



TUGAS AKHIR - IF184802

# RANCANG BANGUN APLIKASI REGISTRASI TAMU PADA PERANGKAT BERGERAK BERBASIS ANDROID

ADIB WAHYU KUNCORO  
NRP 05111540000036

Dosen Pembimbing  
Nurul Fajrin Ariyani, S.Kom., M.Sc.  
Abdul Munif, S.Kom., M.Sc.

Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020





**TUGAS AKHIR - IF184802**

# **RANCANG BANGUN APLIKASI REGISTRASI TAMU PADA PERANGKAT BERGERAK BERBASIS ANDROID**

**ADIB WAHYU KUNCORO**  
NRP 0511154000036

Dosen Pembimbing  
Nurul Fajrin Ariyani, S.Kom, M.Sc.  
Abdul Munif, S.Kom., M.Sc.

Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESIS - IF184802**

**DESIGN OF GUEST REGISTRATION  
APPLICATION FOR ANDROID BASED  
MOBILE DEVICES**

**ADIB WAHYU KUNCORO**  
NRP 0511154000036

Advisor  
Nurul Fajrin Ariyani, S.Kom, M.Sc.  
Abdul Munif, S.Kom., M.Sc.

Department of Informatics Engineering  
Faculty of Intelligent Electrical and Informatics  
Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*[Halaman ini sengaja dikosongkan]*

**LEMBAR PENGESAHAN**  
**RANCANG BANGUN APLIKASI REGISTRASI TAMU**  
**PADA PERANGKAT BERGERAK BERBASIS ANDROID**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer pada  
Rumpun Mata Kuliah Rekayasa Perangkat Lunak  
Program Studi S-1 Teknik Informatika  
Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**ADIB WAHYU KUNCORO**

NRP : 0511154000036

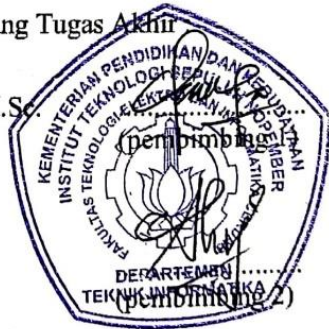
Disetujui oleh Dosen Pembimbing Tugas Akhir

Nurul Fajrin Ariyani, S.Kom, M.Sc.

NIP: 19860722 201504 2 003

Abdul Munif., S.Kom., M.Sc.

NIP: 19860823 201504 1 004



**SURABAYA**  
**JANUARI 2020**

*[Halaman ini sengaja dikosongkan]*



## **RANCANG BANGUN APLIKASI REGISTRASI TAMU PADA PERANGKAT BERGERAK BERBASIS ANDROID**

Nama Mahasiswa : ADIB WAHYU KUNCORO  
NRP : 0511154000036  
Jurusan : Informatika ITS  
Dosen Pembimbing I : Nurul Fajrin Ariyani,S.Kom,M.Sc.  
Dosen Pembimbing II : Abdul Munif., S.Kom., M.Sc.

### **Abstrak**

*Pada sebuah acara, undangan merupakan hal penting sebagai pemberitahuan akan adanya suatu acara dengan harapan agar tamu undangan dapat hadir pada waktu dan tempat yang telah ditetapkan. Undangan biasanya akan diberikan dalam bentuk kertas undangan atau label yang nantinya akan digunakan sebagai tiket masuk sebuah acara. Sebelum memasuki sebuah acara biasanya akan ada proses registrasi, dimana para tamu akan dicek undangannya satu per satu. Proses registrasi biasanya dilakukan dengan cara menuliskan data tamu undangan satu per satu pada buku tamu. Hal ini tentunya akan menjadi masalah jika jumlah tamu undangan mencapai ratusan atau ribuan. Dimana selain memerlukan waktu yang lama, keakuratan pendataan juga tidak terjamin.*

*Untuk membantu menyelesaikan masalah tersebut, dalam Tugas Akhir ini dibangun sebuah aplikasi untuk mempermudah pemilik acara dalam mengelola data tamu dan mempermudah tamu melakukan registrasi ke dalam suatu acara. Pemilik acara dapat membuat acara dan menampilkan acara tersebut ke dalam aplikasi sehingga tamu dapat mendaftar sendiri ke acara tersebut. Setelah tamu terdaftar, maka tamu dapat mengunduh undangan yang nantinya dapat digunakan untuk memasuki sebuah acara. Selain itu, pemilik acara juga dapat memilih sendiri tamu yang dapat menghadiri acara dengan cara menambahkan data tamu yang dipilih dan mengirimkan undangan melalui email data tamu. Undangan berbentuk file PDF dimana didalamnya terdapat kode unik berbentuk QR Code yang digunakan untuk membedakan*

*tamu yang satu dengan yang lainnya. Saat acara dimulai, petugas scanner dapat mencatat kehadiran tamu dengan cara memindai QR Code pada undangan. Dengan hal ini, tentunya dapat memudahkan dan mengurangi jumlah antrian saat proses mencatat kehadiran tamu serta mengurangi biaya dan penggunaan kertas dalam penyelenggaraan sebuah acara.*

***Kata kunci: Android, QR Code, Registrasi.***

## **DESIGN OF GUEST REGISTRATION APPLICATION FOR ANDROID BASED MOBILE DEVICES**

Name : ADIB WAHYU KUNCORO  
NRP : 0511154000036  
Major : Informatika ITS  
Supervisor I : Nurul Fajrin Ariyani, S.Kom, M.Sc.  
Supervisor II : Abdul Munif, S.Kom., M.Sc.

### **Abstract**

*At an event, an invitation is important as a notification of an event in the hope that the invited guests can be present at the designated time and place. Invitations will usually be given in the form of invitation paper or labels that will later be used as an entry ticket for an event. Before entering an event there will usually be a registration process, where guests will be checked for invitations one by one. The registration process is usually done by writing the invitation data one by one in the guest book. This will certainly be a problem if the number of invited guests reaches hundreds or thousands. Where in addition to requiring a long time, the accuracy of data collection is also not guaranteed.*

*To help solve this problem, in this Final Project an application is built to facilitate the event owner in managing guest data and making it easier for guests to register in an event. The event owner can create an event and display the event in the application so guests can register themselves for the event. After a guest is registered, then the guest can download an invitation which can later be used to enter an event. In addition, the event owner can also choose the guests who can attend the event by adding selected guest data and sending invitations via guest data e-mail. The invitation is in the form of a PDF file in which there is a unique code in the form of a QR Code that is used to distinguish guests from one another. When the event starts, the scanner can record the attendance of guests by scanning the QR Code on the invitation. With this, of course it can facilitate and reduce the*

*number of queues when the process of recording guest attendance and reduce the cost and use of paper in organizing an event.*

***Keywords: Android, QR Code, Registration.***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Rancang Bangun Aplikasi Registrasi Tamu Pada Perangkat Bergerak Berbasis Android**”.

Dalam pelaksanaan tugas akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Bapak dan Ibu penulis yang telah memberikan dukungan doa, cinta dan apapun tanpa pernah menghitung kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Mas Kresna dan Mbak Shinta selaku kakak penulis yang selalu memberikan dukungan moral.
4. Ibu Nurul Fajrin Ariyani, S.Kom, M.Sc. selaku dosen wali dan dosen pembimbing tugas akhir pertama yang telah membimbing, memotivasi dan memberikan banyak masukan dalam pengerjaan tugas akhir ini.
5. Bapak Abdul Munif, S.Kom., M.Sc. selaku dosen pembimbing tugas akhir kedua yang selalu memberikan koreksi serta masukan-masukan yang dapat penulis kembangkan dari tugas akhir ini.
6. Bapak dan Ibu dosen Departemen Informatika ITS yang telah mengajarkan banyak ilmu berharga kepada penulis.
7. Mas Rio, Mas Iqbal yang selalu menyemangati dan percaya pada penulis dalam menyelesaikan Tugas Akhir.
8. Teman-teman UKM Catur ITS yang menjadi keluarga selama penulis menimba ilmu di Departemen Informatika ITS.

9. Teman-teman Kos Gang Makam H-18.
10. Teman-teman satu angkatan Informatika ITS 2015 yang saling menyemangati satu sama lain.
11. Pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak. Tidak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan tugas akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Januari 2020

Adib Wahyu Kuncoro

# DAFTAR ISI

<b>Abstrak .....</b>	<b>vii</b>
<b>Abstract .....</b>	<b>ix</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>DAFTAR ISI .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xxi</b>
<b>1. BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Metodologi .....	3
1.6 Sistematika Penulisan .....	5
<b>2 BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Registrasi Tamu .....	7
2.2 QR Code.....	8
2.3 QR Code Generator.....	10
2.4 QR Code Scanner.....	10
2.5 Android .....	11
2.6 Laravel .....	12
2.7 Composer .....	13
2.8 PHP .....	14
2.9 Java .....	14

2.10	MVC (Model-View-Controller) .....	15
2.11	API.....	15
2.12	JSON.....	16
2.13	Perbandingan Aplikasi Sejenis .....	17
<b>3</b>	<b>BAB III ANALISIS DAN PERANCANGAN....</b>	<b>23</b>
3.1	Analisis Sistem .....	23
3.2	Analisis Proses Bisnis.....	24
3.2.1	Proses Bisnis Kotakevent Secara Umum....	24
3.2.2	Tipe Event.....	26
3.3	Deskripsi Umum Sistem.....	32
3.3.1	Spesifikasi Kebutuhan Perangkat Lunak ....	33
3.4	Kasus Penggunaan Kotakevent .....	37
3.4.1	UC-01: Membuat Event.....	39
3.4.2	UC-02: Mengelola Event.....	40
3.4.3	UC-03: Mengelola Petugas Scanner .....	43
3.4.4	UC-04: Mengelola Kategori Tamu .....	46
3.4.5	UC-05: Mengelola Tamu Undangan .....	49
3.4.6	UC-06: Mengelola Pengajuan Pendaftar ....	52
3.4.7	UC-07:Mencari Event.....	55
3.4.8	UC-08:Melakukan Registrasi <i>Event</i> .....	58
3.4.9	UC-09:Mengunduh Tiket.....	60
3.4.10	UC-10: Mencatat Kehadiran Tamu .....	61
3.4.11	UC-11:Mengelola Pengajuan Event.....	63
3.4.12	UC-12:Mengirim Tiket.....	65
3.4.13	UC-13:Melihat Daftar Tiket.....	68



3.4.14	UC-14:Melihat Daftar Hadir.....	69
3.5	Perancangan .....	70
3.5.1	Perancangan Diagram Kelas .....	70
3.5.2	Perancangan Basis Data .....	71
3.5.3	Perancangan Antarmuka Web.....	75
3.5.4	Perancangan Antarmuka Mobile.....	84
3.5.5	Perancangan API Endpoint Petugas Registrasi	86
<b>4</b>	<b>BAB IV IMPLEMENTASI .....</b>	<b>87</b>
4.1	Lingkungan Implementasi.....	87
4.2	Implementasi Antarmuka Sistem .....	87
4.2.1	Antarmuka Web .....	88
4.2.2	Halaman Utama Kotakevent .....	88
4.2.3	Halaman Registrasi <i>Event</i> .....	89
4.2.4	Halaman Membuat Event.....	90
4.2.5	Halaman Mengelola Event.....	91
4.2.6	Halaman Mengelola Tiket.....	91
4.2.7	Halaman Mencari Event.....	92
4.2.8	Halaman Mengelola Pengajuan Event .....	92
4.2.9	Antarmuka Mobile .....	95
4.2.10	Halaman Mencatat Kehadiran .....	95
4.2.11	Halaman Melihat Daftar Hadir .....	96
4.2.12	Implementasi API Endpoint Petugas Scanner	97
<b>5</b>	<b>BAB V UJI COBA DAN EVALUASI .....</b>	<b>99</b>
5.1	Lingkungan Uji Coba.....	99

5.2	Skenario Pengujian .....	100
5.2.1	Pengujian Terhadap Fungsionalitas Sistem 100	
5.2.2	Kasus Pengujian Penanganan Error.....	131
5.2.3	Kasus Pengujian Response Body.....	133
5.2.4	Pengujian Terhadap Pengguna .....	135
5.3	Rekapitulasi Kasus Pengujian .....	146
5.3.1	Hasil Pengujian Fungsional .....	146
5.3.2	Hasil Pengujian Penanganan Error .....	147
5.3.3	Hasil Pengujian Response Body .....	147
<b>6</b>	<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>149</b>
6.1	Kesimpulan.....	149
6.2	Saran.....	149
	<b>DAFTAR PUSTAKA .....</b>	<b>151</b>
	<b>BIODATA PENULIS .....</b>	<b>153</b>

