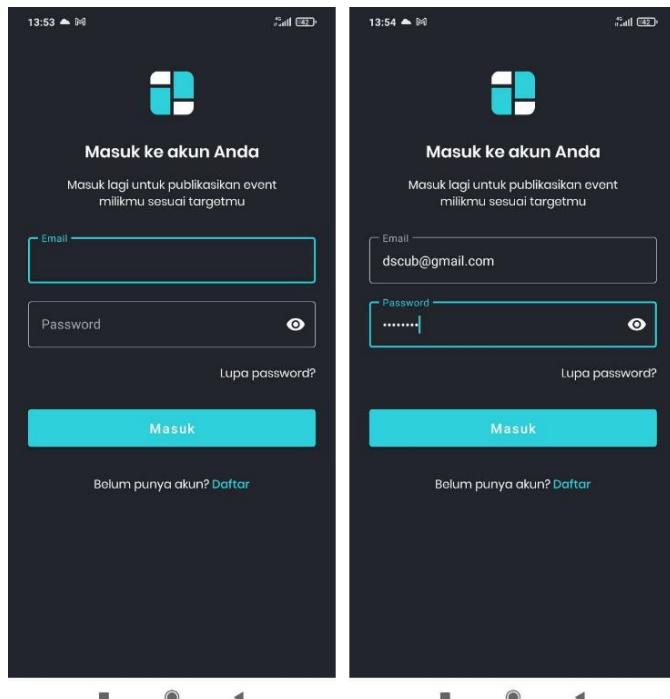
3     deleteDocument(EVENT_COLLECTION, id).first<Unit,Unit>(this) {
4         removeArrayStringValueAtDocField(
5             FirebaseCollection.USER,
6             userId,
7             FirebaseConstant.Field.MY_EVENT,
8             Id
9         )
10        emitAll(deleteFile(EVENT_COLLECTION,id))
11    }
12 }.catch {
13     emit(FirebaseResponse.Error(it.message.toString()))
14 }
```

Pada fungsi `deleteEvent` terdapat parameter data masukan berupa id dari `event` dan id dari pengguna pemilik dari `event`. Pada baris 3 terdapat fungsi `deleteDocument` yang akan menghapus dokumen entitas yang sesuai dengan id yang dimasukan pada parameter. Pada baris 4 id dari `event` yang dihapus digunakan untuk menghapus id yang terdapat pada field `myEvent` yang terdapat pada dokumen milik pengguna. Setelah data berhasil dihapus maka akan dipanggil fungsi `deleteFile` yang akan menghapus gambar dari `event` yang terdapat pada `firebase storage`. Ketika `response` menghapus gagal maka akan di-`emit` pesan `error` dari `response`.

6.4 Implementasi Antarmuka

6.4.1 Implementasi Tampilan Login Penyelenggara Event

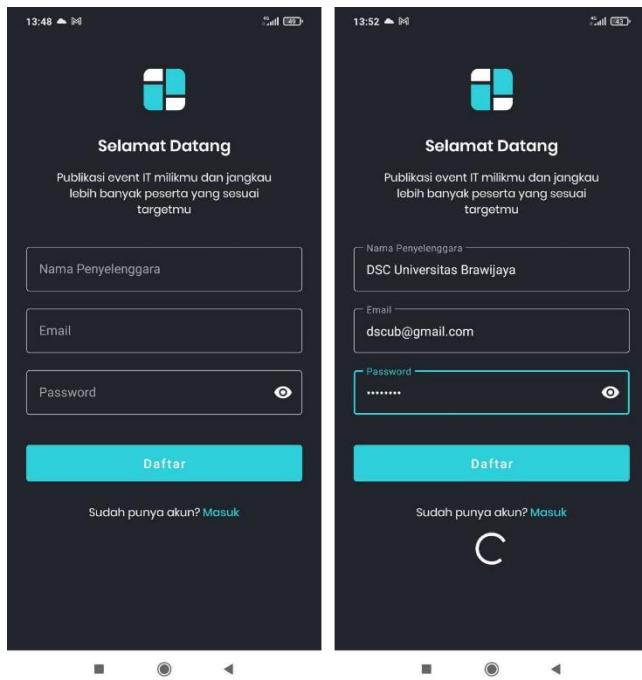
Implementasi tampilan untuk halaman `login` penyelenggara `event` tertera pada Gambar 6.6.



Gambar 6.6 Implementasi Tampilan Login Penyelenggara Event

6.4.2 Implementasi Tampilan Registrasi Penyelenggara Event

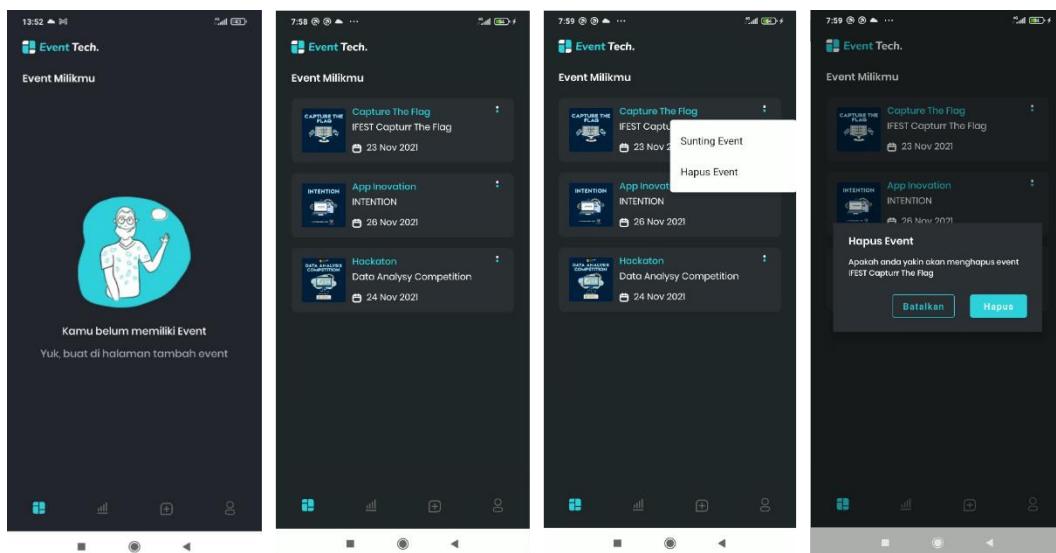
Implementasi tampilan untuk halaman *login* penyelenggara event tertera pada Gambar 6.7.



Gambar 6.7 Implementasi Tampilan Registrasi Penyelenggara Event

6.4.3 Implementasi Tampilan Beranda Penyelenggara Event

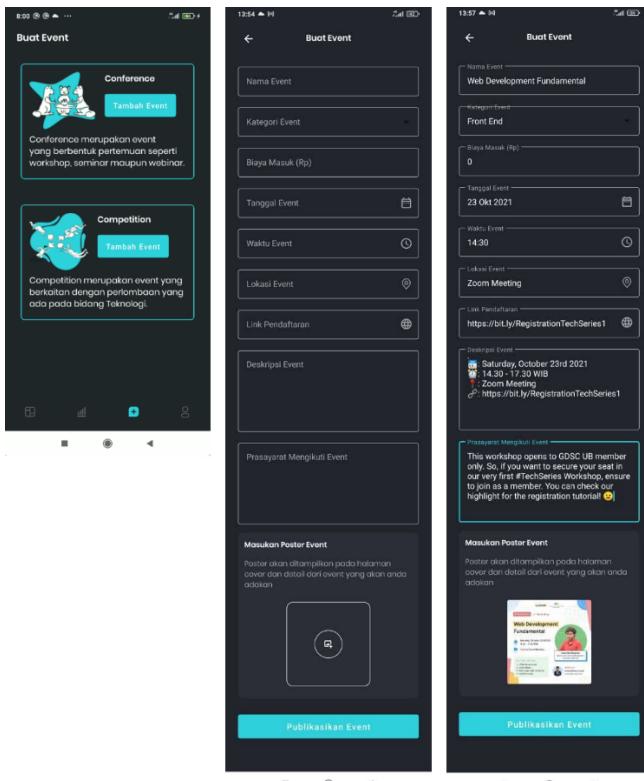
Implementasi tampilan untuk halaman beranda penyelenggara event tertera pada Gambar 6.8.



Gambar 6.8 Implementasi Tampilan Beranda Penyelenggara Event

6.4.4 Implementasi Tampilan Tambahan Event

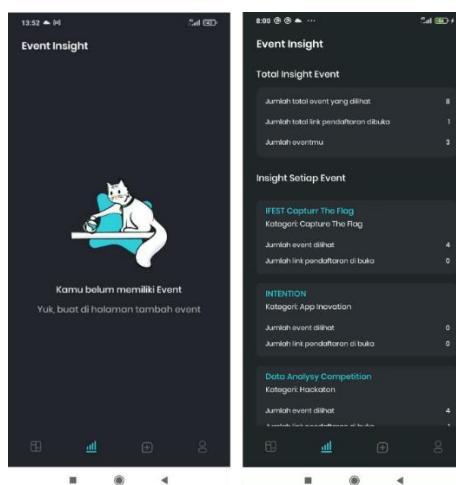
Implementasi Tampilan untuk halaman tambahan *event* tertera pada Gambar 6.9.



Gambar 6.9 Implementasi Tampilan Tambahan Event

6.4.5 Implementasi Tampilan Insight Event

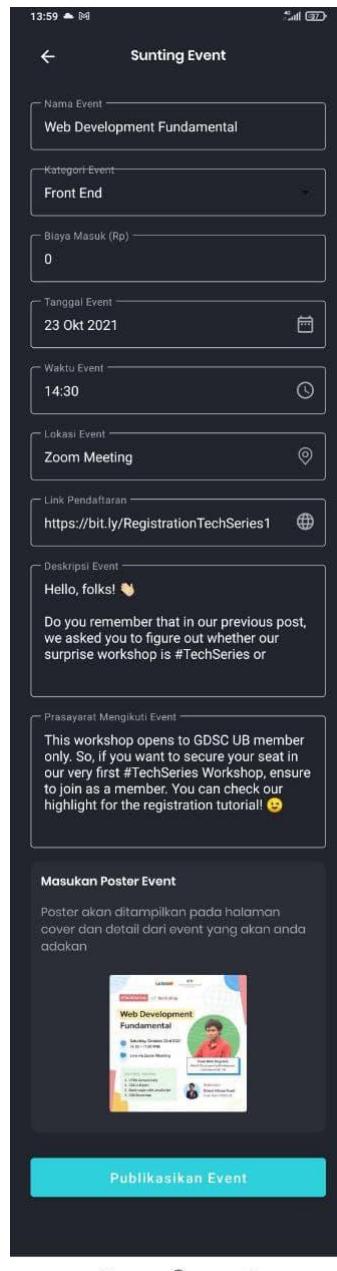
Implementasi tampilan untuk halaman *insight event* tertera pada Gambar 6.10



Gambar 6.10 Implementasi Tampilan Insight Event

6.4.6 Implementasi Tampilan Sunting Event

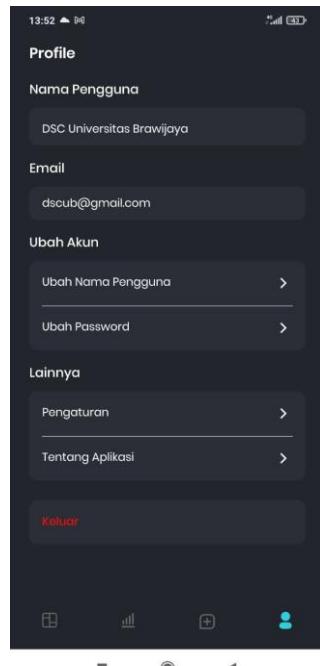
Implementasi tampilan untuk halaman sunting event tertera pada Gambar 6.11



Gambar 6.11 Implementasi Tampilan Sunting Event

6.4.7 Implementasi Tampilan Profil Penyelenggara Event

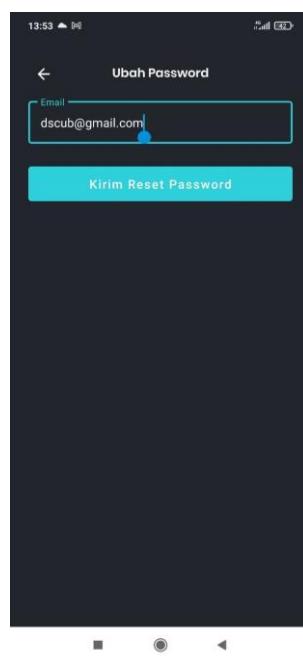
Implementasi tampilan untuk profil penyelenggara event tertera pada Gambar 6.12.