## DAFTAR GAMBAR

Gambar 3.1 Proses Bisnis Kotakevent Secara Umum .....	25
Gambar 3.2 Proses Bisnis Event Tipe Open Berbayar Umum ...	27
Gambar 3.3 Proses Bisnis Event Tipe Open Berbayar Kode.....	28
Gambar 3.4 Proses Bisnis Event Tipe Open Tidak Berbayar Umum .....	29
Gambar 3.5 Proses Bisnis Event Tipe Open Tidak Berbayar Kode .....	29
Gambar 3.6 Proses Bisnis Event Tipe Open Tidak Berbayar Kode .....	30
Gambar 3.7 Proses Bisnis Event Tipe Close Event .....	31
Gambar 3.8 Ilustrasi Arsitektur Sistem.....	32
Gambar 3.9 Diagram Kasus Penggunaan Kotakevent .....	38
Gambar 3.10 Diagram Aktivitas Membuat Event.....	40
Gambar 3.11 Diagram Aktivitas Mengelola <i>Event</i> .....	42
Gambar 3.12 Diagram Aktivitas Mengelola Petugas Scanner....	45
Gambar 3.13 Diagram Aktivitas Mengelola Kategori Tamu.....	48
Gambar 3.14 Diagram Aktivitas Mengelola Data Tamu Undangan .....	51
Gambar 3.15 Diagram Aktivitas Mengelola Pengajuan Pendaftar .....	54
Gambar 3.16 Diagram Aktivitas Mencari Event.....	57
Gambar 3.17 Diagram Aktivitas Melakukan Registrasi <i>Event</i> ...	59
Gambar 3.18 Diagram Aktivitas Mengunduh Tiket .....	61
Gambar 3.19 Diagram Aktivitas Mencatat Kehadiran Tamu .....	63
Gambar 3.20 Diagram Aktivitas Mengelola Pengajuan <i>Event</i> ...	65
Gambar 3.21 Diagram Aktivitas Mengirim Tiket.....	67
Gambar 3.22 Diagram Aktivitas Melihat Daftar Tiket .....	68
Gambar 3.23 Diagram Aktivitas Melihat Daftar Hadir .....	69
Gambar 3.24 Diagram Kelas Controller Kotakevent.....	70
Gambar 3.25 CDM Kotakevent .....	72
Gambar 3.26 PDM Kotakevent.....	73
Gambar 3.27 Rancangan Antarmuka Utama Kotakevent .....	75

Gambar 3.28 Rancangan Antarmuka Detail Event.....	77
Gambar 3.29 Rancangan Antarmuka Membuat Event .....	78
Gambar 3.30 Rancangan Antarmuka Mengelola Event .....	80
Gambar 3.31 Rancangan Antarmuka Mengelola Tiket .....	81
Gambar 3.32 Rancangan Antarmuka Mencari Event .....	82
Gambar 3.33 Rancangan Antarmuka Mengelola Pengajuan Event .....	83
Gambar 3.34 Rancangan Antarmuka Registrasi Event .....	84
Gambar 3.35 Rancangan Antarmuka Melihat Daftar Hadir .....	85
Gambar 4.1 Halaman Utama Kotakevent.....	88
Gambar 4.2 Halaman Mendaftar Event.....	89
Gambar 4.3 Halaman Membuat Event .....	90
Gambar 4.4 Halaman Mengelola Event .....	91
Gambar 4.5 Halaman Mengelola Tiket .....	92
Gambar 4.6 Halaman Mengelola Pengajuan Event.....	93
Gambar 4.7 Halaman Mencari Event .....	94
Gambar 4.8 Halaman Registrasi Event.....	95
Gambar 4.9 Halaman Melihat Daftar Hadir .....	96
Gambar 5.1 Membuat Event Tipe Open Berbayar Umum .....	103
Gambar 5.2 Mencari <i>Event</i> Menggunakan Nama.....	104
Gambar 5.3 Mendaftar <i>Event</i> Berbayar.....	104
Gambar 5.4 Membuat <i>Event</i> Tipe Open Berbayar Kode.....	108
Gambar 5.5 Mencari <i>Event</i> Menggunakan Kode .....	108
Gambar 5.6 Mendaftar <i>Event</i> Tipe Open Tidak Berbayar Umum .....	111
Gambar 5.7 Mendaftar <i>Event</i> Tidak Berbayar.....	112
Gambar 5.8 Membuat <i>Event</i> Tipe Open Tidak Berbayar Kode	115
Gambar 5.9 Membuat <i>Event</i> Tipe Close <i>Event</i> .....	119
Gambar 5.10 Mengirim Tiket Ke Email Tamu Undangan .....	120
Gambar 5.11 <i>Mengimport</i> Data Tamu Undangan .....	120
Gambar 5.12 Mengelola Pengajuan Pendaftar .....	122
Gambar 5.13 Menambah Kategori Tamu .....	124
Gambar 5.14 Mengubah Kategori Tamu .....	124
Gambar 5.15 Menambah Foto Galeri <i>Event</i> .....	126
Gambar 5.16 Tabel Mengelola Petugas Registrasi.....	129

Gambar 5.17 Mencatat Kehadiran Tamu .....	130
Gambar 5.18 Mendaftar Ke <i>Event</i> Yang Telah Selesai .....	132
Gambar 5.19 Melakukan Login Di Luar Waktu <i>Event</i> .....	133

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Perbandingan Aplikasi Sejenis .....	18
Tabel 3.1 Deskripsi Aktor pada Aplikasi Kotakevent .....	36
Tabel 3.2 Spesifikasi Kasus Penggunaan Membuat Event .....	39
Tabel 3.3 Spesifikasi Kasus Penggunaan Mengelola Event .....	41
Tabel 3.4 Spesifikasi Kasus Penggunaan Mengelola Petugas Scanner.....	43
Tabel 3.5 Spesiifikasi Kasus Penggunaan Mengelola Kategori Tamu .....	46
Tabel 3.6 Spesifikasi Kasus Penggunaan Mengelola Tamu Undangan.....	49
Tabel 3.7 Spesifikasi Kasus Penggunaan Mengelola Pengajuan Pendaftar .....	52
Tabel 3.8 Spesifikasi Kasus Penggunaan Mencari Event .....	55
Tabel 3.9 Spesifikasi Kasus Penggunaan Melakukan Registrasi <i>Event</i> .....	58
Tabel 3.10 Spesifikasi Kasus Penggunaan Mengunduh Tiket ....	60
Tabel 3.11 Spesifikasi Kasus Penggunaan Mencatat Kehadiran Tamu .....	61
Tabel 3.12 Spesifikasi Kasus Penggunaan Mengelola Pengajuan <i>Event</i> .....	64
Tabel 3.13 Spesifikasi Kasus Penggunaan Mengirim Tiket.....	66
Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Daftar Tiket	68
Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat Daftar Hadir	69
Tabel 3.16 Penjelasan Antarmuka Utama Kotakevent .....	76
Tabel 3.17 Penjelasan Antarmuka Mendaftar Event.....	77
Tabel 3.18 Penjelasan Antarmuka Membuat Event.....	79
Tabel 3.19 Penjelasan Antarmuka Mengelola Event .....	80
Tabel 3.20 Penjelasan Antarmuka Mengelola Tiket .....	82
Tabel 3.21 Penjelasan Antarmuka Mencari Event.....	83
Tabel 3.22 Penjelasan Antarmuka Registrasi Event .....	85
Tabel 3.23 Perancangan API Endpoint Petugas Registrasi.....	86
Tabel 4.1 Penjelasan Implementasi API Endpoint Petugas Registrasi .....	98

Tabel 5.1 Lingkungan Pengujian Aplikasi Web.....	99
Tabel 5.2 Lingkungan Pengujian Aplikasi Mobile.....	99
Tabel 5.3 Kasus Membuat Event Open Berbayar Umum .....	100
Tabel 5.4 Kasus Mencari Event Open Berbayar Umum .....	101
Tabel 5.5 Kasus Mendaftar Event Open Berbayar Umum .....	102
Tabel 5.6 Kasus Membuat Event Open Berbayar Kode.....	105
Tabel 5.7 Kasus Mencari Event Open Berbayar Kode.....	106
Tabel 5.8 Kasus Mendaftar Event Open Berbayar Kode .....	106
Tabel 5.9 Kasus Membuat Event Open Tidak Berbayar Umum .....	109
Tabel 5.10 Kasus Mencari Event Open Tidak Berbayar Umum .....	110
Tabel 5.11 Kasus Mendaftar Event Open Tidak Berbayar Umum .....	110
Tabel 5.12 Kasus Membuat Event Open Tidak Berbayar Kode .....	112
Tabel 5.13 Kasus Mencari Event Open Tidak Berbayar Kode. 113	
Tabel 5.14 Kasus Mendaftar Event Open Tidak Berbayar Kode .....	114
Tabel 5.15 Kasus Membuat Event Tipe Close Event.....	115
Tabel 5.16 Kasus Menambah Data Tamu .....	116
Tabel 5.17 Kasus Mengimport Data Tamu .....	117
Tabel 5.18 Kasus Mengirim Tiket Tamu.....	118
Tabel 5.19 Kasus Menyetujui Pengajuan Pendaftar .....	121
Tabel 5.20 Kasus Menolak Pengajuan Pendaftar .....	121
Tabel 5.21 Kasus Menambah Kategori Kategori Tamu .....	122
Tabel 5.22 Kasus Mengubah Kategori Tamu .....	123
Tabel 5.23 Kasus Mengubah Data <i>Event</i> .....	125
Tabel 5.24 Kasus Menambah Foto Galeri <i>Event</i> .....	125
Tabel 5.25 Kasus Menambah Petugas Scanner .....	127
Tabel 5.26 Kasus Mengubah Petugas Scanner .....	127
Tabel 5.27 Kasus Menghapus Petugas Scanner .....	128
Tabel 5.28 Kasus Mencatat Kehadiran Tamu.....	129
Tabel 5.29 Mendaftar Ke <i>Event</i> Yang Telah Selesai.....	131
Tabel 5.30 Melakukan Login Di Luar Waktu <i>Event</i> .....	132



Tabel 5.31 Skenario Pengujian Response Body .....	134
Tabel 5.32 Pengujian API Endpoint .....	135
Tabel 5.33 Data Pengisi Kuisisioner .....	136
Tabel 5.34 Hasil Rekapitulasi Kuisisioner Pemilik <i>Event</i> dan Petugas Scanner .....	137
Tabel 5.35 Hasil Rekapitulasi Kuisisioner Tamu .....	138
Tabel 5.36 Persentase Alternatif Jawaban Kuisisioner Pemilik <i>Event</i> Dan Petugas Scanner.....	139
Tabel 5.37 Persentase Alternatif Jawaban Kuisisioner Tamu .....	140
Tabel 5.38 Persentasi Interpretasi Penilaian Kuisisioner Pemilik Event dan Petugas Scanner .....	142
Tabel 5.39 Persentasi Interpretasi Penilaian Kuisisioner Tamu ..	142
Tabel 5.40 Interpretasi Skor Hasil Penilaian Kuisisioner Pemilik <i>Event</i> Dan Petugas Scanner .....	143
Tabel 5.41 Interpretasi Skor Hasil Penilaian Kuisisioner Tamu .	144
Tabel 5.42 Hasil Kuisisioner Penilaian Sistem.....	145
Tabel 5.43 Rekapitulasi Hasil Pengujian Aplikasi Kotakevent	146
Tabel 5.44 Rekapitulasi Hasil Pengujian Penanganan Error .....	147
Tabel 5.45 Rekapitulasi Hasil Pengujian API Endpoint .....	148

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

Bab pendahuluan membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

### **1.1 Latar Belakang**

Acara adalah hal atau pokok yang akan dilakukan secara terencana untuk suatu tujuan. Acara dilaksanakan berdasarkan rencana yang telah disusun sebelumnya, tidak dilakukan secara spontan. Dalam acara resmi, pada umumnya diperlukan suatu undangan sebagai pemberitahuan akan adanya suatu acara dengan harapan agar tamu undangan dapat hadir pada waktu dan tempat yang telah ditetapkan. Undangan juga berfungsi untuk membatasi jumlah tamu yang akan datang berdasarkan keinginan pemilik acara. Undangan biasanya akan diberikan dalam bentuk kertas undangan atau label dan dikirim satu per satu kepada alamat tamu undangan yang nantinya akan digunakan sebagai tiket masuk sebuah acara. Sebelum acara dimulai, para tamu undangan akan melakukan registrasi atau mengisi buku tamu untuk pendataan. Proses registrasi dilakukan dengan cara menuliskan data tamu undangan satu per satu pada buku tamu. Hal ini tentunya akan menjadi masalah jika jumlah tamu undangan mencapai ratusan atau ribuan. Dimana selain memerlukan waktu yang lama, keakuratan pendataan juga tidak terjamin.

Seiring perkembangan teknologi informasi dalam era globalisasi saat ini, muncul suatu inovasi untuk membuat undangan menggunakan QR Code. Tamu undangan akan menerima undangan yang disertai QR Code dari pemilik acara. Teknologi QR Code digunakan untuk melakukan akses cepat ke suatu data (data teks, tautan dan sebagainya), dalam hal ini pemanfaatan QR Code akan disertakan pada setiap undangan yang nantinya akan discan menggunakan alat scanner, sehingga panitia

registrasi dapat dengan cepat melakukan pendataan terhadap undangan tamu tersebut. Dengan menggunakan sistem tersebut tentunya akan dapat memangkas waktu, tenaga dan biaya yang dikeluarkan serta memudahkan pemilik acara untuk mengelola tamu-tamunya.

Teknologi yang sedang berkembang pesat saat ini adalah smartphone berbasis android dimana pada tahun 2017, 73% perangkat mobile global menggunakan Android. Melihat data tersebut, tentunya akan sangat bermanfaat jika dapat memaksimalkan fitur-fitur yang terdapat pada smartphone android. Berdasarkan latar belakang tersebut, penulis ingin memanfaatkan kamera pada smartphone android untuk menggantikan fungsi dari alat scanner untuk menscan QR Code undangan dan menggantikan kertas undangan dengan undangan dalam bentuk file pdf. Dengan hal ini, tentunya dapat memangkas biaya yang dikeluarkan untuk membeli alat scanner dan mencetak undangan. Untuk saat ini sudah terdapat sistem tersebut, namun untuk menggunakannya harus membayar sesuai dengan jumlah undangan yang ada dan masih menggunakan banyak kertas dalam pembuatan undangan.

Oleh karena itu, dalam tugas akhir ini penulis akan mencoba membawa solusi baru dengan dibangunnya “Rancang Bangun Aplikasi Registrasi Tamu Pada Perangkat Bergerak Berbasis Android” yang diberi nama dengan Kotakevent untuk memudahkan proses registrasi tamu menjadi sederhana, praktis dan cepat.

## **1.2 Rumusan Masalah**

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut.

1. Bagaimana men-generate QR Code pada undangan untuk setiap data tamu undangan?

2. Bagaimana melakukan verifikasi tamu undangan yang sudah melakukan registrasi?
3. Bagaimana cara mengelola data tamu undangan berdasarkan kategorinya?

### **1.3 Batasan Masalah**

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan antara lain:

1. Satu tiket hanya diuntukkan untuk satu orang tamu saja
2. Sistem ini hanya dapat digunakan oleh smartphone berbasis android
3. Aplikasi mobile ini tidak bisa digunakan untuk mengelola data event

### **1.4 Tujuan**

Tugas Akhir ini mempunyai beberapa tujuan, antara lain sebagai berikut.

1. Memudahkan proses registrasi tamu dan mengurangi jumlah antrian pada saat melakukan registrasi.
2. Mengurangi biaya dan penggunaan kertas pada saat proses registrasi tamu.

### **1.5 Metodologi**

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

#### **1. Penyusunan Proposal Tugas Akhir**

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar

belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

## **2. Studi literatur**

Pada tahap ini, akan dicari studi literatur yang relevan untuk dijadikan referensi dalam pengerjaan tugas akhir. Studi literatur dapat diambil dari buku, internet, ataupun materi dalam suatu mata kuliah yang berhubungan dengan metode yang akan digunakan.

## **3. Implementasi**

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

## **4. Uji Coba dan Evaluasi**

Pada tahapan ini dilakukan uji coba pada perangkat lunak dengan pengujian berdasarkan *usability* dan *maintenance*.

## **5. Penyusunan Buku Tugas Akhir**

Pada tahapan ini disusun buku yang membuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## **1.6 Sistematika Penulisan**

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

### **1. Bab I. Pendahuluan**

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

### **2. Bab II. Tinjauan Pustaka**

Bab tinjauan pustaka berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir.

### **3. Bab III. Analisis dan Perancangan**

Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.

### **4. Bab IV. Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa tampilan antarmuka yang telah dibuat dan dapat berfungsi untuk mengakomodir kebutuhan fungsional yang ada.

### **5. Bab V. Uji Coba dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

### **6. Bab VI. Kesimpulan dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

*[Halaman ini sengaja dikosongkan]*



## **BAB II**

### **TINJAUAN PUSTAKA**

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

#### **2.1 Registrasi Tamu**

Registrasi berasal dari bahasa Inggris *registration* yang memiliki arti daftar. Menurut Kamus Besar Bahasa Indonesia<sup>1</sup>, Pendaftaran adalah proses, cara, perbuatan mendaftarkan (mendaftarkan); pencatatan nama, alamat, dan sebagainya ke dalam daftar. Jadi, registrasi merupakan proses pencatatan identitas pendaftar yang akan disimpan kedalam suatu tempat penyimpanan yang nantinya akan digunakan dalam proses registrasi.

Registrasi tamu adalah suatu kegiatan yang diselenggarakan oleh petugas registrasi yang berhubungan dengan acara. Kegiatan tersebut meliputi: pencatatan identitas tamu, pencatatan waktu dan tanggal serta pengarsipan yang berhubungan dengan acara. Proses registrasi ini biasanya dilakukan sebelum tamu memasuki sebuah acara. Dalam registrasi tamu sendiri terdapat dua cara yaitu konvensional dan modern.

Dalam cara konvensional, registrasi tamu dilakukan dengan cara manual yaitu dengan menuliskan satu per satu data tamu undangan yang hadir, data bisa berupa : identitas tamu, waktu kehadiran dan data lain yang berhubungan dengan acara. Data-data tersebut akan disimpan dalam buku tamu yang nantinya akan digunakan dalam proses registrasi.

Dalam cara modern, registrasi tamu dapat dilakukan hanya dengan cara menunjukkan undangan yang terdapat QR Code

---

<sup>1</sup> KBBI, <https://jurnal-oldi.or.id/public/kbbi.pdf>

kepada petugas registrasi. QR Code pada undangan akan discan, bisa menggunakan alat scanner, webcam, atau kamera smartphone android lalu secara otomatis sistem akan meverifikasi tamu undangan yang telah melakukan registrasi. Data-data tamu tersebut akan disimpan ke dalam database dan dapat dikelola admin melalui aplikasi berbasis web pada sistem tersebut.

## 2.2 QR Code

QR-code adalah jenis barcode yang berbentuk dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation, 9 sebuah perusahaan di Jepang, yang dipublikasikan pada tahun 1994. QR merupakan singkatan dari Quick Response (respon / tanggapan cepat), sehingga fungsi atau tujuan utama dari teknologi ini adalah penyampaian informasi dengan cepat dan mendapat tanggapan atau respons yang cepat pula. Oleh karena itu QR-code dapat dengan mudah dibaca oleh pemindai. Berbeda dengan barcode biasa yang berbentuk satu dimensi dan menyimpan informasi secara horizontal, QR-code mampu menyimpan informasi secara horisontal dan vertikal. QR-code juga mampu menyimpan teks alfanumerik, kanji, kana, hiragana, simbol, biner, dan control code. QR-code adalah simbol matriks dengan karakteristik sebagai berikut :

1. Karakter yang dapat di-encode:
  - a. Data numerik (digit 0 - 9)
  - b. Data alfanumerik (digit 0 - 9; huruf kapital AZ; sembilan karakter tanda baca: space, \$ % \*+ - . / : )
  - c. Data byte (default: ISO/IEC 8859-1)
  - d. Huruf kanji
2. Versi (Ukuran simbol) (tidak termasuk quiet zone): 21 x 21 modul hingga 177 x 177 modul (Versi 1 hingga 40, setiap versi yang lebih tinggi memiliki ukuran 4 modul lebih besar pada sisinya).

3. Ukuran maksimum simbol QR Code, Versi 40-L:
  - a. Data numerik: 7089 characters
  - b. Data alfanumerik: 4296 characters
  - c. Data byte: 2953 characters
  - d. Data kanji: 1817 characters
  
4. Level koreksi kesalahan yang tersedia: Empat level koreksi kesalahan Reed-Solomon dengan kemampuan koreksi sebagai berikut:
  - a. L 7%
  - b. M 15%
  - c. Q 25%
  - d. H 30%

Kode QR memiliki kapasitas tinggi dalam data pengkodean, yaitu mampu menyimpan semua jenis data, seperti data numerik, data alfabatis, kanji, kana, hiragana, simbol, dan kode biner. Secara spesifik, kode QR mampu menyimpan data jenis numerik sampai dengan 7.089 karakter, data alphanumerik sampai dengan 4.296 karakter, kode binari sampai dengan 2.844 byte, dan huruf kanji sampai dengan 1.817 karakter. Selain itu kode QR memiliki tampilan yang lebih kecil daripada kode batang. Hal ini dikarenakan kode QR mampu menampung data secara horizontal dan vertikal, oleh karena itu secara otomatis ukuran dari tampilannya gambar kode QR bisa hanya sepersepuluh dari ukuran sebuah kode batang. Tidak hanya itu kode QR juga tahan terhadap kerusakan, sebab kode QR mampu memperbaiki kesalahan sampai dengan 30%. Oleh karena itu, walaupun sebagian simbol kode QR kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun sepanjang 360 derajat.[1]

### 2.3 QR Code Generator

QR Code Generator digunakan untuk merubah atau mengkodekan sebuah inputan seperti teks, gambar, email maupun kartu nama. QR Code generator akan meminta masukan sebuah data yang dapat diambil dari sistem operasi komputer. Kemudian file tersebut dibaca sebagai byte stream sebelum diubah menjadi representasi byte, numerik, atau alfanumerik dengan algoritma tertentu untuk kemudian diubah menjadi *qr code* dengan algoritma yang sudah tersedia.

Salah satu *library* yang dapat digunakan untuk menghasilkan *qr code* adalah PHP *qr code*. *Library* ini dikembangkan berdasarkan *library libqrencode* C, yang menyediakan API (*Application Programming Interface*) untuk membentuk gambar *qr code* dan diimplementasikan pada PHP.[2]

Fitur – fitur yang terdapat pada *library php qr code* diantaranya :

1. Mendukung versi *qr code* 1 sampai 40
2. Mendukung *encode* angka, huruf, 8 bit *encoding* dan kanji
3. Diimplementasikan di PHP
4. Dapat diekspor ke format png dan jpeg
5. Mudah dalam mengkonfigurasi
6. Merupakan *library open source*

### 2.4 QR Code Scanner

Scanner merupakan proses pemindaian terhadap QR Code menggunakan alat pemindai untuk membaca file data yang terdapat pada kode tersebut. Scanner ini berfungsi sebagai jembatan penghubung antar file. QR Code Scanner ini akan membaca QR Code, kemudian menampilkan hasilnya baik itu berupa text maupun gambar. Untuk menscan QR Code dalam pemrograman android bisa menggunakan *library ZXing* atau *ZBar*.

*ZXing* merupakan sebuah *library* pengolahan citra barcode 1D/2D yang diterapkan dalam bahasa pemrograman Java. *Library*

ini dikembangkan dengan fokus agar dapat melakukan pemindaian barcode menggunakan kamera yang terpasang pada sebuah alat tanpa harus berkomunikasi dengan sebuah server. Akan tetapi library ini juga dapat digunakan untuk menyandikan dan menerjemahkan sebuah barcode pada desktop dan server. Pada versi terbarunya ZXing telah mendukung format-format berikut: UPC-A, UPC-E, EAN-8, EAN-13, Code 39, Code 93, Code 128, ITF, Codabar, RSS-14 (semua jenis), RSS Expanded (hampir semua jenis), QR Code, Data Matrix, Aztec ('beta' quality), dan PDF 417 ('alpha' quality). [3]

ZBar merupakan sebuah library untuk pemindaian barcode melalui berbagai cara, dengan video streams, file gambar dan sensor. Mendukung banyak jenis barcode meliputi EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5 dan QR Code. Zbar juga menyediakan library untuk pengembangan aplikasi berbasis mobile Android maupun iOS. [4]

Kelebihan ZXing : mendukung banyak format, pemindaian cepat dan akurasi tinggi. Kekurangan ZXing : Sulit untuk diintegrasikan, hanya mendukung satu orientasi QR Code pada satu waktu[5]. Sedangkan kelebihan ZBar : pemindaian sangat cepat, mudah untuk diintegrasikan, mendukung orientasi QR Code yang berbeda. Kekurangan ZBar : akurasi rendah, dokumentasi terbatas.[6]

## 2.5 Android

*Android* adalah sistem operasi yang dirancang untuk perangkat bergerak dengan layar sentuh seperti smartphone atau tablet dengan didukung kernel yang berbasis *Linux*. *Android* bersifat *open source* sehingga memungkinkan untuk dimodifikasi secara bebas dan didistribusikan oleh pembuat perangkat tersebut. *Android* juga memudahkan pengembang untuk membangun aplikasi baru dan inovatif seperti pengembang dapat menggabungkan informasi yang diperoleh dari *web* dengan data yang ada pada ponsel.

*Android* dibangun benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. Selain itu *Android* menyediakan akses yang sangat luas kepada pengguna untuk menggunakan *library* yang diperlukan dan *tools* yang dapat digunakan untuk membangun aplikasi yang lebih baik. Hal ini tentu akan mempermudah pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.[7]

## 2.6 Laravel

Laravel adalah kerangka kerja aplikasi web dengan sintaks yang ekspresif, jelas dan menghemat waktu yang bertujuan untuk membuat proses pengembangan tanpa mengorbankan fungsionalitas aplikasi. Laravel yang dikembangkan pertama kali oleh Taylor Otwell bekerja dengan konsep MVC (model-view-controller) yang merupakan pengembangan website berbasis MVP (model-view-presenter) yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak.[8] Walaupun termasuk pemain baru, namun komunitas pengguna laravel sudah berkembang pesat dan mampu menjadi alternatif utama dari sejumlah framework besar seperti CodeIgniter & Yii. Laravel oleh para developer disetarakan dengan CodeIgniter dan FuelPHP namun memiliki keunikan tersendiri dari sisi coding yang lebih ekspresif dan elegan.[9]

Beberapa fitur yang ada di Laravel :

- a. Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
- b. Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang mengatasi masalah pada hubungan objek database.
- c. Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
- d. Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.

- e. Restful controllers, memisahkan logika dalam melayani https GET and POST.
- f. Class Auto Loading, menyediakan loading otomatis untuk class PHP.
- g. View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading.
- h. IoC Container, memungkinkan obyek baru dihasilkan dengan pembalikan controller.
- i. Migration, menyediakan sistem kontrol untuk skema database.
- j. Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi.
- k. Automatic Pagination, menyederhanakan tugas dari penerapan halaman.