Gambar 6.12 Implementasi Tampilan Profil Penyelenggara Event

6.4.8 Implementasi Tampilan Ubah Password Penyelenggara

Implementasi tampilan untuk halaman ubah *password* penyelenggara tertera pada Gambar 6.13.



Gambar 6.13 Implementasi Tampilan Ubah Password Penyelenggara

6.4.9 Implementasi Tampilan Ubah Nama Penyelenggara

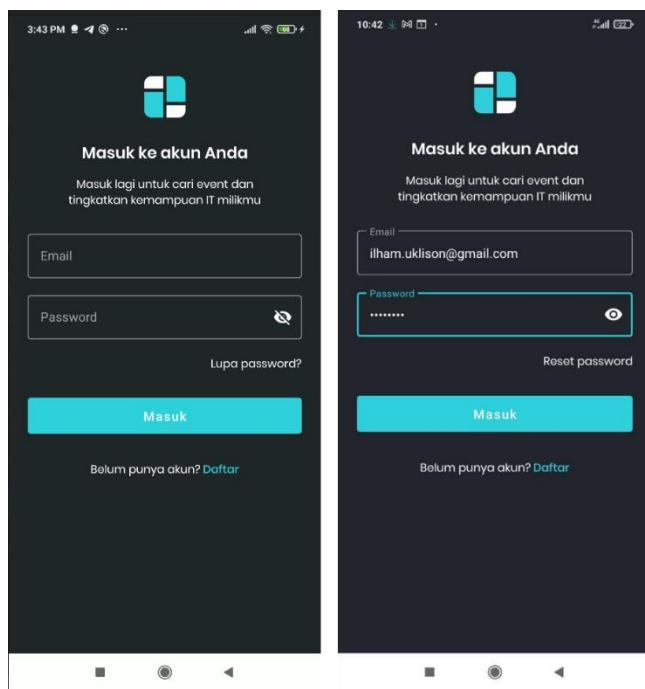
Implementasi tampilan untuk halaman ubah nama penyelenggara tertera pada Gambar 6.14



Gambar 6.14 Implementasi Tampilan Ubah Nama Penyelenggara

6.4.10 Implementasi Tampilan Login Pencari Event

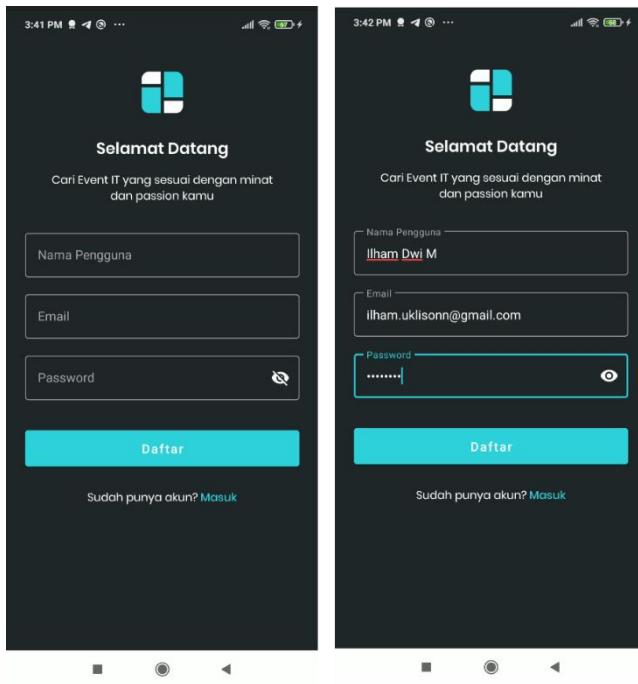
Implementasi tampilan untuk halaman *login* pencari event tertera pada Gambar 6.15.



Gambar 6.15 Implementasi Tampilan Login Pencari Event

6.4.11 Implementasi Tampilan Registrasi Pencari Event

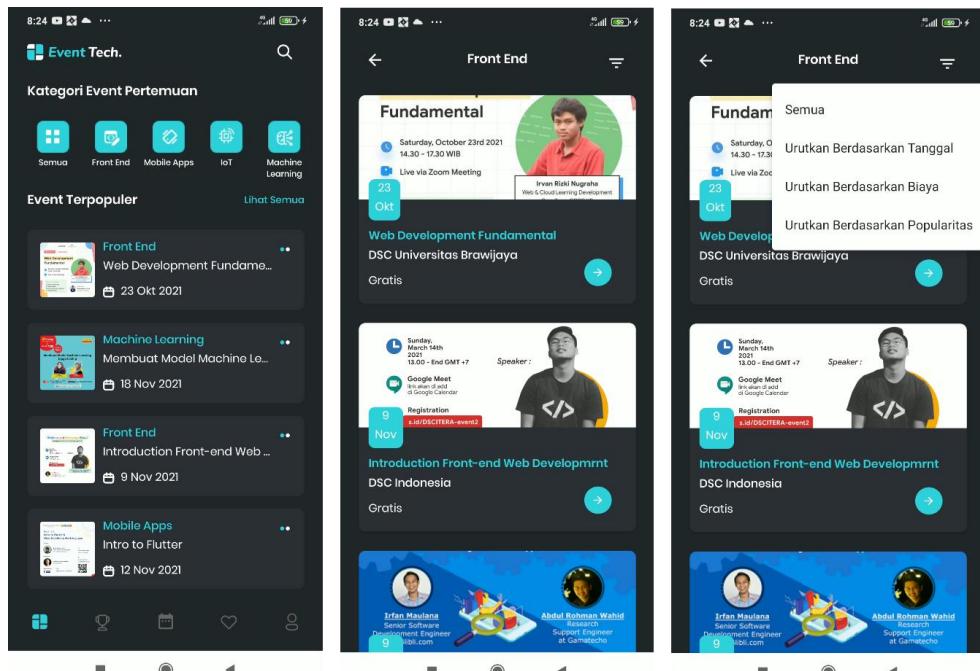
Implementasi tampilan untuk halaman registrasi penyelenggara event tertera pada Gambar 6.16



Gambar 6.16 Implementasi Tampilan Registrasi Pencari Event

6.4.12 Implementasi Tampilan Daftar Event Conference

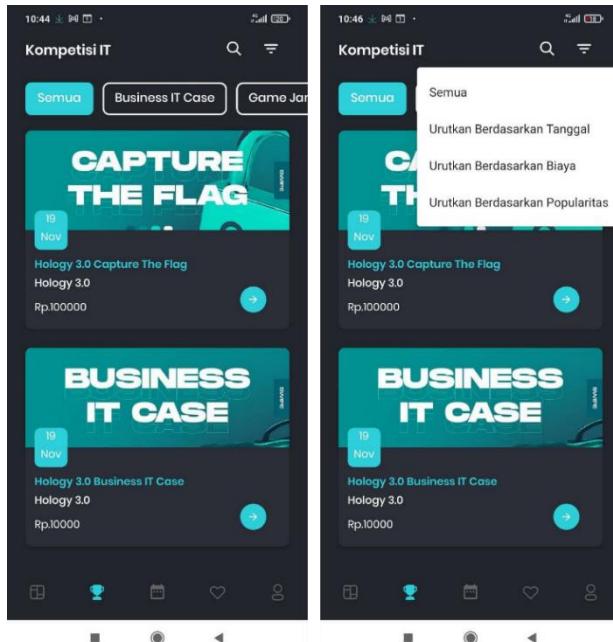
Implementasi tampilan untuk halaman daftar event conference tertera pada Gambar 6.17.



Gambar 6.17 Implementasi Tampilan Daftar Event Conference

6.4.13 Implementasi Tampilan Daftar Event Kompetisi IT

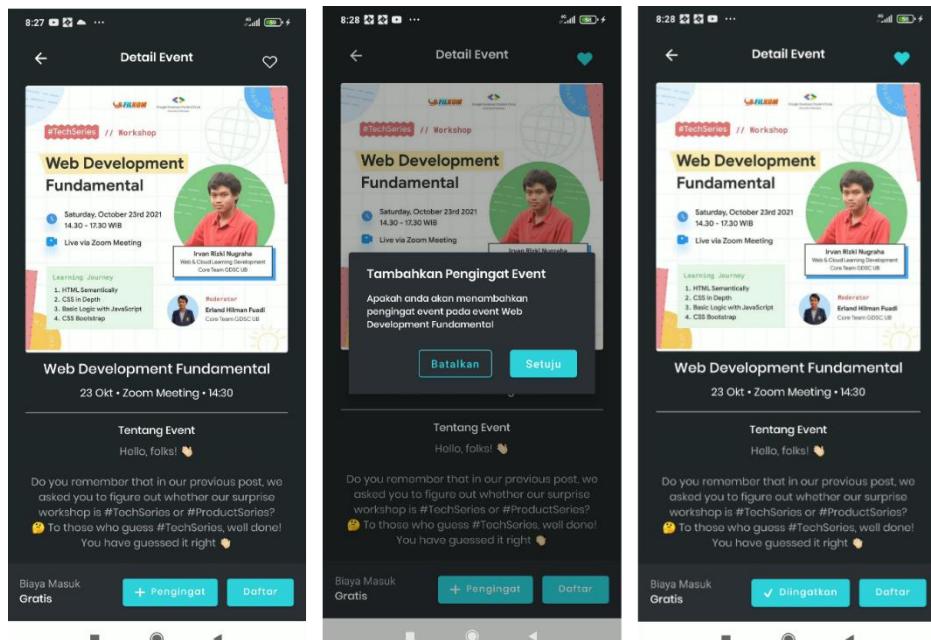
Implementasi tampilan untuk halaman *daftar event kompetisi IT* tertera pada Gambar 6.18



Gambar 6.18 Implementasi Tampilan Daftar Event Kompetisi IT

6.4.14 Implementasi Tampilan Detail Event Conference

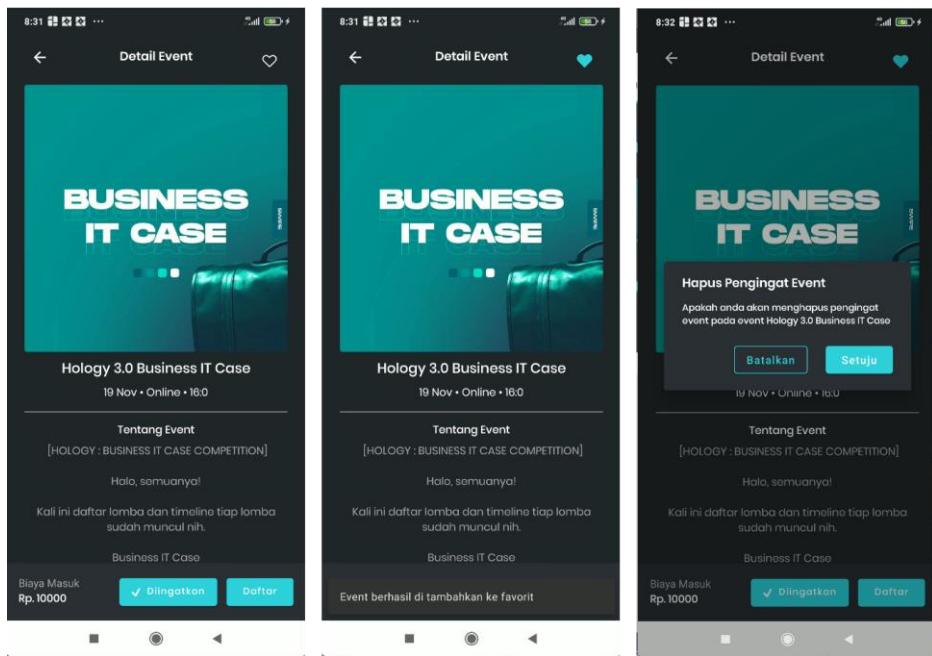
Implementasi tampilan untuk halaman detail *event conference* tertera pada Gambar 6.19



Gambar 6.19 Implementasi Tampilan Detail Event Conference

6.4.15 Implementasi Tampilan Detail Event Kompetisi IT

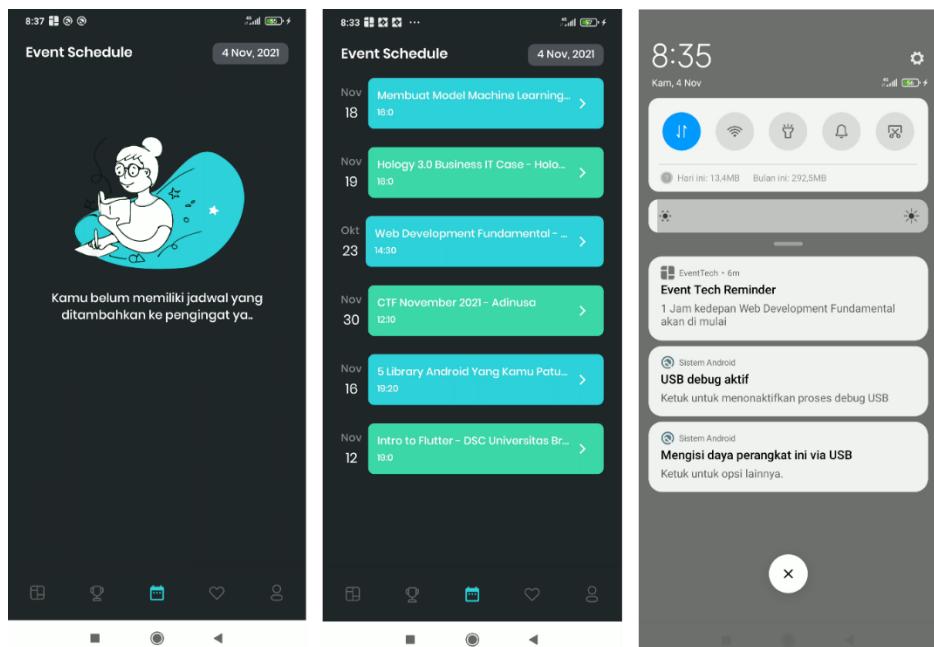
Implementasi tampilan untuk halaman detail *event* kompetisi IT tertera pada Gambar 6.20



Gambar 6.20 Implementasi Tampilan Detail Event Kompetisi IT

6.4.16 Implementasi Tampilan Jadwal Event

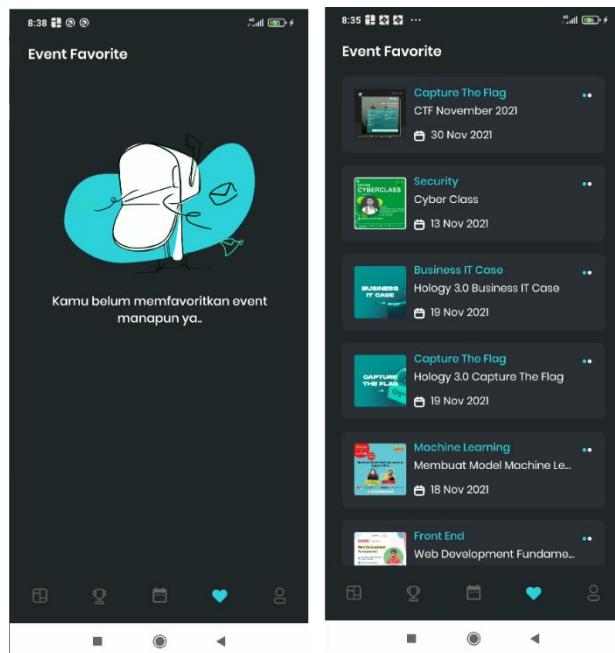
Implementasi tampilan untuk halaman jadwal *event* tertera pada Gambar 6.21



Gambar 6.21 Implementasi Tampilan Jadwal Event

6.4.17 Implementasi Tampilan Event Favorit

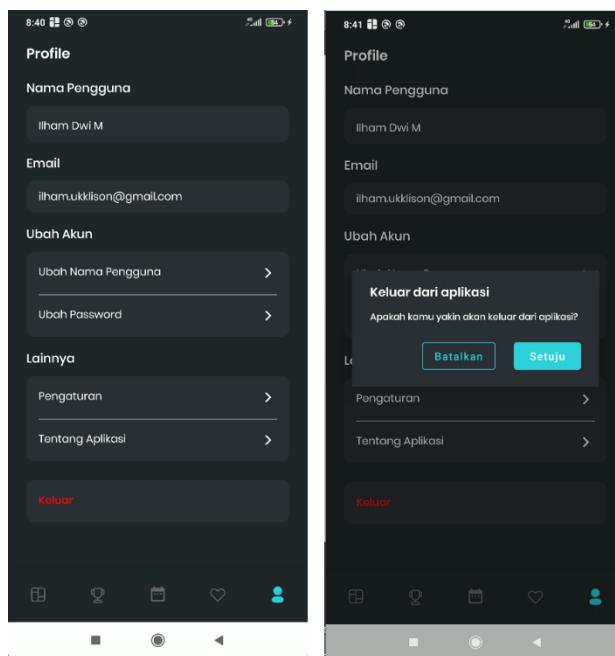
Implementasi antarmuka untuk halaman *event favorit* tertera pada Gambar 6.22



Gambar 6.22 Implementasi Tampilan Event Favorit

6.4.18 Implementasi Tampilan Profil Pencari Event

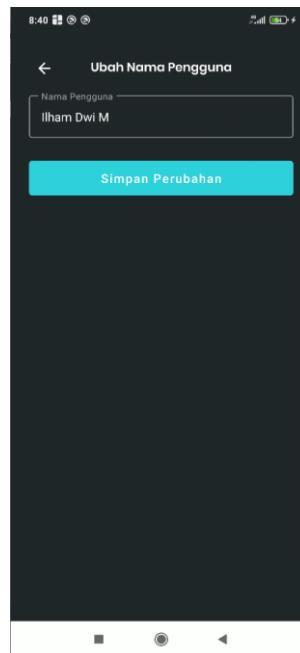
Implementasi tampilan untuk halaman profil pencari *event* tertera pada Gambar 6.23



Gambar 6.23 Implementasi Tampilan Profil Pencari Event

6.4.19 Implementasi Tampilan Ubah Nama Pengguna

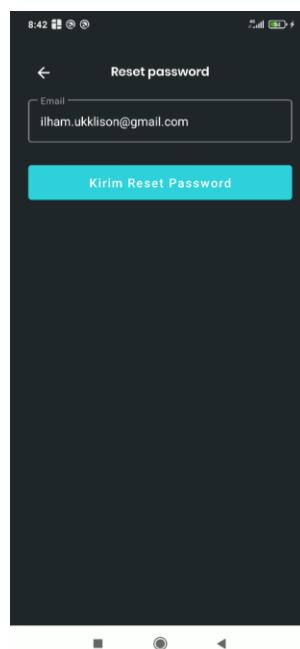
Implementasi tampilan untuk halaman ubah nama pengguna tertera pada Gambar 6.24



Gambar 6.24 Implementasi Tampilan Ubah Nama Pengguna

6.4.20 Implementasi Tampilan Ubah Password

Implementasi tampilan untuk halaman ubah *password* tertera pada Gambar 6.25.



Gambar 6.25 Implementasi Tampilan Ubah Password

6.4.21 Implementasi Tampilan Pengaturan

Implementasi tampilan untuk halaman pengaturan tertera pada Gambar 6.26.



Gambar 6.26 Implementasi Tampilan Pengaturan

6.4.22 Implementasi Tampilan Tentang Aplikasi

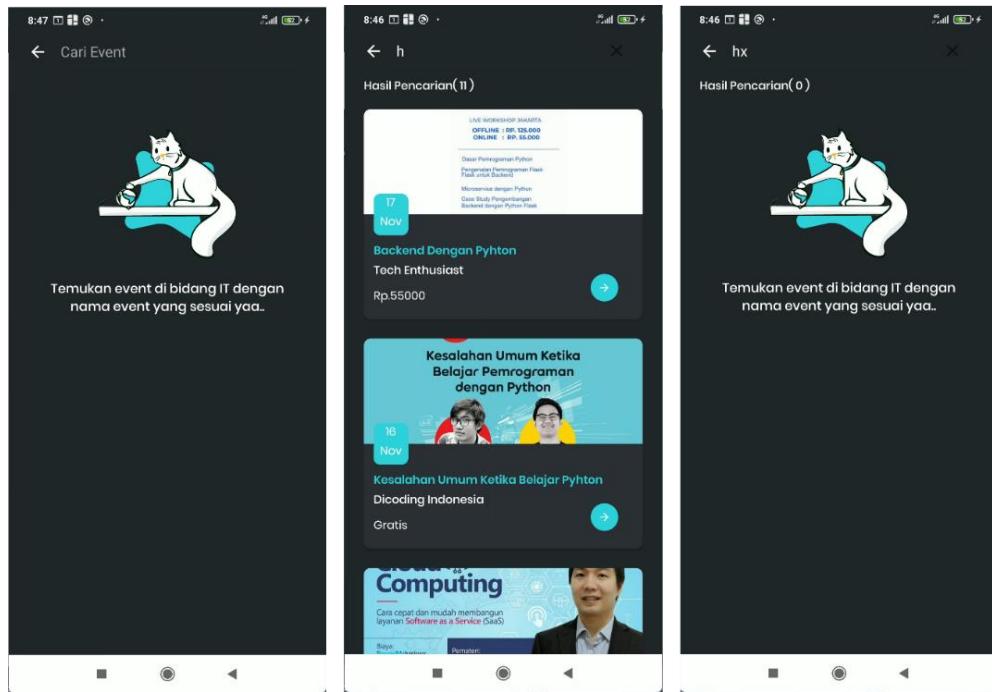
Implementasi tampilan untuk halaman tentang aplikasi tertera pada Gambar 6.27



Gambar 6.27 Implementasi Tampilan Tentang Aplikasi

6.4.23 Implementasi Tampilan Pencarian Event

Implementasi tampilan untuk halaman pencari event tertera pada Gambar 6.28



Gambar 6.28 Implementasi Tampilan Pencarian Event

BAB 7 PENGUJIAN

Bab pengujian berisi tentang hasil pengujian dari pengembangan perangkat lunak pada penelitian ini. Bab ini memuat penjabaran tentang pengujian validasi menggunakan *Black Box*, pengujian *usability* menggunakan *System Usability Scale*, pengujian *Compatibility*, dan hasil analisis dari pengujian.

7.1 Pengujian Validasi

Pada bagian ini dijelaskan tentang hasil dari pengujian validasi dengan menggunakan metode *black box*. Pengujian dilakukan berdasarkan kebutuhan fungsional yang terdapat pada sistem. Pengujian *black box* dapat dilihat pada Tabel 7.1 sampai dengan Tabel 7.28.