## 2.7 Composer

Composer adalah aplikasi package manager untuk bahasa pemrograman PHP yang menyediakan format standar untuk mengelola dependensi PHP dan pustaka-pustaka yang diperlukan. Composer dikembangkan oleh Nils Adermann dan Jordi Boggiano, yang terus mengelola proyek. Mereka mulai pengembangan pada bulan April 2011 dan merilis Composer pertama kali pada tanggal 1 Maret 2012. Composer sangat terinspirasi oleh package manager Node.js "npm" dan Ruby "bundler".

Composer berjalan melalui baris perintah dan menginstal dependensi (pustaka perangkat lunak) untuk aplikasi. Hal ini juga memungkinkan pengguna untuk menginstal aplikasi PHP yang tersedia di "Packagist" yang mana itu adalah repositori utamanya. Hal ini juga menyediakan kemampuan autoload untuk pustaka yang menentukan autoload informasi untuk kemudahan penggunaan kode pihak ketiga. [10]

## 2.8 PHP

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia). PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group.

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: Hypertext Preprocessor.

PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek Open Source. [11]

## 2.9 Java

Java adalah bahasa pemrograman berbasis objek atau disebut “OOP” yang merupakan singkatan dari object oriented programming atau pemrograman yang berorientasi pada objek. Bahasa ini merupakan adopsi dari bahasa pemrograman C dan C++ dimana sintaks pada Java dan kedua bahasa pemrograman ini hampir sama namun Java memiliki keunggulan yaitu dapat dijalankan pada beberapa platform sistem operasi yang berbeda.[12]



Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* sistem operasi komputer, sehingga pemrogram cukup menulis sebuah program Java dan dikompilasi sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri yang dapat diunduh dari situs Java untuk menginterpretasikan *bytecode* tersebut.

## 2.10 MVC (Model-View-Controller)

MVC (*Model-View-Controller*) adalah pola desain perangkat lunak untuk mengembangkan aplikasi *web*. Pola *Model-View-Controller* terdiri dari tiga komponen, yaitu *model*, *view*, dan *controller*. Setiap komponen tersebut dibuat untuk menangani pengembangan aspek spesifik dari sebuah aplikasi. *Model* adalah tingkat terendah yang bertanggung jawab untuk memelihara data. *View* bertanggung jawab untuk menampilkan semua atau sebagian data kepada pengguna. *Controller* merupakan bagian kode perangkat lunak yang mengendalikan interaksi antara *model* dan *view*. [13]

## 2.11 API

API (*Application Programming Interfaces*) adalah alat yang membuat data situs web dapat dicerna untuk komputer. Karakteristik yang membuat situs web optimal bagi manusia, namun membuat sulit digunakan oleh komputer sehingga dibutuhkanlah sebuah API. Melalui API, komputer dapat melihat dan mengedit data, sama seperti seseorang yang dapat memuat halaman dan submit formulir. Apa yang mungkin membutuhkan waktu berjam-jam dengan manusia untuk menyelesaikannya, namun dapat memakan waktu beberapa detik dengan komputer melalui API.

Ketika dua sistem terhubung melalui API, dapat dikatakan bahwa dua sistem tersebut terintegrasi. Satu sisi yaitu server dan

sisi lain yaitu klien. Dalam sisi server sebenarnya yang menyediakan API. Jika dalam sisi klien, dapat mengetahui data apa yang tersedia melalui API dan dapat memanipulasinya, biasanya atas permintaan dari pengguna. Request method nantinya akan memberitahu server tindakan apa yang diinginkan pengguna untuk diambil oleh server.[14] Terdapat empat method yang paling sering digunakan di API adalah :

1. GET - Meminta server untuk mengambil data
2. POST - Meminta server untuk create data
3. PUT - Meminta server untuk mengedit / memperbarui data
4. DELETE - Meminta server untuk menghapus data

## 2.12 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh komputer. Format ini merupakan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 –Desember 1999. JSON merupakan format teks yang menggunakan gaya bahasa yang umum digunakan oleh programmer, seperti bahasa C, C++, C#, Java, JavaScript, Python, dll. JSON terbuat dari dua struktur:

Kumpulan pasangan nama atau nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai object, record, struct, dictionary, hash table, keyed list, atau associative array.

Daftar nilai terurutkan (an ordered list of values). Pada beberapa bahasa, hal ini dinyatakan sebagai array, vector, list, atau sequence.

Struktur data ini disebut sebagai struktur data universal, karena format data mudah dipertukarkan dengan semua bahasa pemrograman modern yang juga berdasarkan pada struktur data ini. [15] JSON menggunakan bentuk sebagai berikut:

- Objek, adalah sepasang nama atau nilai yang tidak terurutkan. Penulisan objek dimulai dengan kurung kurawal buka ( { ) dan di akhir dengan kurung kurawal

tutup ( } ). Lalu setiap nama diikuti dengan titik dua (:), serta setiap pasangan nama atau nilai dipisahkan oleh koma (,).

- Array, adalah sekumpulan nilai yang terurutkan. Penulisan array dimulai dengan kurung siku buka ( [ ) dan diakhiri dengan kurung siku tutup ( ] ), serta setiap nilai dipisahkan oleh koma (,).
- Nilai (value) biasanya berupa string dalam tanda kutip ganda ("), atau integer, juga dapat berupa true atau false atau null, serta dapat berupa sebuah objek atau array.
- String, adalah kumpulan dari beberapa karakter Unicode yang berada dalam tanda kutip ganda. String sangat mirip dengan string pada bahasa C atau Java.
- Angka, juga mirip dengan bahasa C atau Java, kecuali format oktal dan heksadesimal yang tidak digunakan.

### **2.13 Perbandingan Aplikasi Sejenis**

Terdapat implementasi sistem aplikasi registrasi tamu sejenis sebelumnya, yaitu Buku Tamu Digital dan Eventbrite. Pada Buku Tamu Digital, tamu akan mendapatkan sebuah tiket yang di dalamnya terdapat Barcode. Barcode ini berisi kode unik yang dapat membedakan tamu yang satu dengan tamu lainnya. Sebelum event dimulai, petugas scanner akan menscan Barcode yang ada pada tiket menggunakan alat scanner. Kemudian tamu akan mendapatkan sebuah label dari petugas scanner. Aplikasi Buku Tamu Digital dalam pengelolaan maupun pencatatan kehadiran sudah bagus dan cepat, tetapi kita perlu mengeluarkan biaya yang cukup mahal untuk menyewa sistem tersebut. Buku Tamu Digital juga masih memakai banyak kertas untuk menjalankan sistemnya yaitu kertas pada tiket dan label yang diberikan.

Sedangkan pada Eventbrite, pemilik event dapat membuat event sendiri melalui web/mobile app Eventbrite. Pemilik event juga dapat mengelola acara, mengelola data tamu, dan melihat buku tamu menggunakan web/mobile app Eventbrite. Untuk mendapatkan tiket, tamu dapat melakukan registrasi event pada

web/mobile app eventbrite dan tiket akan dikirim melalui email. Pemilik event juga dapat mengirim tiket kepada tamu yang dipilih dengan cara mengisi data tamu yang disertai email, kemudian tiket akan dikirim ke email tersebut dalam bentuk file pdf yang disertai QR Code. Kemudian untuk mencatat kehadiran dapat dilakukan dengan menscan QR Code pada tiket menggunakan kamera *smartphone* android. Eventbrite ini memiliki banyak fitur yang didukung dengan pemrosesan data yang cepat. Namun aplikasi Eventbrite ini cukup sulit dipahami karena selain menggunakan bahasa inggris, terdapat fitur yang mungkin akan tidak terpakai seperti fitur kebijakan pengembalian dana tiket dan menambahkan pajak pada pembelian tiket. Selain itu dalam pembuatan *event* berbayar, Eventbrite akan mengenakan *fee* atau biaya servis dalam pembelian tiket.

Dari perbandingan antara Buku Tamu Digital dan Evenbrite, penulis akan mencoba mengembangkan aplikasi sejenis dengan tujuan memudahkan pemilik *event* dalam mengelola *event*, mengurangi jumlah antrian saat proses pencatatan kehadiran dan mengurangi biaya dan penggunaan kertas pada saat proses pencatatan kehadiran. Untuk mempermudah penyebutan, penulis memberi nama aplikasi tersebut dengan nama Kotakevent. Perbandingan lebih rinci dengan aplikasi sejenis dapat dilihat di Tabel 2.1

**Tabel 2.1 Perbandingan Aplikasi Sejenis**

No	Fitur	Buku Tamu Digital	Evenbrite	Kotakevent
1	Aplikasi dapat mengunduh tiket event	Ya	Ya	Ya
2	Aplikasi dapat mengirimkan tiket melalui email dalam format pdf	Ya	Ya	Ya

No	Fitur	Buku Tamu Digital	Evenbrite	Kotakevent
3	Aplikasi dapat mengelola pendaftaran event	Tidak	Ya	Ya
4	Aplikasi dapat mengelola data tamu event	Ya	Ya	Ya
5	Aplikasi dapat mengunduh daftar tamu dalam bentuk pdf	Ya	Ya	Ya
6	Aplikasi dapat mencatat kehadiran peserta dengan menscan qr code tiket	Ya	Ya	Ya
7	Aplikasi dapat menambah data tamu yang telah hadir	Ya	Ya	Tidak
8	Aplikasi dapat membatasi jumlah pendaftar	Tidak	Ya	Tidak
9	Aplikasi dapat membatasi jumlah tiket	Tidak	Ya	Tidak
10	Aplikasi dapat membuat waitlist ketika semua tiket terjual atau kapasitas pendaftar sudah penuh	Tidak	Ya	Tidak
11	Aplikasi dapat mengubah pesan saat mengirim tiket melalui email	Ya	Ya	Tidak
12	Aplikasi dapat mempromosikan event ke facebook dan instagram	Tidak	Ya	Tidak

<b>No</b>	<b>Fitur</b>	<b>Buku Tamu Digital</b>	<b>Evenbrite</b>	<b>Kotakevent</b>
13	Aplikasi dapat mengundang tamu melalui email	Tidak	Ya	Tidak
14	Aplikasi dapat mengelola metode pembayaran	Tidak	Ya	Tidak
15	Aplikasi mempunyai kebijakan refund policy atau pengembalian dana pembelian tiket	Tidak	Ya	Tidak
16	Aplikasi dapat menambahkan pajak pada pembelian tiket	Tidak	Ya	Tidak
17	Aplikasi dapat menambahkan kode diskon pada tiket	Tidak	Ya	Tidak
18	Aplikasi dapat melihat rekapitulasi uang penjualan tiket	Ya	Ya	Tidak
19	Aplikasi dapat mengimport data tamu dari file excel	Tidak	Tidak	Ya
20	Aplikasi dapat melakukan pencarian menggunakan kode event	Tidak	Tidak	Ya
21	Aplikasi dapat menolak / menyetujui pembuatan event	Tidak	Tidak	Ya
22	Aplikasi dapat mengelola data petugas scanner	Tidak	Tidak	Ya

Dari perbandingan aplikasi sejenis yaitu Buku Tamu Digital dan Evenbrite, dapat dilihat bahwa ada empat fitur yang hanya dimiliki Kotakevent yaitu sebagai berikut :

- a. Kotakevent menyediakan fitur import data tamu melalui file excel dengan format yang telah disediakan. Format import berisi *field*: nama tamu, email tamu, alamat tamu dan kategori tamu. Pemilik *event* dapat dengan mudah memasukkan data tamu-tamunya melalui file excel, pemilik event juga dapat memasukkan kategori tamu sesuai keinginan. Dengan menggunakan fitur ini tentu akan mempermudah pemilik *event* dalam mengolah data tamu yang akan menghadiri *event*.
- b. Kotakevent menyediakan fitur mengelola data petugas scanner. Dengan menggunakan fitur ini, pemilik *event* dapat memilih siapa saja yang dapat menjadi petugas scanner dengan cara menambahkannya menggunakan data email, alamat, dan password. Sehingga ini akan mempermudah saat proses pencatatan kehadiran karena pemilik *event* dapat menambah petugas scanner sesuai keinginannya.
- c. Kotakevent menyediakan fitur pencarian event menggunakan kode event. Fitur ini berfungsi agar event hanya bisa dicari dan dilihat oleh orang yang mempunyai kode event. Dengan tujuan agar tidak sembarang orang bisa mendaftar event, sehingga mempermudah pemilik *event* dalam memilih tamu yang mendaftar.
- d. Kotakevent menyediakan fitur untuk menyetujui / menolak event yang telah dibuat. Fitur ini bertujuan untuk memilih event yang akan ditampilkan di Kotakevent sehingga tidak sembarang event bisa ditampilkan.

*[Halaman ini sengaja dikosongkan]*



## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab analisis dan perancangan berisi pembahasan analisis proses bisnis dan perancangan sistem yang akan dibangun.. Perancangan sistem akan mencantumkan analisis kebutuhan yang diperlukan di perangkat lunak setelah mendapatkan proses bisnis. Perancangan sistem ini akan direpresentasikan dengan diagram *Unified Modelling Language* (disebut UML).

#### **3.1 Analisis Sistem**

Proses registrasi tamu pada sebuah acara biasanya dilakukan dengan cara menuliskan data tamu undangan satu per satu pada buku tamu. Hal ini tentunya akan menjadi masalah jika jumlah tamu undangan mencapai ratusan atau ribuan. Dimana selain memerlukan waktu yang lama, keakuratan pendataan juga tidak terjamin. Proses registrasi yang cepat dan efisien sangat diperlukan untuk mempermudah pemilik acara dalam mengelola tamu-tamunya. Sistem registrasi yang cepat dan efisien sejauh ini menggunakan QR Code untuk memvalidasi kehadiran. Dengan QR Code, proses registrasi tamu dapat dilakukan dengan lebih cepat dan akurat. Namun sistem ini membutuhkan biaya yang tidak murah karena harus membayar sesuai dengan jumlah undangan yang ada dan masih menggunakan banyak kertas dalam pembuatan undangan serta alat scanner untuk menscan QR Code.