Tabel 7.1 Pengujian Black Box Registrasi Penyelenggara Event

Registrasi Penyelenggara Event (ETO-F-001)				
No.	Test Case	Expected Result	Result	Status
1.	Aktor mengisi nama penyelenggara, <i>email</i> dan <i>password</i>	Aktor masuk ke dalam sistem menjadi penyelenggara event	Aktor masuk ke dalam sistem menjadi penyelenggara event	Berhasil
2.	Aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan	Sistem menampilkan pesan untuk mengisi data	Sistem menampilkan pesan untuk mengisi data	Berhasil
3.	Aktor mendaftarkan <i>email</i> yang telah terdaftar	Sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan	Sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan	Berhasil

Tabel 7.2 Pengujian Black Box Login Penyelenggara Event

Login (ETO-F-002)				
No.	Test Case	Expected Result	Result	Status
1.	Aktor memasukan <i>email</i> dan <i>password</i> yang sesuai	Aktor dapat masuk ke dalam sistem	Aktor dapat masuk ke dalam sistem	Berhasil
2.	Aktor tidak mengisi salah satu atau	Sistem akan menampilkan	Sistem akan menampilkan	Berhasil

	seluruh data yang dibutuhkan	pesan untuk mengisi data	pesan untuk mengisi data	
3.	Aktor tidak memasukan <i>email</i> dan <i>password</i> yang tidak sesuai	Sistem maka akan menampilkan pesan masuk gagal	Sistem maka akan menampilkan pesan masuk gagal	Berhasil

Tabel 7.3 Pengujian Registrasi Pencari Event

Registrasi Pencari Event (ET-F-001)				
No.	Test Case	Expected Result	Result	Status
1.	Aktor mengisi nama penyelenggara, <i>email</i> dan <i>password</i>	Aktor masuk ke dalam sistem menjadi pencari event	Aktor masuk ke dalam sistem menjadi pencari event	Berhasil
2.	Aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan	Sistem menampilkan pesan untuk mengisi data	Sistem menampilkan pesan untuk mengisi data	Berhasil
3.	Aktor mendaftarkan <i>email</i> yang telah terdaftar	Sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan	Sistem akan menampilkan pesan bahwa <i>email</i> telah digunakan	Berhasil

Tabel 7.4 Pengujian Black Box Login Pencari Event

Login (ET-F-002)				
No.	Test Case	Expected Result	Result	Status
1.	Aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan	Sistem akan menampilkan pesan untuk mengisi data	Sistem akan menampilkan pesan untuk mengisi data	Berhasil
2.	Aktor tidak memasukan <i>email</i> dan <i>password</i> yang tidak sesuai	Sistem maka akan menampilkan pesan masuk gagal	Sistem maka akan menampilkan pesan masuk gagal	Berhasil

3.	Aktor tidak mengisi salah satu atau seluruh data yang dibutuhkan	Sistem akan menampilkan pesan untuk mengisi data	Sistem akan menampilkan pesan untuk mengisi data	Berhasil
----	--	--	--	----------

Tabel 7.5 Pengujian Black Box Membuat Event

Membuat Event (ETO-F-003)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event mengisi kelengkapan informasi dari event yang akan ditambahkan	Penyelenggara Event berhasil membuat event	Penyelenggara Event berhasil membuat event	Berhasil
2.	Penyelenggara Event tidak mengisi salah satu atau keseluruhan informasi dari event yang akan ditambahkan	Sistem menampilkan pesan untuk mengisi data	Sistem menampilkan pesan untuk mengisi data	Berhasil

Tabel 7.6 Pengujian Black Box Menyunting Event

Menyunting Event (ETO-F-004)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event mengganti field data informasi yang ingin diubah	Penyelenggara Event berhasil menyunting event	Penyelenggara Event berhasil menyunting event	Berhasil

Tabel 7.7 Pengujian Black Box Menghapus Event

Menghapus Event (ETO-F-005)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event memilih tombol hapus pada dialog	Penyelenggara Event berhasil menghapus event	Penyelenggara Event berhasil menghapus event	Berhasil

	yang muncul untuk konfirmasi		menghapus event	
2.	Penyelenggara Event memilih tombol batalkan pada dialog yang muncul untuk konfirmasi	Dialog konfirmasi menghilang dan event tidak dihapus	Dialog konfirmasi menghilang dan event tidak dihapus	Berhasil

Tabel 7.8 Pengujian Black Box Melihat Insight Dari Event

Melihat Insight Dari Event (ETO-F-006)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event masuk ke dalam halaman event insight	Penyelenggara Event melihat insight dari semua event milik aktor	Penyelenggara Event melihat insight dari semua event milik aktor	Berhasil

Tabel 7.9 Pengujian Black Box Melihat Daftar Event

Melihat Daftar Event (ETO-F-007)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event masuk ke dalam halaman beranda	Penyelenggara Event melihat daftar event yang telah dipublikasikannya	Penyelenggara Event melihat daftar event yang telah dipublikasikannya	Berhasil

Tabel 7.10 Pengujian Black Box Mengubah Nama Penyelenggara

Mengubah Nama Penyelenggara (ETO-F-008)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event mengisi nama pada field nama penyelenggara	Penyelenggara Event berhasil mengubah nama penyelenggara	Penyelenggara Event berhasil mengubah nama penyelenggara	Berhasil

--	--	--	--

Tabel 7.11 Pengujian Black Box Mengubah Password Penyelenggara Event

Mengubah password (ETO-F-009)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event mengisi <i>email</i> pada <i>field email</i> penyelenggara	Penyelenggara Event berhasil mendapatkan tautan <i>reset password</i> pada <i>emailnya</i>	Penyelenggara Event berhasil mendapatkan tautan <i>reset password</i> pada <i>emailnya</i>	Berhasil

Tabel 7.12 Pengujian Black Box Logout Penyelenggara Event

Logout (ETO-F-010)				
No.	Test Case	Expected Result	Result	Status
1.	Penyelenggara Event menekan tombol <i>logout</i>	Penyelenggara Event berhasil <i>logout dari sistem</i>	Penyelenggara Event berhasil <i>logout dari sistem</i>	Berhasil

Tabel 7.13 Pengujian Black Box Melihat Daftar Event Conference

Melihat Daftar Event Conference (ET-F-003)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event memilih kategori <i>event pada halaman event conference</i>	Pencari Event dapat melihat daftar <i>event</i> sesuai dengan kategori yang dipilih	Pencari Event dapat melihat daftar <i>event</i> sesuai dengan kategori yang dipilih	Berhasil

Tabel 7.14 Pengujian Black Box Melihat Daftar Event Kompetisi IT

Melihat Daftar Event Kompetisi IT (ET-F-004)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event memilih kategori	Pencari Event dapat melihat daftar <i>event</i> sesuai	Pencari Event dapat melihat daftar <i>event</i> sesuai dengan	Berhasil

	<i>event pada halaman event kompetisi IT</i>	dengan kategori yang dipilih	kategori yang dipilih	
--	--	------------------------------	-----------------------	--

Tabel 7.15 Pengujian Black Box Melihat Detail Event Conference

Melihat Detail Event Conference (ET-F-005)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event memilih <i>event</i> dari daftar <i>event</i> yang conference	Pencari Event melihat detail <i>event conference</i>	Pencari Event melihat detail <i>event conference</i>	Berhasil

Tabel 7.16 Pengujian Black Box Melihat Detail Event Kompetisi IT

Melihat Detail Event Kompetisi IT (ET-F-006)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event memilih <i>event</i> dari daftar <i>event</i> yang kompetisi IT	Pencari Event melihat detail <i>event kompetisi IT</i>	Pencari Event melihat detail <i>event kompetisi IT</i>	Berhasil

Tabel 7.17 Pengujian Black Box Menambahkan Notifikasi Pengingat

Menambahkan Notifikasi Pengingat (ET-F-007)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol setuju pada dialog konfirmasi menambahkan pengingat pada <i>event</i>	Pencari Event berhasil menambahkan notifikasi pengingat pada aplikasi	Pencari Event berhasil menambahkan notifikasi pengingat pada aplikasi	Berhasil
2.	Pencari Event menekan tombol batalkan pada dialog konfirmasi menambahkan	Sistem menutup tombol dialog dan Pencari Event tidak menambahkan	Sistem menutup tombol dialog dan Pencari Event tidak	Berhasil

	pengingat pada event	notifikasi pengingat pada aplikasi	menambahkan notifikasi pengingat pada aplikasi	
--	----------------------	------------------------------------	--	--

Tabel 7.18 Pengujian Black Box Menghapus Notifikasi Pengingat

Menghapus Notifikasi Pengingat (ET-F-008)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol setuju pada dialog konfirmasi menghapus pengingat pada event	Pencari Event berhasil menghapus notifikasi pengingat pada aplikasi	Pencari Event berhasil menghapus notifikasi pengingat pada aplikasi	Berhasil
2.	Pencari Event menekan tombol batalkan pada dialog konfirmasi menghapus pengingat pada event	Sistem menutup tombol dialog dan Pencari Event tidak menghapus notifikasi pengingat pada aplikasi	Sistem menutup tombol dialog dan Pencari Event tidak menghapus notifikasi pengingat pada aplikasi	Berhasil

Tabel 7.19 Pengujian Black Box x Mendaftar Event

Mendaftar Event (ET-F-009)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol daftar pada halaman detail	Pencari Event berada pada halaman webview yang berisi pendaftaran yang disediakan oleh penyelenggara event	Pencari Event berada pada halaman webview yang berisi pendaftaran yang disediakan oleh penyelenggara event	Berhasil

Tabel 7.20 Pengujian Black Box Melihat Jadwal Event

Melihat Jadwal Event (ET-F-010)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event masuk ke dalam halaman jadwal event	Pencari Event melihat jadwal event yang telah ditambahkan pengingat	Pencari Event melihat jadwal event yang telah ditambahkan pengingat	Berhasil

Tabel 7.21 Pengujian Black Box Menambahkan Favorit Event

Menambahkan Favorit Event (ET-F-011)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol favorit pada halaman detail	Sistem menambahkan favorit pada event	Sistem menambahkan favorit pada event	Berhasil

Tabel 7.22 Pengujian Black Box Menghapus favorit event

Menghapus Favorit Event (ET-F-012)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol favorit pada halaman detail yang sudah ditambahkan favorit	Sistem menghapus favorit pada event	Sistem menghapus favorit pada event	Berhasil

Tabel 7.23 Pengujian Black Box Melihat Daftar Favorit Event

Melihat Daftar Favorit Event (ET-F-013)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event masuk ke halaman daftar favorit event	Pencari Event melihat semua daftar event yang telah ditambahkan favorit	Pencari Event melihat semua daftar event yang telah ditambahkan favorit	Berhasil

			tambahkan favorit	
--	--	--	-------------------	--

Tabel 7.24 Pengujian Black Box Mengubah Nama Pengguna

Mengubah Nama Pengguna (ET-F-014)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event mengisi nama pada field nama pengguna	Pencari Event berhasil mengubah nama pengguna	Pencari Event berhasil mengubah nama pengguna	Berhasil

Tabel 7.25 Pengujian Black Box Mengubah Password Pencari Event

Mengubah Password (ET-F-015)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event mengisi email pada field email penyelenggara	Pencari Event berhasil mendapatkan tautan reset password pada emailnya	Pencari Event berhasil mendapatkan tautan reset password pada emailnya	Berhasil

Tabel 7.26 Pengujian Black Box Logout Pencari Event

Logout (ET-F-016)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event menekan tombol logout	Pencari Event berhasil logout dari sistem	Pencari Event berhasil logout dari sistem	Berhasil

Tabel 7.27 Pengujian Black Box Melakukan Pencarian Event Conference

Melakukan Pencarian Event Conference (ET-F-017)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event mengisi kolom pencarian dengan	Pencari Event dapat melihat daftar event yang memiliki kata kunci pencarian yang	Pencari Event dapat melihat daftar event yang memiliki kata kunci yang	Berhasil

	kata kunci yang benar	sesuai dengan yang diisi oleh aktor	sesuai dengan yang diisi oleh aktor	
2.	Pencari Event mengisi kolom pencarian dengan kata kunci yang salah	Sistem menampilkan informasi pencarian tidak ditemukan	Sistem menampilkan informasi pencarian tidak ditemukan	Berhasil

Tabel 7.28 Pengujian Black Box Melakukan Pencarian Event Kompetisi IT

Melakukan Pencarian Event Kompetisi IT (ET-F-018)				
No.	Test Case	Expected Result	Result	Status
1.	Pencari Event mengisi kolom pencarian dengan kata kunci yang benar	Pencari Event dapat melihat daftar event yang memiliki kata kunci pencarian yang sesuai dengan yang diisi oleh aktor	Pencari Event dapat melihat daftar event yang memiliki kata kunci yang sesuai dengan yang diisi oleh aktor	Berhasil
2.	Pencari Event mengisi kolom pencarian dengan kata kunci yang salah	Sistem menampilkan informasi pencarian tidak ditemukan	Sistem menampilkan informasi pencarian tidak ditemukan	Berhasil

7.2 Pengujian Usability

Pada bagian ini pengujian *usability* dibagi menjadi dua jenis pengujian, yaitu pengujian *usability* pada aplikasi pencari *event* dan pengujian *usability* pada aplikasi penyelenggara *event*. Selain itu, ditambahkan juga *confidence interval* pada hasil rata-rata jumlah nilai *usability*. *Confidence interval* digunakan untuk menentukan keakuratan dari rata-rata dan memprediksi nilai rata-rata yang aktual untuk pengukuran rata-rata (Ilham,2020). Jumlah data *sample* juga mempengaruhi nilai rata-rata yang ditambahkan *confidence interval*.

7.2.1 Pengujian *Usability* Pencari Event

Pengujian untuk aplikasi pencari event dilakukan pada 10 orang pencari event dibidang teknologi informasi. Tabel untuk rekapitulasi nilai pengisian kuesioner oleh responden pencari event dapat dilihat pada Tabel 7.29.

Tabel 7.29 Rekapitulasi Nilai Kuesioner Responden Pencari Event

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1.	Responden pertama	5	2	4	1	4	1	4	2	4	3
2.	Responden kedua	4	1	5	1	5	2	4	1	3	1
3.	Responden ketiga	4	2	4	3	5	2	3	2	4	2
4.	Responden keempat	5	2	5	1	4	1	3	2	5	1
5.	Responden kelima	4	2	5	3	5	2	4	2	5	3
6.	Responden keenam	5	2	4	1	4	1	4	2	4	2
7.	Responden ketujuh	5	1	4	2	5	1	4	1	5	3
8.	Responden kedelapan	5	2	5	1	4	2	5	1	5	1
9.	Responden kesembilan	4	1	5	1	5	1	5	1	5	1
10.	Responden kesepuluh	5	1	5	1	5	1	5	1	5	1

Nilai kuesioner responden pencari event yang didapat dari pengisian 10 pertanyaan *System Usability Scale* akan diolah menjadi hasil hitung data. Setiap nilai jawaban dari bermotor ganjil nilai akan dikurangkan dengan angka sebesar 1 dan pada setiap soal bertipe genap nilai diambil dari angka 5 dikurangi dengan nilai kuesioner. Hasil pengolahan nilai dapat dilihat pada Tabel 7.30.

Tabel 7.30 Rekapitulasi Nilai SUS Responden Pencari Event

No	Responden	Total Hasil Hitung Data	Usability Score
1.	Responden pertama	32	80
2.	Responden kedua	35	87,5

3.	Responden ketiga	29	72,5
4.	Responden keempat	35	87,5
5.	Responden kelima	31	77,5
6.	Responden keenam	33	82,5
7.	Responden ketujuh	35	87,5
8.	Responden kedelapan	37	92,5
9.	Responden kesembilan	39	97,5
10.	Responden kesepuluh	40	100
Total score			865
Rata-rata score			86,5
Confidence Interval ($\alpha=0,1$)			4,5

Total *score* yang didapat dari responden pencari *event* adalah 865, total score ini berasal dari jumlah setiap total hasil hitung data dikalikan dengan 2,5. Rata-rata yang terdapat pada pengujian SUS dari responden pencari *event* adalah 86,5. Selain itu, penulis menambahkan *confidence interval* pada pengujian dengan alpha level 10% dan didapatkan nilai 4,5. Nilai aktual rata-rata SUS adalah $86,5 \pm 4,5$ atau 82 sampai dengan 92.

7.2.2 Pengujian *Usability* Penyelenggara *Event*

Pengujian untuk aplikasi penyelenggara *event* dilakukan pada 5 orang penyelenggara *event* dibidang teknologi informasi. Tabel untuk rekapitulasi nilai pengisian kuesioner oleh responden penyelenggara *event* dapat dilihat pada Tabel 7.31.

Tabel 7.31 Rekapitulasi Nilai Kuesioner Responden Penyelenggara *Event*

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1.	Responden pertama	5	2	4	1	5	1	5	1	5	2
2.	Responden kedua	5	1	5	2	5	1	5	1	5	2
3.	Responden ketiga	5	1	5	1	5	1	5	1	1	2
4.	Responden keempat	4	2	5	1	5	1	4	1	5	1
5.	Responden kelima	4	2	5	3	4	1	5	2	4	3

Nilai kuesioner responden penyelenggara *event* yang didapat dari pengisian 10 pertanyaan *System Usability Scale* akan diolah menjadi hasil hitung data. Setiap nilai jawaban dari bernomor ganjil nilai akan dikurangkan dengan angka sebesar 1 dan pada setiap soal bertipe genap nilai diambil dari angka 5 dikurangi dengan nilai kuesioner. Hasil pengolahan nilai dapat dilihat pada Tabel 7.32.

Tabel 7.32 Rekapitulasi Nilai SUS Responden Penyelenggara Event

No	Responden	Total Hasil Hitung Data	Usability Score
1.	Responden pertama	37	92,5
2.	Responden kedua	38	95
3.	Responden ketiga	35	87,5
4.	Responden keempat	37	92,5
5.	Responden kelima	31	77,5
Total score			445
Rata-rata score			89
Confidence Interval ($\alpha = 0,1$)			5,1

Total *score* yang didapat dari responden penyelenggara *event* adalah 445, total *score* ini berasal dari jumlah setiap total hasil hitung data dikalikan dengan 2,5. Rata-rata yang terdapat pada pengujian SUS dari responden penyelenggara *event* adalah 89. Rata-rata *score* 89 termasuk dalam predikat baik. Selain itu, penulis menambahkan *confidence interval* pada pengujian dengan alpha level 10% dan didapatkan nilai 5,1. Nilai aktual rata-rata SUS pada pengujian *usability* aplikasi penyelenggara *event* adalah $89 \pm 5,1$ atau 83,9 sampai dengan 94,1.

7.3 Pengujian *Compatibility*

Pada pengujian *compatibility* aplikasi yang telah selesai dikembangkan dijalankan pada beberapa perangkat untuk melihat kompatibilitas dari aplikasi dengan beberapa versi SDK pada sistem operasi Android. Pengujian dilakukan pada sistem operasi Android dengan SDK 23 sampai dengan 30. Hasil pengujian *compatibility* pada beberapa perangkat dapat dilihat pada Tabel 7.35.

Tabel 7.33 Hasil Pengujian Compatibility

No	Perangkat	SDK	Hasil
1.	Google Pixel 3	23	Aplikasi kompatibel pada perangkat
2.	Zenfone Max Plus M1	24	Aplikasi kompatibel pada perangkat
3.	Google Pixel 3XL	25	Aplikasi kompatibel pada perangkat
4.	Google Pixel 4	26	Aplikasi kompatibel pada perangkat

5.	Google Pixel 4XL	27	Aplikasi kompatibel pada perangkat
6.	Redmi Note 8	28	Aplikasi kompatibel pada perangkat
7.	Redmi Note 7	29	Aplikasi kompatibel pada perangkat
8.	Vivo 1904	30	Aplikasi kompatibel pada perangkat

BAB 8 KESIMPULAN DAN SARAN

Pada bab ini dituliskan rangkuman dari hasil penelitian yang terdapat pada penelitian ini. Rangkuman dari hasil penelitian dijabarkan ke dalam kesimpulan dan saran.

8.1 Kesimpulan

Berdasarkan penelitian tentang pengembangan aplikasi perangkat bergerak sistem informasi *event* di bidang teknologi informasi berbasis Android diperoleh beberapa kesimpulan sebagai berikut:

1. Berdasarkan analisis kebutuhan pada bab analisis kebutuhan didapatkan kebutuhan fungsional untuk penyelenggara *event* sebanyak 10 kebutuhan fungsional dan 19 kebutuhan fungsional untuk pencari *event*, sedangkan untuk kebutuhan non fungsional didapatkan sebanyak 2 kebutuhan. Kebutuhan fungsional yang terdapat pada bab analisis juga dijabarkan ke dalam bentuk *use case diagram* dan *use case scenario*. Kebutuhan fungsional yang dari penyelenggara *event* antara lain adalah registrasi, *login*, membuat *event*, menyunting *event*, menghapus *event*, melihat *insight* dari *event*, melihat melihat daftar *event*, mengubah nama penyelenggara, mengubah *password* dan *logout*. Pada pencari *event* kebutuhan fungsional antara lain melihat daftar *event conference*, melihat daftar *event* kompetisi IT, melihat detail *event conference*, melihat detail *event* kompetisi IT, menambahkan notifikasi pengingat, menghapus notifikasi pengingat, mendaftar *event*, melihat jadwal *event*, menambahkan favorit, melihat favorit *event*, menghapus favorit *event*, melihat daftar favorit *event*, mengubah nama pengguna, mengubah *password*, *logout*, melakukan pencarian *event conference*, melakukan pencarian *event* kompetisi IT. Kebutuhan non fungsional yang terdapat pada penelitian ini adalah *compatibility* dan *usability*. *Compatibility* digunakan untuk melihat kecocokan aplikasi dengan versi pada beberapa sistem operasi Android dan *usability* digunakan untuk memeriksa fungsionalitas dari aplikasi.
2. Berdasarkan pengujian pada bab sebelumnya telah didapatkan hasil untuk pengujian validasi dengan menggunakan *black box* sebesar 100% berhasil. Pengujian *usability* dilakukan dengan menggunakan *System Usability Scale* dan didapatkan nilai rata-rata aktual sebesar $86,5 \pm 4,5$ dengan nilai alpha level sebesar 10% untuk aplikasi pencari *event* dan nilai rata-rata aktual sebesar $89 \pm 5,1$ dengan nilai alpha level sebesar 10% untuk aplikasi penyelenggara *event* yang berarti kedua aplikasi masuk dalam kategori *best imaginable* dan memiliki predikat *acceptable*. Selain itu, terdapat pula pengujian compatibility dengan hasil perangkat lunak dapat digunakan pada sistem operasi Android dengan versi SDK 23 sampai dengan SDK 30.

8.2 Saran

Saran yang dapat penulis berikan kepada peneliti selanjutnya yang ingin melanjutkan penelitian adalah sebagai berikut:

1. Menambahkan fitur *chat* antara penyelenggara *event* dan pencari *event* untuk mendapatkan informasi apabila terdapat informasi yang masih belum dipahami oleh pencari *event*.
2. Menambahkan opsi fitur daftar *event* secara langsung pada aplikasi secara langsung bukan berasal dari tautan yang disediakan oleh penyelenggara *event* seperti yang terdapat pada aplikasi saat ini.
3. Mengembangkan penelitian ini dengan menggunakan platform lain seperti perangkat bergerak dengan sistem operasi iOS dan platform berbasis website.

DAFTAR REFERENSI

- Ariani S., R. & Shalahuddin, M., 2018. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.
- Aulia Aziiza, A. dan Nur Fadhilah, A., 2020. Analisis Metode Identifikasi dan Verifikasi Kebutuhan Non Fungsional. *Applied Technology and Computing Science Journal*, 3(1), hal.13–21.
- Bangor, A., Kortum, P. and Miller, J., 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, pp.114–123.
- Dwanoko, Y.S., 2016. Implementasi Software Development Life Cycle (Sdlc) Dalam Penerapan Pembangunan Aplikasi Perangkat. *Jurnal Teknologi Informasi*, 7(2), hal.83–94.
- Effendi, F. R. 2013. Android Sistem Operasi pada Smartphone. Universitas Surabaya, Surabaya.
- Farouqi, M.I., Aknuranda, I. dan Herlambang, A.D., 2018. Evaluasi Usability pada Aplikasi Go-Jek Dengan Menggunakan Metode Pengujian Usability. 2(9), hal.3110–3117.
- Firmansyah, H. dan Arnie, R., 2017. Model Sistem Informasi Promosi Dan Management Event Berbasis Web. *Jutisi*, 6(2), hal.1547–1558.