Pada Tugas Akhir ini mengembangkan alternatif pada sistem registrasi tamu yang praktis, efisien dan menghabiskan biaya yang relatif lebih rendah dibandingkan dengan sistem registrasi tamu yang sudah ada. Dengan memanfaatkan perkembangan teknologi informasi, khususnya teknologi perangkat bergerak dan internet, penggunaan alat scanner dapat diganti menggunakan kamera pada smartphone android, dan kertas undangan dapat diganti menggunakan file pdf. Sehingga, pemilik acara dapat mengelola

tamu-tamunya dengan mudah dan proses registrasi yang dilakukan menjadi lebih sederhana, praktis dan cepat.

## **3.2 Analisis Proses Bisnis**

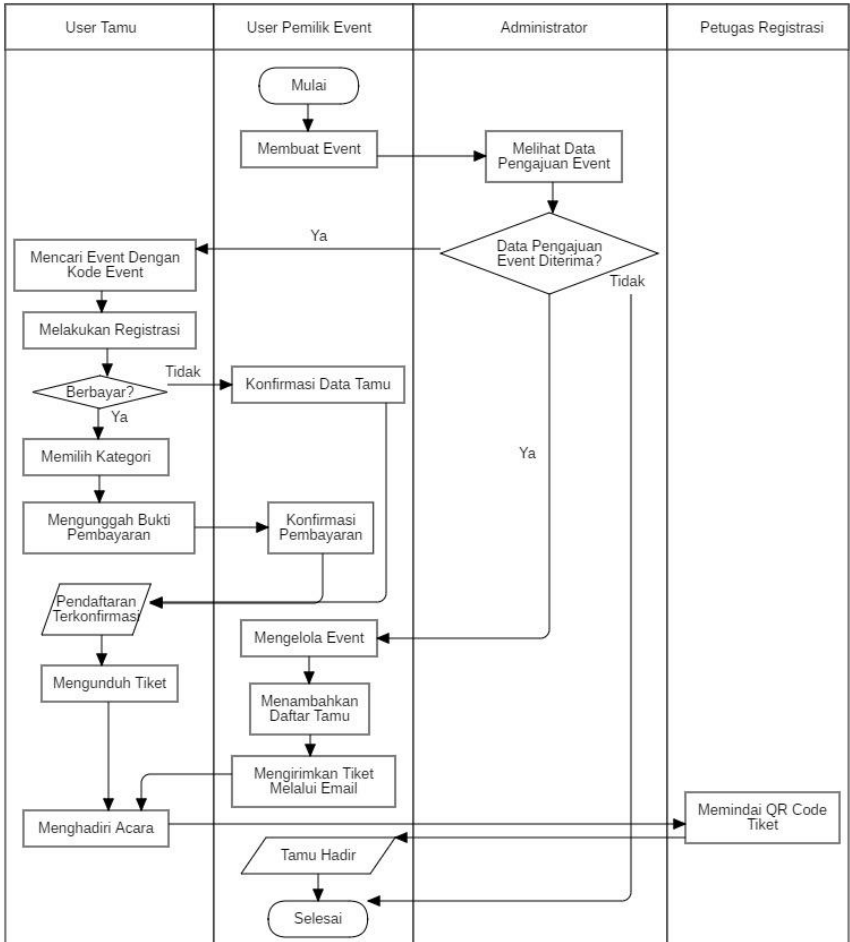
Analisis proses bisnis Kotakevent dapat dijabarkan sebagai berikut:

### **3.2.1 Proses Bisnis Kotakevent Secara Umum**

Untuk membuat *event*, pemilik *event* harus melakukan *register* akun terlebih dahulu ke Kotakevent. Setelah akun terdaftar, pemilik *event* dapat membuat *event* dengan memilih menu “Buat Event”. Terdapat dua jenis *event* yang dapat dipilih, yaitu *open event* dan *close event*. *Open event* berarti tamu dapat melakukan registrasi sendiri ke *event* melalui akun Kotakevent dan tamu juga dapat ditambahkan oleh pemilik *event*, sedangkan *close event* berarti tamu hanya dapat ditambahkan oleh pemilik *event* saja. Setelah *event* berhasil dibuat, pemilik *event* harus menunggu persetujuan admin untuk pengajuan pembuatan *event*. *Event* dapat dikelola oleh pemilik *event* jika pengajuan pembuatan *event* telah disetujui oleh admin.

Untuk *open event*, tamu dapat melakukan registrasi sendiri dengan cara mencari *event* yang ingin didaftar, pencarian bisa menggunakan nama atau kode *event*. Setelah memilih *event* yang ingin didaftar, tamu dapat menekan tombol “Daftar” dan tamu akun langsung terdaftar jika *event* tersebut tidak berbayar, jika berbayar maka tamu harus memilih kategori terlebih dahulu dan mengunggah bukti pembayaran untuk mendapatkan tiket. Untuk *close event* tamu hanya dapat ditambahkan pemilik *event* melalui *import file excel* atau ditambahkan satu per satu menggunakan form data tamu. Setelah berhasil ditambahkan, pemilik *event* dapat mengirim tiket dengan cara menekan tombol “Kirim Tiket” pada data tamu yang dipilih. Kemudian tiket akan terkirim ke email data tamu dalam bentuk file PDF.

Setelah mendapatkan tiket, tamu dapat membawa tiket tersebut saat *event* berlangsung. Kemudian petugas scanner akan memindai qr code pada tiket untuk mencatat kehadiran tamu. Jika berhasil, maka tamu akan dipersilakan untuk memasuki *event*. Proses bisnis *event* secara umum akan dijelaskan Gambar 3.1



**Gambar 3.1** Proses Bisnis Kotakevent Secara Umum

### 3.2.2 Tipe Event

Terdapat 5 tipe event yang dapat dibuat di Kotakevent, yaitu open berbayar umum, open berbayar kode, open tidak berbayar umum, open tidak berbayar kode dan close event dengan penjelasan sebagai berikut :

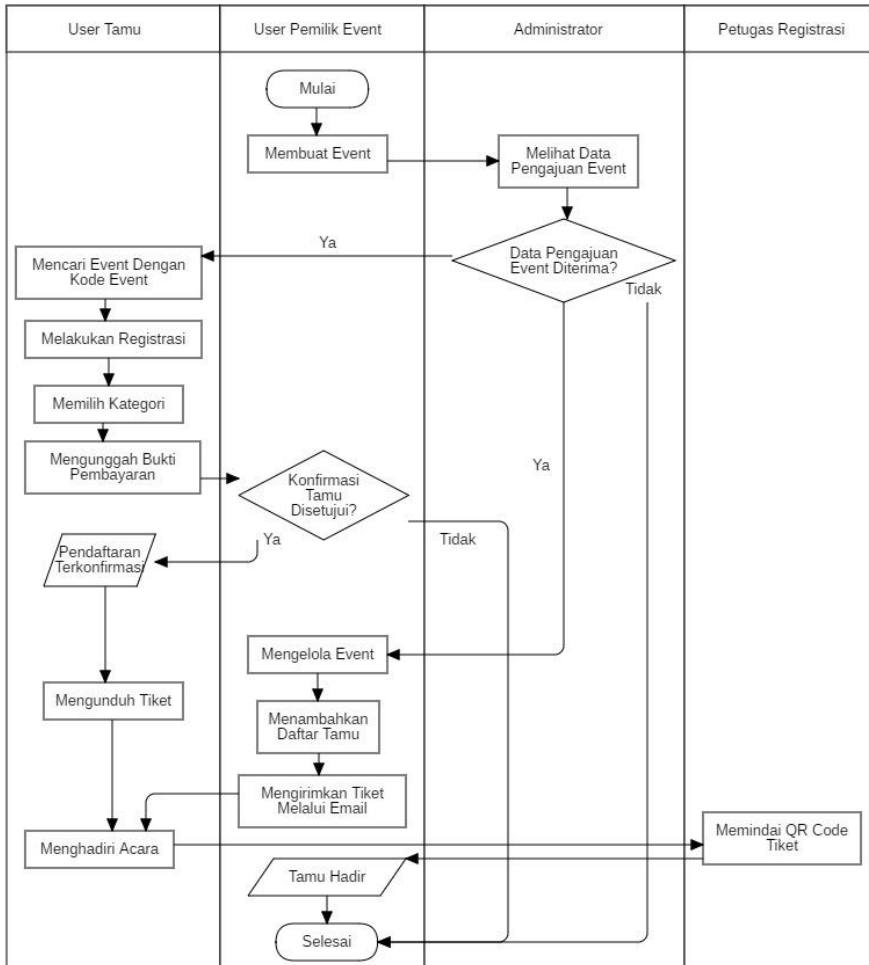
- Open berarti tamu dapat melakukan registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event.
- Close berarti tamu tidak dapat melakukan registrasi melalui akun Kotakevent dan tamu hanya dapat ditambahkan oleh pemilik event.
- Tidak Berbayar berarti event tersebut tidak berbayar, dan tamu bisa melakukan registrasi tanpa perlu mengunggah bukti pembayaran.
- Berbayar berarti event tersebut berbayar, dan tamu harus mengunggah bukti pembayaran untuk melakukan registrasi
- Umum berarti event tersebut dapat dicari dan dilihat oleh umum
- Kode berarti event tersebut hanya dapat dicari dan dilihat dengan menggunakan kode.

#### 3.2.2.1 Open Berbayar Umum

Tamu dari event tipe ini bisa berasal dari dua jalur, yaitu tamu bisa melakukan registrasi sendiri ke *event* menggunakan akun Kotakevent dan tamu juga bisa ditambahkan oleh pemilik *event*.

*Event* pada tipe ini dapat dicari menggunakan nama *event* pada halaman pencarian. Untuk melakukan registrasi, tamu harus memilih kategori dan mengunggah bukti pembayaran terlebih dahulu. Setelah disetujui pemilik *event* maka tamu baru bisa

mengunduh tiket yang nantinya dapat digunakan saat *event* berlangsung.

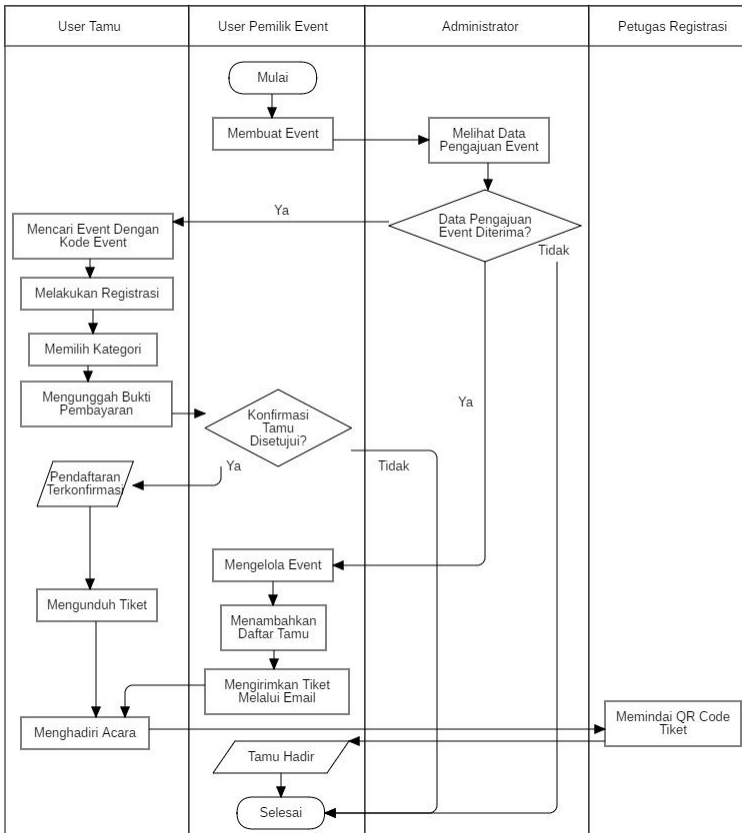


**Gambar 3.2** Proses Bisnis Event Tipe Open Berbayar Umum

### 3.2.2.2 Open Berbayar Kode

Tamu dari event tipe ini bisa berasal dari dua jalur, yaitu tamu bisa melakukan registrasi sendiri ke *event* menggunakan akun Kotakevent dan tamu juga bisa ditambahkan oleh pemilik *event*.

*Event* pada tipe ini hanya dapat dicari menggunakan kode *event* pada halaman pencarian. Untuk melakukan registrasi, tamu harus memilih kategori dan mengunggah bukti pembayaran terlebih dahulu. Setelah disetujui pemilik *event* maka tamu baru bisa mengunduh tiket yang nantinya dapat digunakan saat *event* berlangsung.