- Google Android Developer, 2021. *Platform architecture : Android developers*. [online] Android Developers. Available at: <https://developer.android.com/guide/platform> [Diakses 29 Juli. 2021].
- Google Android Developer, 2021. *Meet android studio : Android developers*. [online] Android Developers. Available at: <https://developer.android.com/studio/intro> [Diakses 29 Juli. 2021].
- H.N, I.A., Nugroho, P.I. dan Ferdiana, R., 2015. Pengujian Usability Website Menggunakan System Usability Scale. *JURNAL IPTEKKOM : Jurnal Ilmu Pengetahuan & Teknologi Informasi*, 17(1), hal.31.
- Hasibuan, D.P., Santoso, H.B., Yunita, A. and Rahmah, A., 2020. An Indonesian adaptation of the E-Learning Usability Scale. *Journal of Physics: Conference Series*, 1566, p.012051.
- Ilham, 2020. *Cara Menghitung confidence interval (CI) (interval Kepercayaan)*. [online] LABMUTU. Available at: <<https://www.labmutu.com/2020/08/confidence-interval.html>> [Diakses 10 Nov. 2021].

Irwansyah, E. dan Moniaga, J. V, 2014. *Pengantar Teknologi Informasi*. Yogyakarta: Deepublish.

Jeevuz, 2017. [online] *Misconceptions clean architecture*. Available at: <<https://discourse.world/h/2017/08/11/Misconceptions-Clean-Architecture>> [Diakses 16 Agustus. 2021].

Julianto, R., 2020. *Apa Itu firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya*. [online] Dicoding Blog. Available at: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya> [Diakses 29 Juli. 2021].

Lingga , T.I.A., 2021. Pengembangan Sistem Informasi Event Seminar Dan Workshop(EVENTIFY). S2. Universitas Syiah Kuala

Maulidia, S., 2019. bpptik.kominfo. [Online] Available at: <https://bpptik.kominfo.go.id/2020/07/09/8009/kebutuhan-sumber-daya-manusia-bidang-tik-di-era-revolusi-industri-4-0/> [Diakses 18 Juli 2021]

Martin, R.C., 2018. *Clean architecture: A craftsman's guide to software structure and Design*. Prentice Hall.

Martoko., 2019. Rancang bangun Sistem Informasi Manajemen Publikasi Event Berbasis Android. S2. Universitas Pesantren Tinggi Darul 'Ulum.

Nilakandi, Z., 2020. Pengertian pseudocode beserta Fungsi, Struktur Dan Contohnya. [online] Nesabamedia. Available at: <<https://www.nesabamedia.com/pengertian-pseudocode/>> [Accessed 30 Sep. 2021].

Noor, A., 2009. *Manajemen Event*. Bandung: alfabeta.

Rahmatsyah, 2019. Pengembangan Aplikasi Perangkat Bergerak Layanan berbasis Lokasi Penghubung Sukarelawan dengan Kegiatan Sukarela. 3(10), hal.9859–9866.

Riadi, M., 2019. kajianpustaka.com. [Online] Available at: <https://www.kajianpustaka.com/2019/04/manajemen-event.html> [Diakses 18 Juli 2021]

Setiyani, L. dan Tjandra, E., 2021. Analisis Kebutuhan Fungsional Aplikasi Penanganan Keluhan Mahasiswa Studi Kasus: STMIK Rosma Karawang. *Jurnal Inovasi Pendidikan dan Teknologi Informasi (JIPTI)*, 2(1), hal.105–114.

Sondha, A.T., Sa, U., Hardiansyah, F.F., Bagus, M. dan Rasyid, A., 2020. Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture (Framework and Code Generator for Android

Development with Clean Architecture Principles Implementation). *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 9(4), hal.327–335.

Statcounter, 2021. *Mobile operating system market share indonesia*. [online] StatCounter Global Stats. Available at: <https://gs.statcounter.com/os-market-share/mobile/indonesia> [Diakses 18 Juli 2021].

Stonehem, B., 2016. *Google Android Firebase : Learning the Basics*. Ke1 penyunt.Bengalu:First Rank Publication.

Tolle, H., Pinandito, A., Kharisma, A.P. dan Dewi, R.K., 2017. *Pengembangan Aplikasi Perangkat Bergerak*. Malang: UB Press.

LAMPIRAN A WAWANCARA

A.1 Wawancara Tentang Penyelenggaraan *Event*

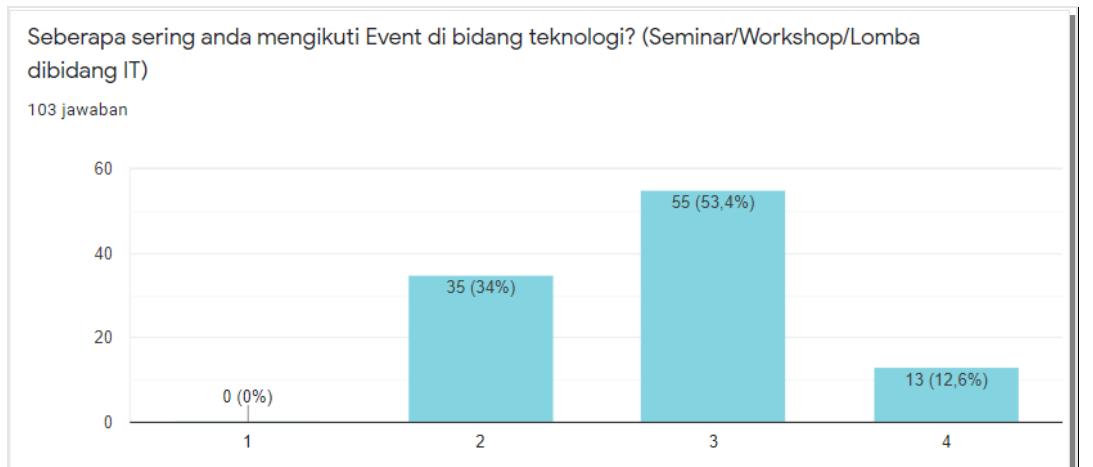
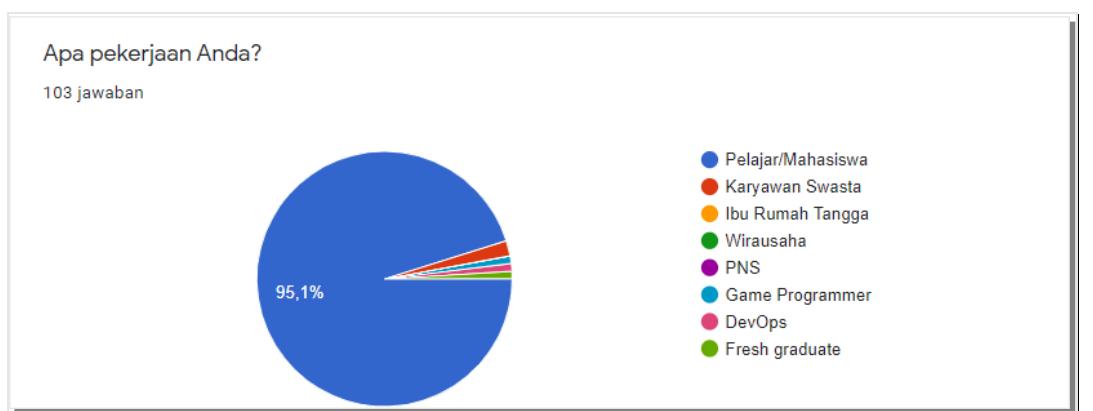
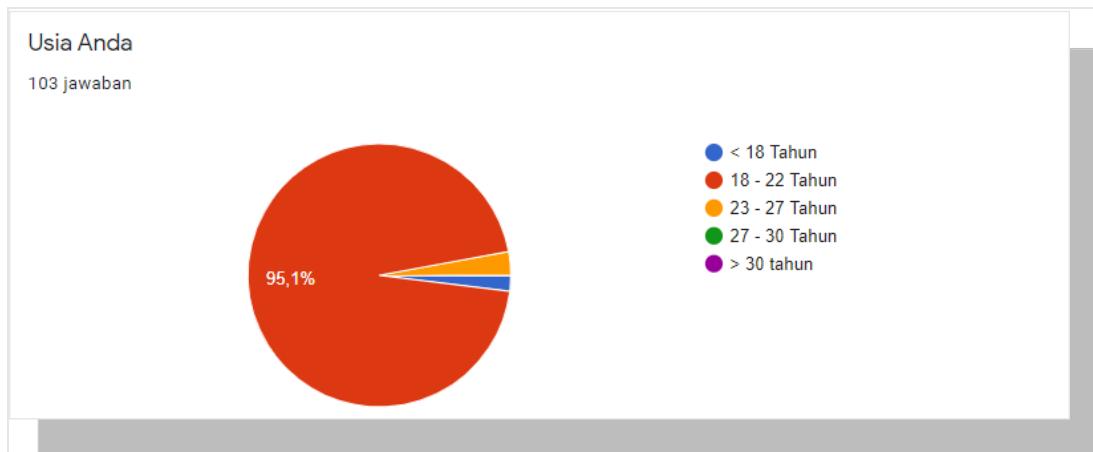
- Narasumber: Muhammad Irfan Yusuf
- Tahun: 2021
- Judul: Wawancara Tentang Penyelenggaraan *Event*
- Diwawancara oleh: Ilham Dwi Muchlison
- Teknis wawancara: Wawancara melalui *instant messenger*
- Tanggal Wawancara: 16 Juli

Pernyataan	Jawaban
Boleh memperkenalkan diri dan peran dalam <i>event</i> dibidang IT?	Nama saya Muhammad Irfan Yusuf, Pernah menjadi ketua pelaksana Hology 3.0.
<i>Event</i> yang ada di Hology ada apa saja?	Di hology ada 3 rangkaian acara. Semua acara bertemakan IT. Terdapat webinar, kompetisi kompetisi yang berkaitan dengan IT, dan academy.
Apakah ada kendala dalam penyebaran Informasi tentang <i>Event</i> yang mas adakan?	Kendalanya, sulit mencari media partner yang bisa mencakup banyak audiens yang menyukai <i>event</i> di bidang teknologi informasi.
Bagaimana cara mas menyebarkan informasi <i>Event</i> tersebut?	Penyebaran informasi pada hology melalui sosial media, media partner, sampai pengiklanan digital
Permasalahan yang dihadapi selama sebelum acara dan saat acara?	Kuota target pendaftaran ada yg kurang memenuhi target. Kurangnya antusias dalam mengikuti acara (karena <i>event</i> berlangsung secara online)
Apakah terdapat peserta yang mendaftar namun tidak mendatangi <i>eventnya</i> ?	Ada pada <i>event</i> saya namun tidak terlalu banyak, kalau di acara acara lain yang saya tahu banyak.

<p>Menurut mas selaku orang yang pernah menyelenggarakan <i>Event IT</i> Apabila ada aplikasi untuk memberikan informasi <i>Event IT</i> (Workshop/Seminar/Lomba) berdasarkan kategori kategori yang terdapat fitur Calender Daftar <i>Event</i> yang anda inginkan dan Reminder ketika sudah mendekati <i>Event</i>, akan berguna bagi peserta dan juga penyelenggara?</p>	<p>Untuk sumber informasi <i>event IT</i> saya rasa cukup berguna. Untuk reminder pada beberapa <i>event</i> yang baik sebenarnya akan memberikan reminder melalui <i>email</i> atau kontak yg sudah dikumpulkan saat pendaftaran.</p>
<p>Apakah ada kritik dan saran untuk pengembangan saya?</p>	<p>Lebih bagus lagi kalau ada fasilitas untuk penyelenggara mempromosikan <i>event</i> nya pada aplikasi tersebut</p>

LAMPIRAN B KUESIONER

B.1 Hasil Kuesioner Kepada Peserta *Event Teknologi Informasi*



Darimanakah anda mendapatkan informasi Event IT yang ingin anda ikuti?

103 jawaban

Teman, IG dan komunitas

Instagram, line, website, dll

Instagram ads, instagram acc dari suatu organisasi

Sosial media atau kadang dari teman

Instagram, YouTube, Teman, Komunitas/Organisasi

Instagram, OA Kampus, Grup, Email

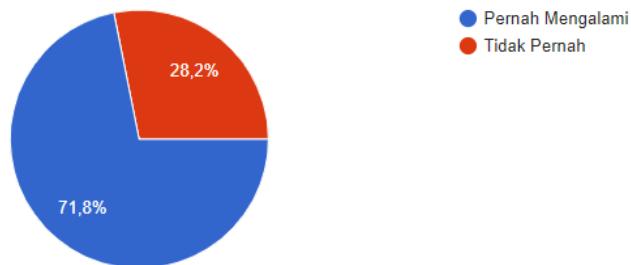
media sosial

Social Media, website official, teman dekat

Grup Whatsapp, Telegram, Instagram

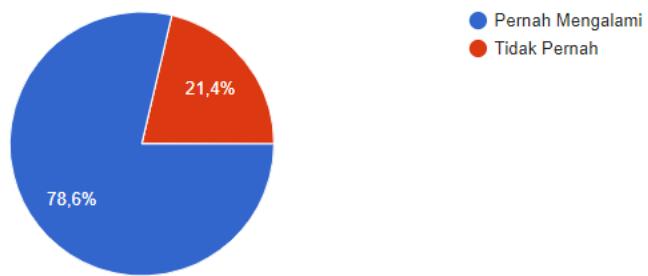
Pernahkah anda kesulitan mendapatkan informasi Event IT dengan topik yang ingin anda ikuti? (contoh : Workshop/Lomba untuk bidang security)

103 jawaban



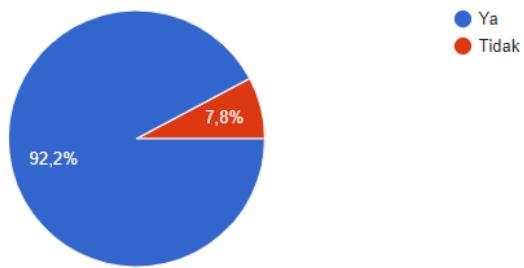
Pernahkan anda melewatkkan Event IT yang ingin anda datangi sebelumnya karena terlupa?

103 jawaban



Apabila ada aplikasi untuk memberikan informasi Event IT (Workshop/Seminar/Lomba) berdasarkan kategori kategori yang terdapat fitur Calender Daftar Event yang anda inginkan dan Reminder ketika sudah mendekati Event, akan berguna bagi anda ?

103 jawaban



Apakah Ada Permasalahan lainnya yang anda alami ketika hendak mengikuti Event IT?

103 jawaban

Jadwal event nya bertabrakan dengan event yang lain

Kadang jumlahnya sedikit

Tidak ada ,hanya sering melewatkkan event tersebut karena lupa

kualifikasi minimal yang tidak dicantumkan diawal informasi event

Kadang ga relate sma judul

Udah daftar lupa datang ke eventnya

ketika lombanya ada beberapa dan di waktu yang dekat kadang sulit mau menentukan ikut yang mana

lupa

Prerequirement untuk mengikuti event kurang jelas

LAMPIRAN C HASIL EVALUASI PROTOTYPE

C.1 Hasil Evaluasi *Prototype* Pertama Kepada Pencari Event

Hasil Evaluasi Prototype	
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari event Design to code, event tersebut berhubungan dengan pengembangan aplikasi mobile	Terdapat kendala dalam menjalankan <i>test scenario</i>
Coba berikan favorit kepada event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal event yang telah kamu tambahkan pengingat Coba buka design to code dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
<ol style="list-style-type: none">Pengingatnya untuk jadwal event 1 hari sebelumnyaTambahkan bulan pada jadwal eventTambahkan navigasi jadwal event ke detail	

- | |
|--|
| d. Tambah pencarian untuk mempermudah pengguna mencari event |
|--|

Hasil Evaluasi Prototype	
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari event Design to code, event tersebut berhubungan dengan pengembangan aplikasi mobile	Terdapat kendala dalam menjalankan <i>test scenario</i>
Coba berikan favorit kepada event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal event yang telah kamu tambahkan pengingat Coba buka design to code dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
a. Pengingatnya untuk jadwal event 6 jam sebelumnya b. Menambahkan <i>Light Mode</i> c. Menambahkan bulan di jadwal event d. Menambahkan search untuk mempermudah dan memberi tahu teman	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Ifadhatul Khoiroh
Waktu Pelaksanaan	30 Agustus 2021
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event Design to code</i> , <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka <i>design to code</i> dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
a. Pengingatnya untuk jadwal <i>event</i> 15 atau 30 menit sebelumnya b. Tambahkan fitur search untuk mencari judul <i>event</i>	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Munirottul Millah
Waktu Pelaksanaan	30 Agustus 2021
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event Design to code</i> , <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka <i>design to code</i> dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
a. Namanya ditambahin kategori <i>event conference</i> pada halaman conference supaya lebih memperjelas b. <i>Button</i> pengingat di tambahkan <i>post condition</i> c. <i>Button</i> daftar di tambahkan <i>post condition</i> d. Di tambahkan snackbar untuk <i>post condition</i> e. Informasi jam lokasi lebih dibuat terlihat pada halaman detail	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Lina Arifin
Waktu Pelaksanaan	30 Agustus 2021
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event Design to code</i> , <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Terdapat kendala dalam menjalankan <i>test scenario</i>
Coba berikan favorit kepada <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka <i>design to code</i> dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
a. Di tambahkan fitur untuk pencarian b. Di tambahkan filter tag c. Memperbaiki padding dan header jarak yang kepotong d. Menambahkan navigasi dari halaman jadwal <i>event</i> e. Di tambahkan keterangan pada halaman jadwal <i>event</i> hari ini masuk hari apa f. Pengingat 1 jam atau 30 menit sebelumnya	

C.2 Hasil Evaluasi *Prototype* Pertama Penyelenggara *Event*

Hasil Evaluasi Prototype	
<i>Test Scenario</i>	Hasil Test
Silahkan untuk membuat akun baru dan masuk ke halaman beranda	Berhasil melakukan sesuai <i>test scenario</i>
Buat <i>Event</i> baru untuk kategori competition	Berhasil melakukan sesuai <i>test scenario</i>
Sunting <i>event</i> Hology Game Development	Berhasil melakukan sesuai <i>test scenario</i>
Hapus <i>event</i> Hology Game development	Berhasil melakukan sesuai <i>test scenario</i>
Cari <i>insight</i> dari <i>event</i> yang telah dipublish	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
a. Dari segi <i>user experience</i> dapat dipahami b. Add <i>event</i> bisa memakai FAB dan label nama c. Halaman <i>insight</i> di tambahkan dapat dilihat jumlah yang <i>event</i> yang telah dilihat	

C.3 Hasil Evaluasi Prototype Kedua Pencari Event

Hasil Evaluasi Prototype	
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari event Design to code, event tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar event design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal event yang telah kamu tambahkan pengingat Coba buka design to code dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
Sudah cukup dan tidak ada saran	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Alfirsya Dhamasyifa
Waktu Pelaksanaan	7 September 2021

Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event Design to code</i> , <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka <i>design to code</i> dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
Sudah cukup dan tidak ada saran	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Ifadhatul Khoiroh
Waktu Pelaksanaan	7 September 2021
Test Scenario	Hasil Test

Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event</i> Design to code, <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka design to code dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
Sudah cukup dan tidak ada saran	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Munirottul Millah
Waktu Pelaksanaan	7 September 2021
<i>Test Scenario</i>	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>

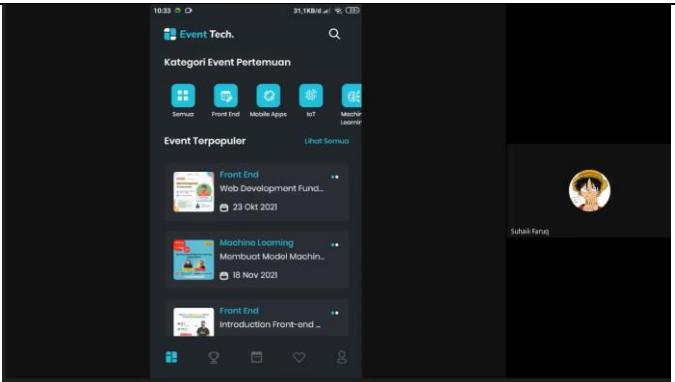
Coba cari <i>event</i> Design to code, <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>
Coba berikan favorit kepada <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event</i> design to code	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka design to code dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
Sudah cukup dan tidak ada saran	

Hasil Evaluasi Prototype	
Nama pengamat	Ilham Dwi Muchlison
Nama pengguna	Lina Arifin
Waktu Pelaksanaan	7 September 2021
Test Scenario	Hasil Test
Coba masuk ke halaman beranda dari halaman login	Berhasil melakukan sesuai <i>test scenario</i>
Coba cari <i>event</i> Design to code, <i>event</i> tersebut berhubungan dengan pengembangan aplikasi mobile	Berhasil melakukan sesuai <i>test scenario</i>

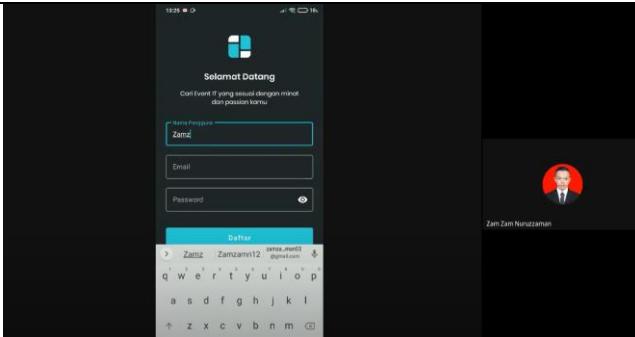
Coba berikan favorit kepada <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Tambahkan pengingat <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba daftar <i>event design to code</i>	Berhasil melakukan sesuai <i>test scenario</i>
Coba kamu lihat jadwal <i>event</i> yang telah kamu tambahkan pengingat Coba buka <i>design to code</i> dari halaman favorit	Berhasil melakukan sesuai <i>test scenario</i>
Coba logout dari aplikasi	Berhasil melakukan sesuai <i>test scenario</i>
Kritik dan saran pengguna	
Sudah cukup dan tidak ada saran	

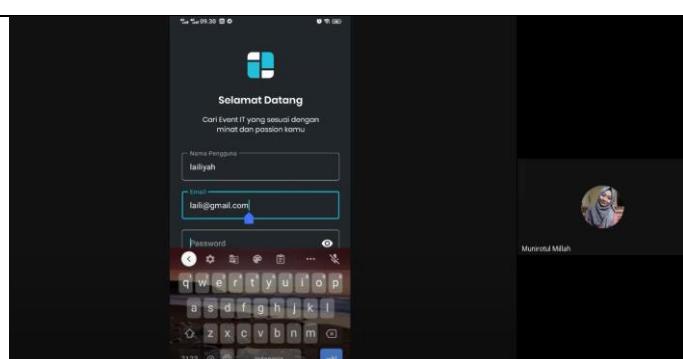
LAMPIRAN D PENGUJIAN USABILITY

D.1 Pengujian *Usability* Pencari Event

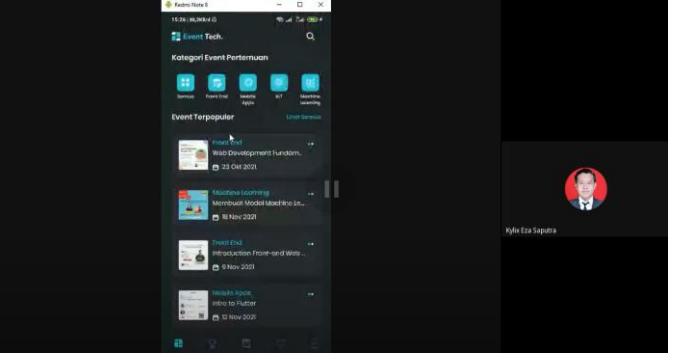
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Suhaili Faruq
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	03 November 2021 pukul 10:33 WIB
Tautan rekaman	https://drive.google.com/file/d/1kF2cKOIq5N6oQqP6upwiYCmMj9vmSJcr/view?usp=sharing
Bukti tangkapan layar	 A screenshot of a mobile application interface titled "Event Tech". The top bar shows the time as 10:33 and battery level at 31%. The main screen displays a search bar and a grid of category icons: Semua, Front End, Mobile Apps, IoT, and Machine Learning. Below this is a section titled "Event Terpopuler" (Popular Events) showing three event cards: "Front End Web Development Fundamentals" (23 Oct 2021), "Machine Learning Membuat Model Machine Learning" (18 Nov 2021), and "Front End Introduction Front-end Development" (25 Nov 2021). At the bottom of the screen, there are navigation icons for back, forward, search, and profile.