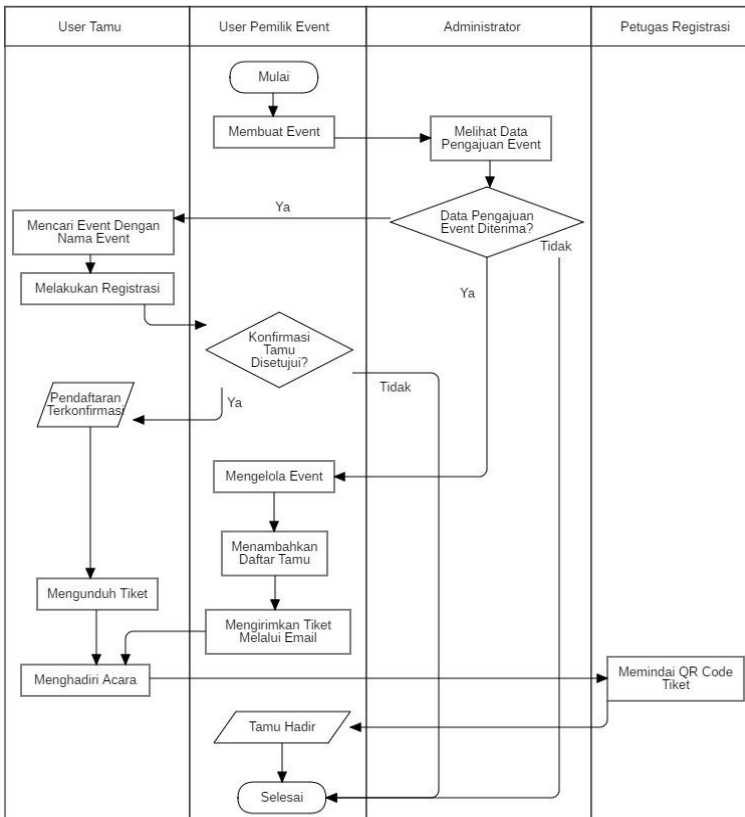


**Gambar 3.3** Proses Bisnis Event Tipe Open Berbayar Kode

### 3.2.2.3 Open Tidak Berbayar Umum

Tamu dari event tipe ini bisa berasal dari dua jalur, yaitu tamu bisa melakukan registrasi sendiri ke *event* menggunakan akun Kotakevent dan tamu juga bisa ditambahkan oleh pemilik event.

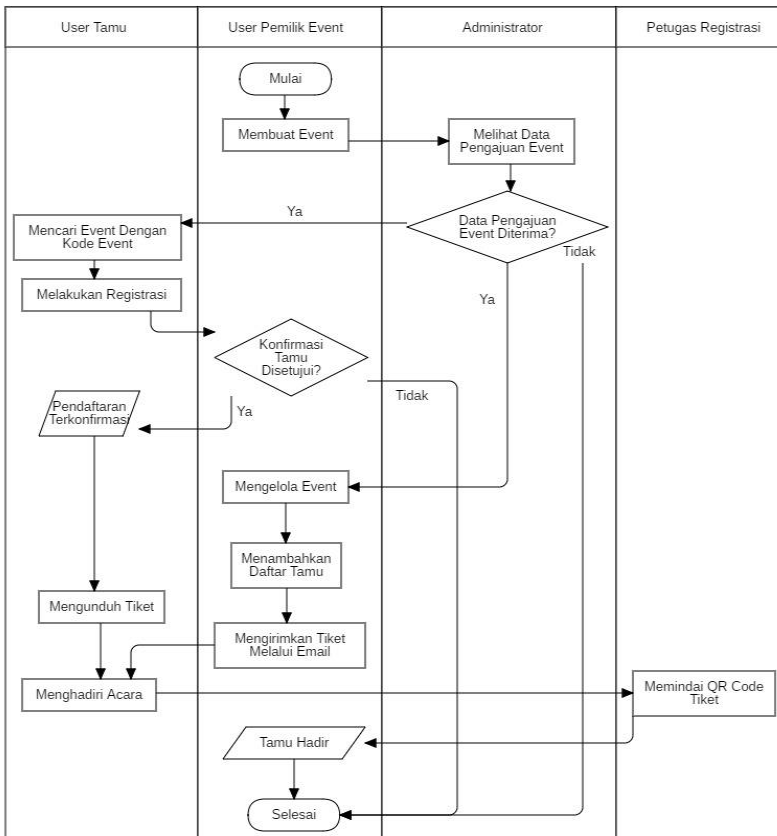
*Event* pada tipe ini dapat dicari menggunakan nama *event* pada halaman pencarian. Untuk melakukan registrasi, tamu hanya perlu menekan tombol “Daftar Event” pada halaman detail *event* dan menunggu persetujuan dari pemilik *event*. Setelah disetujui pemilik *event* maka tamu baru bisa mengunduh tiket yang nantinya dapat digunakan saat *event* berlangsung.



**Gambar 3.4** Proses Bisnis Event Tipe Open Tidak Berbayar Umum

### 3.2.2.4 Open Tidak Berbayar Kode

Tamu dari event tipe ini bisa berasal dari dua jalur, yaitu tamu bisa melakukan registrasi sendiri ke *event* menggunakan akun Kotakevent dan tamu juga bisa ditambahkan oleh pemilik event. *Event* pada tipe ini hanya dapat dicari menggunakan kode *event* pada halaman pencarian. Untuk melakukan registrasi, tamu hanya perlu menekan tombol “Daftar Event” pada halaman detail *event* dan menunggu persetujuan dari pemilik *event*. Setelah disetujui pemilik *event* maka tamu baru bisa mengunduh tiket yang nantinya dapat digunakan saat *event* berlangsung.



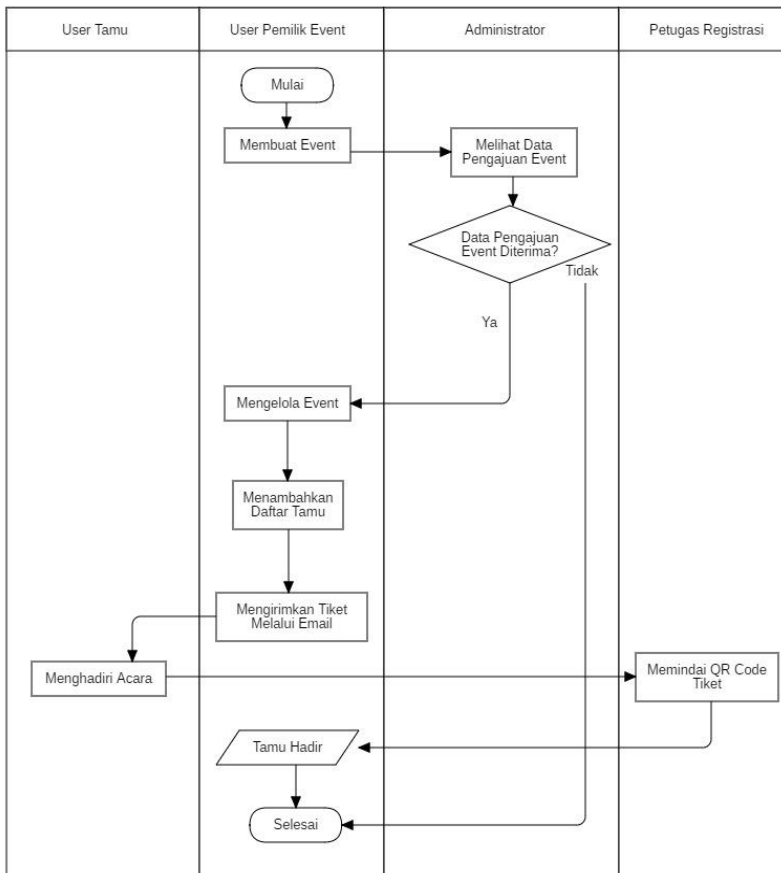
Gambar 3.6 Proses Bisnis Event Tipe Open Tidak Berbayar Kode



### 3.2.2.5 Close Event

Tamu dari *event* tipe ini hanya berasal dari satu jalur, yaitu tamu hanya bisa ditambahkan oleh pemilik event melalui *import file excel* atau ditambahkan satu per satu menggunakan form data tamu.

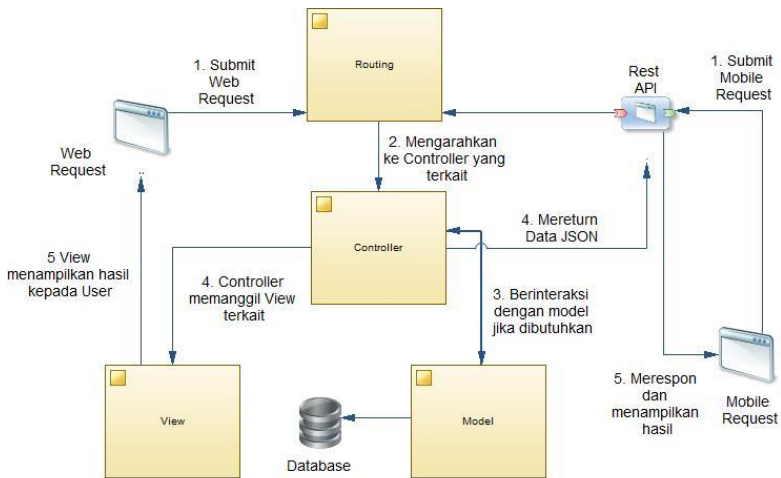
Setelah berhasil ditambahkan, pemilik *event* dapat mengirim tiket dengan cara menekan tombol “Kirim Tiket” pada data tamu yang dipilih. Kemudian tiket akan terkirim ke email data tamu dalam bentuk file PDF.



**Gambar 3.7 Proses Bisnis Event Tipe Close Event**

### 3.3 Deskripsi Umum Sistem

Arsitektur sistem yang digunakan dalam Tugas Akhir ini yaitu menggunakan kerangka kerja (*framework*) Laravel dengan pola perancangan *Model-View-Controller* (MVC) pada aplikasi web dan aplikasi mobile menggunakan android dengan bahasa pemrograman java. Ilustrasi dari arsitektur sistem dapat dilihat pada Gambar 3.3



Gambar 3.8 Ilustrasi Arsitektur Sistem

Ketika melakukan *request* data pada halaman web, *Routing* akan menerima *request* tersebut dan mengarahkan *request* tersebut kepada *controller* yang terkait dengan *request* yang diberikan. *Controller* akan berinteraksi dengan model yang dibutuhkan, dan model akan melakukan *query* terhadap database. Model mengembalikan data yang dibutuhkan kepada *controller*. Kemudian *controller* memanggil *view* terkait, untuk menampilkan hasil yang diminta oleh user melalui browser.

Ketika melakukan *request* data pada *mobile*, *rest api* akan menerima *request* tersebut dan mengarahkannya ke *routing*. Dari *routing* akan mengarah ke *controller* dan berinteraksi dengan model yang dibutuhkan. Model akan melakukan *query* terhadap database dan mengembalikan data dalam bentuk json. Kemudian data tersebut akan direspon dan ditampilkan dalam halaman *mobile*.

### 3.3.1 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan cakupan perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak agar dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Diharapkan dengan adanya spesifikasi ini dapat menyesuaikan kebutuhan-kebutuhan pengguna. Spesifikasi kebutuhan perangkat lunak adalah penjelasan mengenai kebutuhan sistem yang diinginkan pelanggan atau klien dalam bentuk tulisan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional dan deskripsi aktor.

#### 3.3.1.1 Kebutuhan Fungsional

Kebutuhan Fungsional terkait aplikasi Kotakevent dapat dilihat pada Tabel 3.1

**Table 3.1 Kebutuhan Fungsional**

<b>Kode</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
F-01	Melakukan register akun	Melakukan pengisian form register akun berupa nama, email, alamat dan password
F-02	Mendaftar event	Melakukan pendaftaran pada sebuah event menggunakan akun kotakevent.

<b>Kode</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
F-03	Membuat event	Melakukan pengisian detail event berupa foto,nama, lokasi, tanggal mulai & selesai, jenis pendaftaran, pencarian event, kategori dan deskripsi.
F-04	Mengelola event	Melakukan pengelolaan terkait event yang telah dibuat. Pengelolaan ini berupa pengeditan event dan penambahan foto galery event.
F-05	Mengelola pendaftaran event	Melakukan pengelolaan terkait pendaftaran event. Pengelolaan ini berupa pembukaan / penutupan pendaftaran event.
F-06	Mengelola data tamu event	Melakukan pengelolaan terkait data tamu event. Pengelolaan ini berupa penambahan data tamu, penambahan kategori, pengubahan data tamu dan penghapusan data tamu
F-07	Mengunduh format import data tamu	Mengunduh format excel untuk mengimport data tamu
F-08	Mengimport data tamu	Menambahkan data tamu melalui file excel
F-09	Mengirim tiket event	Mengirimkan tiket event ke email data tamu yang ditambahkan
F-010	Mengelola data pengajuan pendaftar	Melakukan pengelolaan terkait data pendaftar event. Pengelolaan ini berupa persetujuan / penolakan data pendaftar.
F-011	Mengelola data pendaftar event	Melakukan pengelolaan terkait data pendaftar event. Pengelolaan ini berupa perubahan kategori pendaftar dan penghapusan data pendaftar.
F-012	Mengunduh data tamu terdaftar	Mengunduh data tamu yang terdaftar dalam event dengan format file pdf

<b>Kode</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
F-013	Mengelola data petugas registrasi	Melakukan pengelolaan terkait petugas registrasi. Pengelolaan ini berupa penambahan, pengubahan dan penghapusan data petugas registrasi.
F-014	Mengunggah Bukti Pembayaran	Mengunggah bukti pembayaran event yang telah didaftar
F-015	Mengunduh Tiket	Mengunduh tiket event. Tiket didapat setelah tamu terdaftar dalam acara.
F-016	Mencari event	Melakukan pencarian event yang telah disetujui admin. Pencarian dapat dilakukan menggunakan nama / kode event.
F-017	Melihat detail event	Menampilkan detail event berupa foto, nama, jadwal, deskripsi, dan galeri foto event.
F-018	Melihat daftar event	Menampilkan data event yang telah dibuat baik yang sudah/belum disetujui.
F-019	Melihat daftar tiket	Menampilkan data tiket yang sudah / belum disetujui. Data tiket di dapat setelah mendaftar ke sebuah event.
F-020	Melihat pengajuan pendaftar event	Menampilkan data pendaftar event yang telah melakukan pendaftaran dan belum dikonfirmasi pemilik event
F-021	Melihat daftar peserta	Menampilkan data peserta event yang telah melakukan pendaftaran dan sudah dikonfirmasi pemilik event
F-022	Melihat daftar hadir	Menampilkan data peserta event yang telah melakukan registrasi.
F-023	Melihat petugas registrasi	Menampilkan data petugas registrasi
F-024	Mengunduh daftar hadir	Mengunduh data tamu yang telah hadir atau telah melakukan registrasi
F-025	Mengelola pengajuan event	Melakukan pengelolaan terkait pengajuan pembuatan event. Pengelolaan berupa persetujuan / penolakan event.