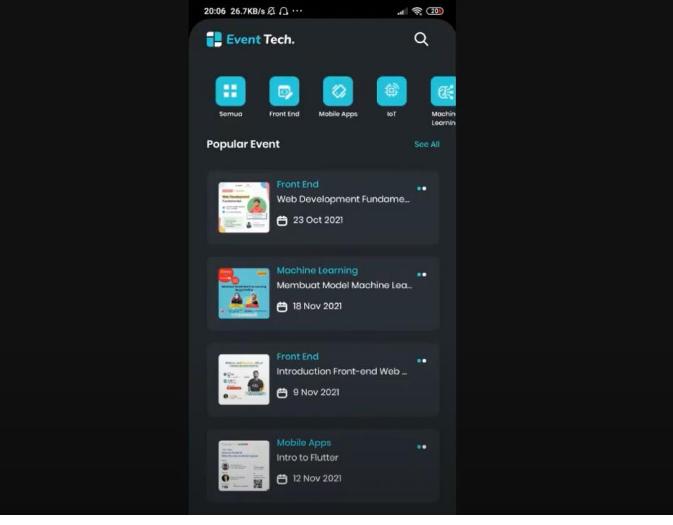
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Zam Zam Nuruzzaman
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	03 November 2021 pukul 13:24 WIB
Tautan rekaman	https://drive.google.com/file/d/1vnvaplg9uWMww-8M4Tt8_6OwwByBjzBj/view?usp=sharing

Bukti tangkapan layar	 <p>The screenshot shows a mobile application's login screen. At the top, it says "Selamat Datang" and "Caril Event ff yang sesuai dengan minat dan passion kamu". Below that are fields for "Nama Pengguna" (containing "Zamz"), "Email" (containing "zamzamz123@gmail.com"), and "Password". A "Daftar" button is at the bottom of the form. A keyboard is visible at the bottom of the screen. On the right side, there is a profile picture placeholder and the text "Zam Zam Nasutamin".</p>
-----------------------	---

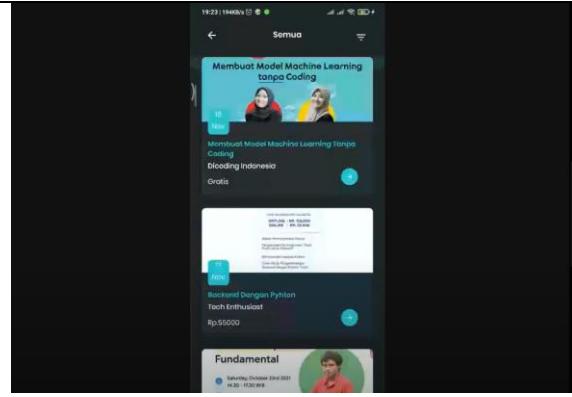
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Munirottul Millah
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	01 November 2021 pukul 19:29 WIB
Tautan rekaman	https://drive.google.com/file/d/1hb560Ge0WLdmG6CCSO_MzkiRdf5Cfcn/view?usp=sharing
Bukti tangkapan layar	 <p>This screenshot shows the same mobile application login screen as the previous one, but now with a successful login. The "Email" field now contains "laili@gmail.com". On the right side, there is a profile picture placeholder with a woman's face and the text "Munirottul Millah".</p>

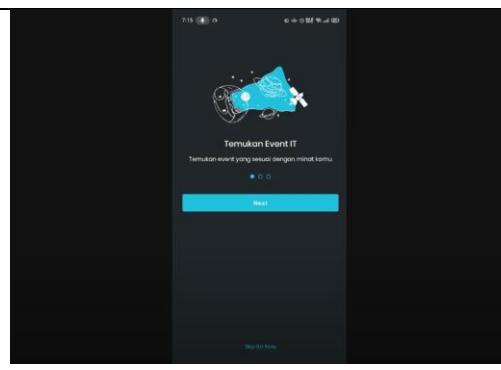
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Kylix Eza Syahputra
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	03 November 2021 pukul 10:33 WIB
Tautan rekaman	https://drive.google.com/file/d/10Lajp3rucoE7V211ddChP_RwdF7F3Vyl/view?usp=sharing

Bukti tangkapan layar	
-----------------------	--

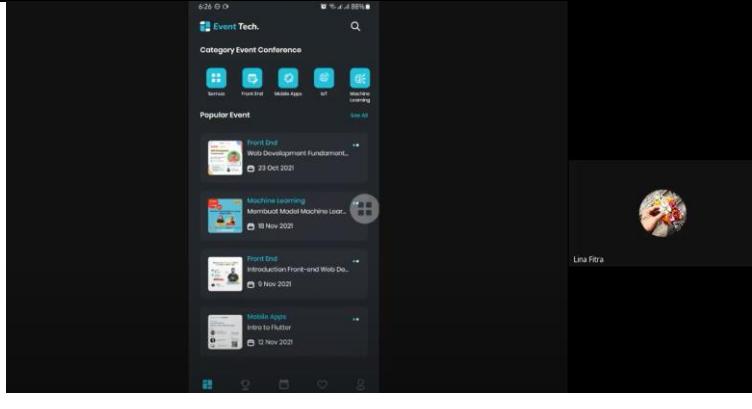
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Kartika Sari
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	02 November 2021 pukul 20:06 WIB
Tautan rekaman	https://drive.google.com/file/d/18Ie2JRIbRYHMa8dw6G4lb3ufKFzIU5Wj/view?usp=sharing
Bukti tangkapan layar	

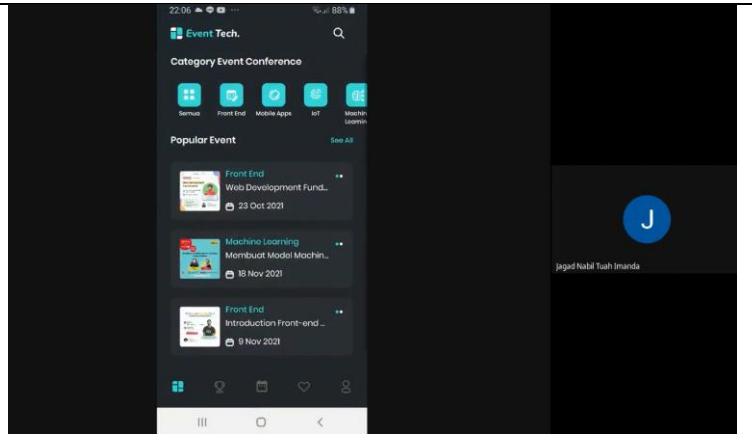
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Alfirs Faizulhaqq
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	03 November 2021 pukul 10:33 WIB

Tautan rekaman	
https://drive.google.com/file/d/1aWSBEEmk-peA1zcf1eBd1k7i5OC0Lcsj/view?usp=sharing	
Bukti tangkapan layar	

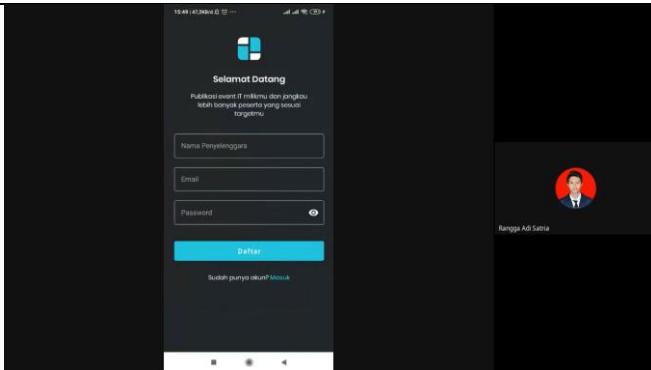
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Dina Noviana
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	02 November 2021 pukul 7:15 WIB
Tautan rekaman	https://drive.google.com/file/d/1OxjJHQeNUNSyHMEz2UD4lukPIf8E7Akq/view?usp=sharing
Bukti tangkapan layar	

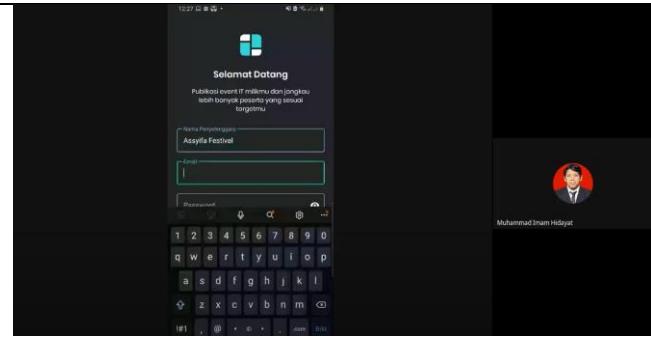
Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Lina Fitra
Karakteristik pengguna	Pencari Event

Waktu pelaksanaan	03 November 2021 pukul 06:26 WIB
Tautan rekaman	
https://drive.google.com/file/d/1Y27YCqhDaE0HwCndM2ePhCkpjz5G1z-G/view?usp=sharing	
Bukti tangkapan layar	

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Jagad Nabil Tuah Imanda
Karakteristik pengguna	Pencari Event
Waktu pelaksanaan	01 November 2021 pukul 22:06 WIB
Tautan rekaman	
https://drive.google.com/file/d/13zqwNHMbG0q35LH6g7X09mJay4oK-Cxr/view?usp=sharing	
Bukti tangkapan layar	

D.1 Pengujian *Usability* Penyelenggara Event

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Rangga Adi Satria
Karakteristik pengguna	Penyelenggara Event
Waktu pelaksanaan	03 November 2021 pukul 15:40 WIB
Tautan rekaman	https://drive.google.com/file/d/1ka9WOCOvtsEQbD7mJk-TNSBDnnH9Kreq/view?usp=sharing
Bukti Tangkapan Layar	

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Muhammad Imam Hidayat
Karakteristik pengguna	Penyelenggara Event
Waktu pelaksanaan	03 November 2021 pukul 12:29 WIB
Tautan rekaman	https://drive.google.com/file/d/1yyOjq26YWJhfJXVleHBc9dmFNRTLY3zF/view?usp=sharing
Bukti Tangkapan Layar	

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Shafatyra Reditha Shalsadila
Karakteristik pengguna	Penyelenggara Event
Waktu pelaksanaan	03 November 2021 pukul 14:09 WIB
Tautan rekaman	https://drive.google.com/file/d/1emC8LFittGaneteOUGOiYm9Z4Q2nW-zQ/view?usp=sharing
Bukti Tangkapan Layar	

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Halimah Septya Mikayla
Karakteristik pengguna	Penyelenggara Event
Waktu pelaksanaan	03 November 2021 pukul 13:45 WIB
Tautan rekaman	https://drive.google.com/file/d/1ASO5hE6gdpt_8XajlhBhE05snFIW7zST/view?usp=sharing
Bukti Tangkapan Layar	

Pengujian Usability	
Nama pengamat	Ilham Dwi Muchlison
Nama responden	Dhika Aldhika
Karakteristik pengguna	Penyelenggara Event
Waktu pelaksanaan	03 November 2021 pukul 13:12 WIB
Tautan rekaman	https://drive.google.com/file/d/1iPGh63VfBvDHI4MN92Ly9SDho4BJtd0/view ?usp=sharing
Bukti Tangkapan Layar	