<b>Kode</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
F-026	Melihat data pengajuan event	Menampilkan data pengajuan event. Data pengajuan event didapat setelah pembuatan event.
F-027	Melihat daftar event dibuat	Menampilkan semua data event baik yang dikonfirmasi atau ditolak
F-028	Melihat daftar event selesai	Menampilkan data event yang jadwalnya sudah selesai.
F-029	Memindai qr code tiket	Melakukan scan pada tiket untuk melakukan registrasi.
F-030	Melihat bukti pembayaran	Melihat bukti pembayaran tamu yang telah mendaftar ke <i>event</i> berbayar

### 3.3.1.2 Deskripsi Aktor

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Dalam sistem untuk tugas akhir ini aplikasi Kotakevent memiliki empat aktor yang dibahas lebih rinci pada Tabel 3.1

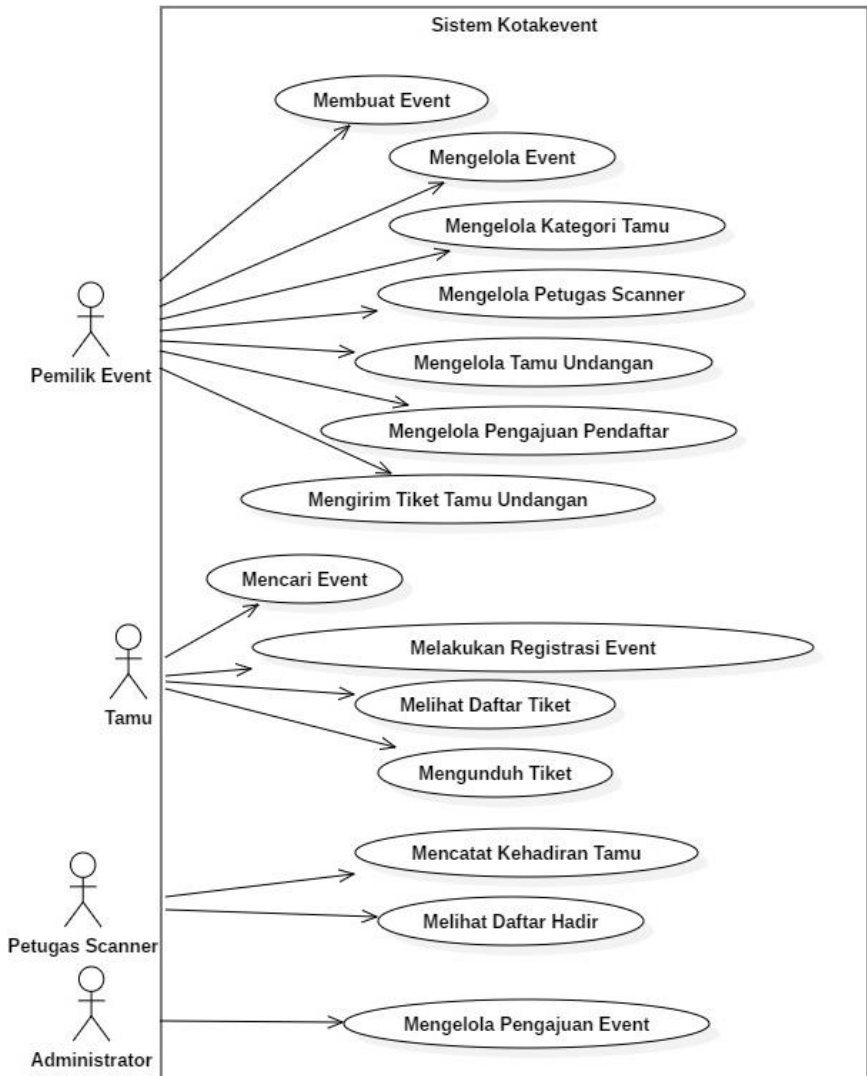
**Tabel 3.1 Deskripsi Aktor pada Aplikasi Kotakevent**

<b>No</b>	<b>Aktor</b>	<b>Deskripsi</b>
1	Administrator	<ul style="list-style-type: none"> <li>• Mengelola pengajuan event</li> <li>• Melihat data pengajuan event</li> <li>• Melihat daftar event dibuat</li> <li>• Melihat daftar event selesai</li> </ul>
2	Pemilik Event	<ul style="list-style-type: none"> <li>• Melakukan registrasi</li> <li>• Membuat event</li> <li>• Melihat daftar event</li> <li>• Mengelola event</li> <li>• Mengelola pendaftaran event</li> <li>• Mengelola data tamu event</li> <li>• Mengimport data tamu</li> <li>• Mengirim tiket event</li> </ul>

No	Aktor	Deskripsi
		<ul style="list-style-type: none"> <li>• Mengelola data pengajuan pendaftar</li> <li>• Mengelola data pendaftar event</li> <li>• Mengunduh daftar peserta</li> <li>• Mengelola petugas registrasi</li> <li>• Melihat daftar peserta</li> <li>• Melihat daftar hadir</li> <li>• Melihat bukti pembayaran</li> </ul>
3	Tamu	<ul style="list-style-type: none"> <li>• Melakukan registrasi</li> <li>• Mencari event</li> <li>• Melihat detail event</li> <li>• Mendaftar event</li> <li>• Melihat daftar tiket</li> <li>• Mengelola tiket event</li> </ul>
4	Petugas Scanner	<ul style="list-style-type: none"> <li>• Memindai qr code tiket</li> <li>• Melihat daftar hadir</li> <li>• Mencari peserta</li> </ul>

### 3.4 Kasus Penggunaan Kotakevent

Bagian ini menjelaskan secara rinci kasus penggunaan (*use case*, selanjutnya disingkat menjadi UC) yang terdapat pada perangkat lunak. Selain itu, terdapat juga spesifikasi kasus penggunaan dan diagram aktivitas pada tiap kasus penggunaan. Diagram kasus penggunaan Aplikasi Kotakevent ditunjukkan pada Gambar 3.9



**Gambar 3.9 Diagram Kasus Penggunaan Kotakevent**



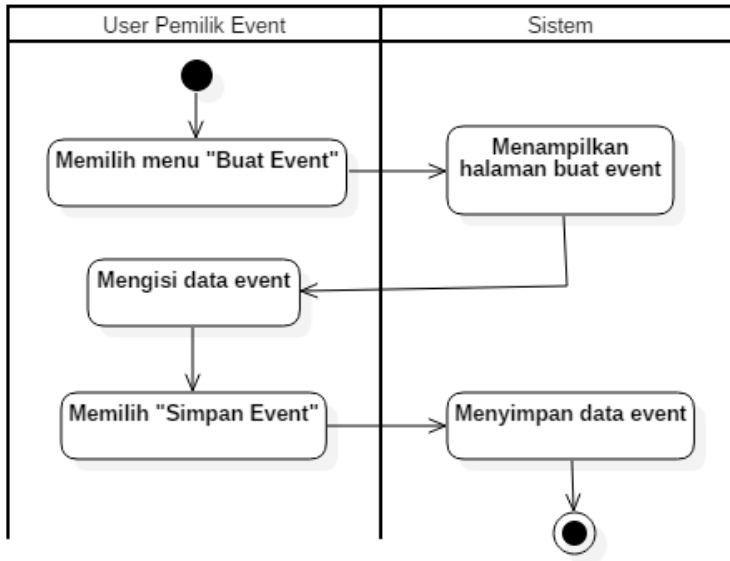
### 3.4.1 UC-01: Membuat Event

Pada kasus penggunaan ini, Pemilik Event dapat membuat event dengan mengisi form yang telah disediakan. Form event berisi foto, nama, lokasi, tanggal mulai & selesai, jenis event, jenis pendaftaran, pencarian event, kategori dan deskripsi. Setelah selesai pembuatan event, user harus menunggu konfirmasi admin terlebih dahulu agar event dapat dikelola. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.2 dan diagram aktivitas pada Gambar 3.10

**Tabel 3.2 Spesifikasi Kasus Penggunaan Membuat Event**

<b>Use Case</b>	<b>UC-01</b>		
<b>Nama Use Case</b>	Membuat Event		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk membuat dan mendaftarkan event ke aplikasi Kotakevent		
<b>Kondisi Awal</b>	Event belum terdaftar dalam Kotakevent		
<b>Kondisi Akhir</b>	Event telah terdaftar dalam Kotakevent		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Aktor memilih menu “Buat Event”.		
		2.	Menampilkan halaman buat <i>event</i> yang terdiri dari field: foto, nama, lokasi, jadwal mulai, jadwal selesai, jenis event, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i> .
3.	Mengisikan data yang diinginkan sesuai dengan field yang tertera		
4.	Memilih “Simpan Event”		

		5.	Menyimpan data event ke dalam sistem
Alur Alternatif	-		
Eksepsi	-		



**Gambar 3.10 Diagram Aktivitas Membuat Event**

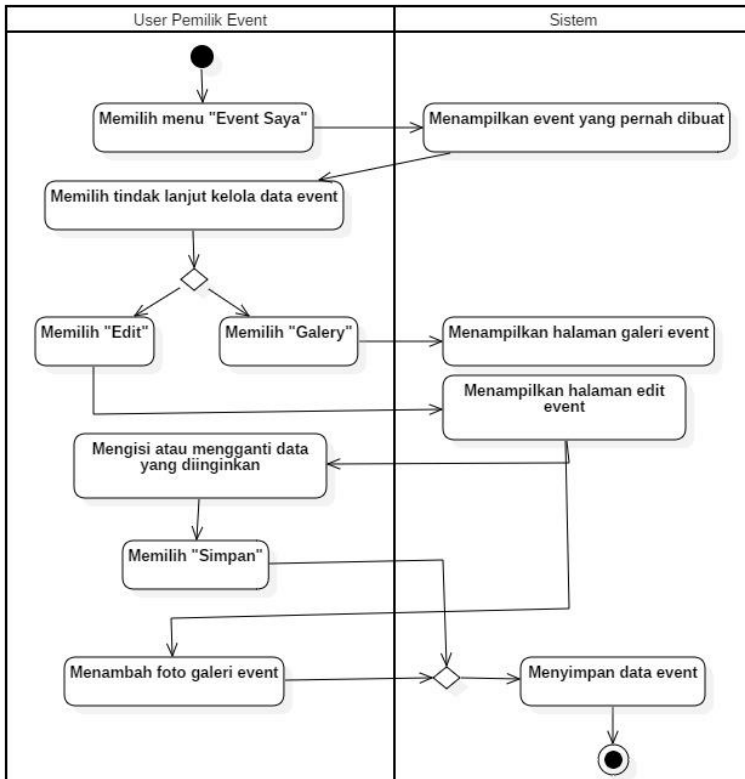
### 3.4.2 UC-02: Mengelola Event

Pada kasus penggunaan ini, User Pemilik Event dapat melakukan pengelolaan terhadap event. Pengelolaan berupa mengubah data event, dan menambah foto galeri *event*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.3 dan diagram aktivitas pada Gambar 3.11

Tabel 3.3 Spesifikasi Kasus Penggunaan Mengelola Event

<b>Use Case</b>	UC-02		
<b>Nama Use Case</b>	Mengelola Event		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk mengubah data <i>event</i> dan menambahkan foto galeri <i>event</i>		
<b>Kondisi Awal</b>	Data mengenai <i>event</i> masih sesuai dengan kondisi sebelum perubahan dilakukan.		
<b>Kondisi Akhir</b>	Data <i>event</i> yang diubah ataupun ditambah telah tersimpan ke dalam sistem		
<b>Alur Normal</b>			
	Aktor	Sistem	
1.	Aktor memilih menu “Event Saya”		
		2.	Menampilkan semua event yang pernah dibuat oleh aktor.
3.	Memilih “Edit” A1. Memilih “Galeri”		
		4.	Menampilkan halaman edit <i>event</i> yang terdiri dari field: nama <i>event</i> , lokasi <i>event</i> jadwal mulai, jadwal selesai, jenis <i>event</i> , jenis pendaftaran, pilihan pencarian dan deskripsi <i>event</i> .
5.	Mengisikan atau mengganti data yang diinginkan sesuai dengan field yang tertera.		
6.	Memilih “Simpan”		
		7.	Menyimpan data <i>event</i> ke dalam sistem
<b>Alur Alternatif</b>			
A1. Memilih “Galery”			

Aktor		Sistem	
		A1.1	Menampilkan halaman galeri <i>event</i>
A1.2	Memilih dan menambahkan foto galeri <i>event</i>		
		A1.3	Menyimpan data event ke dalam sistem
Eksepsi		-	



**Gambar 3.11 Diagram Aktivitas Mengelola *Event***

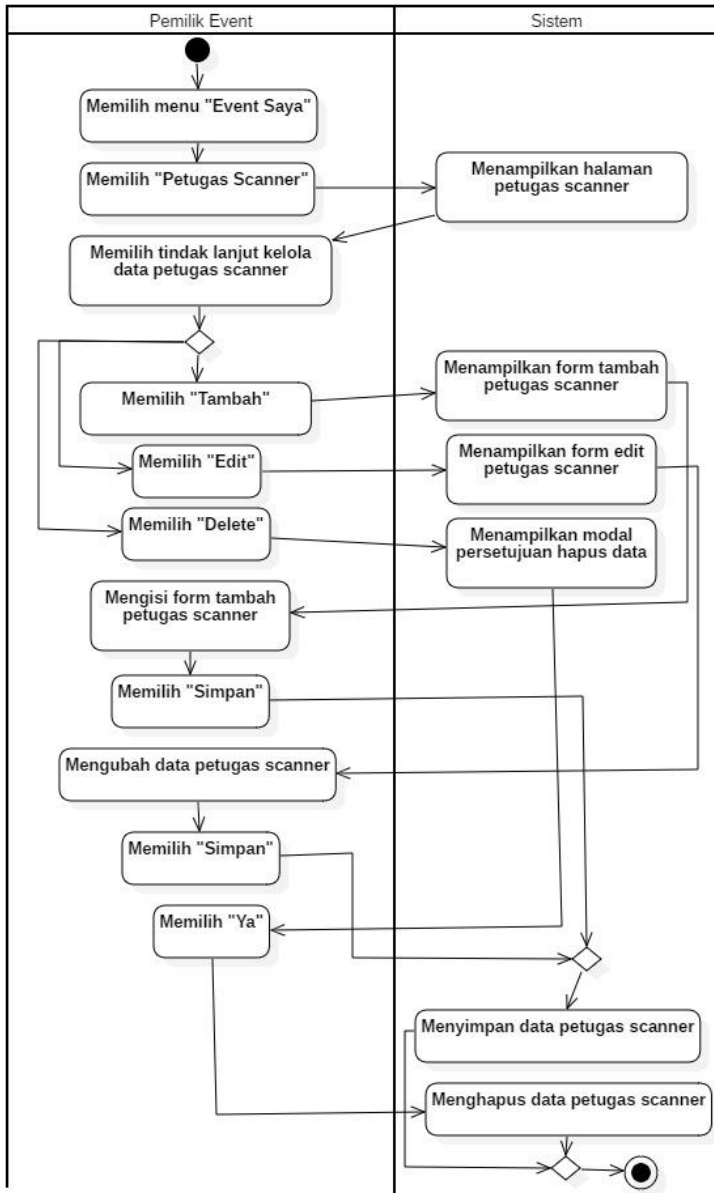
### 3.4.3 UC-03: Mengelola Petugas Scanner

Pada kasus penggunaan ini, User Pemilik Event dapat melakukan pengelolaan terhadap Petugas Scanner *Event*. Pengelolaan berupa menambah, mengubah dan menghapus data petugas scanner. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.4 dan diagram aktivitas pada Gambar 3.12

**Tabel 3.4 Spesifikasi Kasus Penggunaan Mengelola Petugas Scanner**

<b>Use Case</b>	UC-03		
<b>Nama Use Case</b>	Mengelola Petugas Scanner		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk menambahkan, mengubah dan menghapus data petugas scanner pada <i>event</i> .		
<b>Kondisi Awal</b>	Data petugas scanner belum dikelola.		
<b>Kondisi Akhir</b>	Data petugas scanner telah tersimpan ke dalam sistem		
<b>Alur Normal</b>			
	Aktor	Sistem	
1.	Aktor memilih menu “Event Saya”.		
2.	Aktor memilih “Petugas Scanner”		
		3.	Menampilkan halaman daftar petugas scanner
4.	Memilih “Tambah” A1. Memilih “Edit” A2. Memilih “Delete”		
		5.	Menampilkan form tambah Petugas Scanner
6.	Mengisikan nama, email, alamat dan password petugas scanner.		
7.	Memilih “Simpan”		

		8.	Menyimpan data petugas scanner ke dalam sistem.
<b>Alur Alternatif</b>			
A1. Memilih Edit			
Aktor		Sistem	
		A1.1	Sistem menampilkan form edit petugas scanner
A1.2	Mengubah data petugas scanner sesuai kebutuhan		
		A1.3	Menyimpan data petugas scanner ke dalam sistem
A2. Memilih Delete			
Aktor		Sistem	
		A2.1	Sistem menampilkan form persetujuan hapus data petugas scanner
A2.2	Memilih “Ya”		
		A2.3	Menghapus data petugas scanner
Eksepsi	-		



**Gambar 3.12 Diagram Aktivitas Mengelola Petugas Scanner**

### 3.4.4 UC-04: Mengelola Kategori Tamu

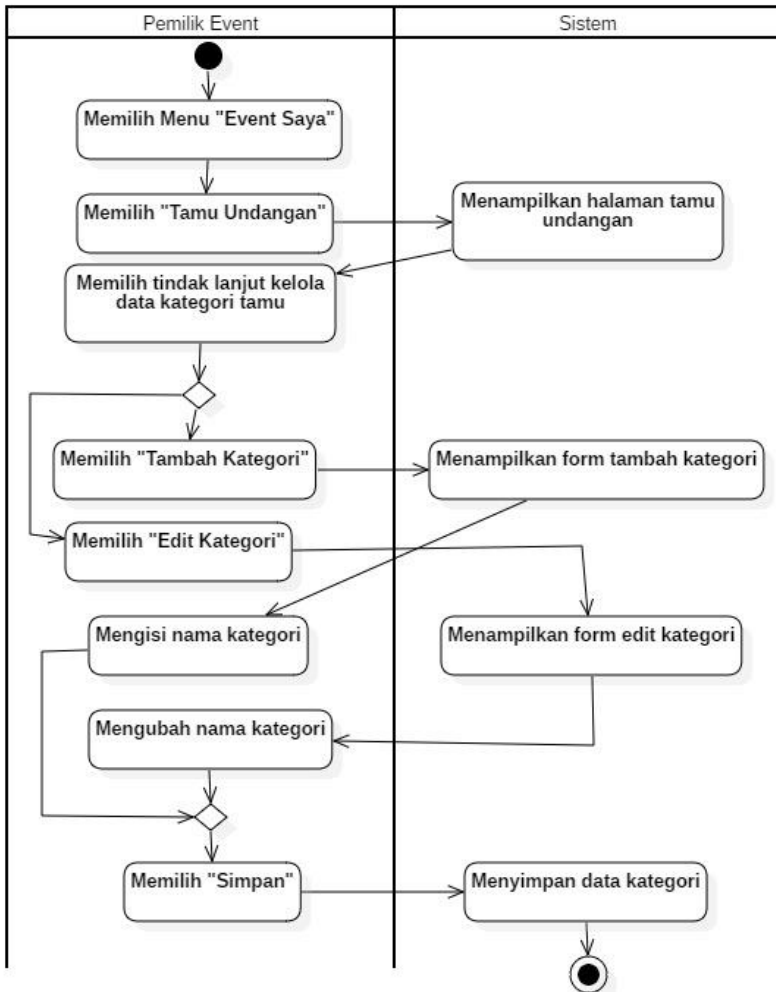
Pada kasus penggunaan ini, User Pemilik Event dapat melakukan pengelolaan terhadap kategori *event*. Pengelolaan berupa menambah kategori *event* dan mengubah kategori tamu. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.5 dan diagram aktivitas pada Gambar 3.13

**Tabel 3.5 Spesifikasi Kasus Penggunaan Mengelola Kategori Tamu**

<b>Use Case</b>	<b>UC-03</b>		
<b>Nama Use Case</b>	Mengelola Kategori Tamu		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk menambahkan dan mengubah kategori tamu dalam sebuah event.		
<b>Kondisi Awal</b>	Kategori tamu belum dikelola.		
<b>Kondisi Akhir</b>	Kategori tamu yang baru telah tersimpan di basis data.		
<b>Alur Normal</b>			
	Aktor		Sistem
1.	Aktor memilih menu “Event Saya”.		
2.	Aktor memilih “Tamu Undangan”		
		3.	Menampilkan halaman daftar tamu undangan dari event yang terpilih.
4.	Memilih “Tambah Kategori” A1. Memilih “Edit Kategori”		
		5.	Menampilkan form tambah kategori
6.	Mengisikan nama kategori tamu		
7.	Memilih “Simpan”.		



		8	Menyimpan data kategori tamu ke dalam sistem.
<b>Alur Alternatif</b> A1.Memilih “Edit Kategori”			
Aktor		Sistem	
		A1.1	Menampilkan form edit kategori
A1.2	Mengubah kategori tamu		
		A1.3	Menyimpan data kategori tamu ke dalam sistem
Eksepsi	-		



**Gambar 3.13 Diagram Aktivitas Mengelola Kategori Tamu**

### 3.4.5 UC-05: Mengelola Tamu Undangan

Pada kasus penggunaan ini, User Pemilik Event dapat melakukan pengelolaan terhadap Data Tamu Undangan. Pengelolaan berupa menambah, mengubah dan menghapus data tamu undangan. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.6 dan diagram aktivitas pada gambar 3.14

**Tabel 3.6 Spesifikasi Kasus Penggunaan Mengelola Tamu Undangan**

<b>Use Case</b>	<b>UC-05</b>		
<b>Nama Use Case</b>	Mengelola Tamu Undangan		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk menambahkan, mengubah dan menghapus data tamu undangan		
<b>Kondisi Awal</b>	Data tamu undangan belum dikelola.		
<b>Kondisi Akhir</b>	Data tamu undangan telah tersimpan ke dalam sistem		
<b>Alur Normal</b>			
	Aktor	Sistem	
1.	Aktor memilih menu “Event Saya”.		
2.	Aktor memilih “Tamu Undangan”		
		3.	Menampilkan halaman daftar tamu undangan dari event yang terpilih.
4.	Memilih “Tambah” A1. Memilih “Edit” A2. Memilih “Delete”		
		5.	Menampilkan form tambah data tamu , berisi field nama tamu, email tamu, alamat tamu, dan kategori tamu

6.	Mengisikan nama, email, alamat dan kategori tamu		
7.	Memilih “Simpan”.		
		8.	Menyimpan data tamu undangan ke dalam sistem.

**Alur Alternatif**  
A1. Memilih “Edit”

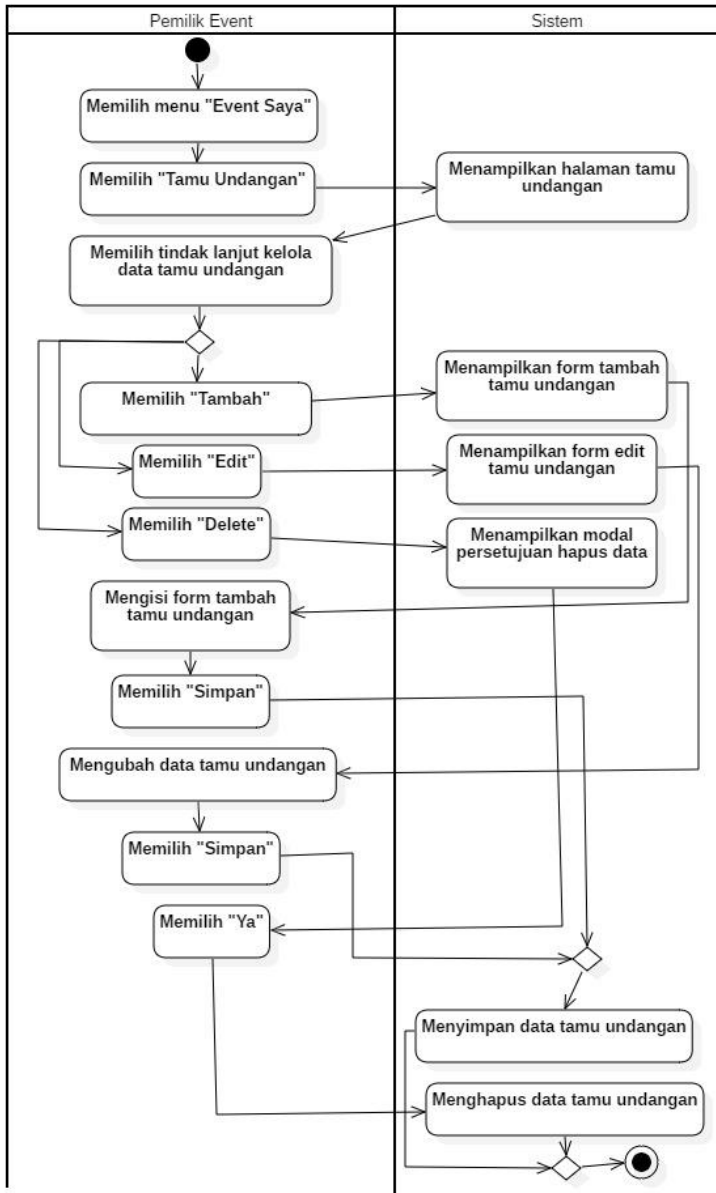
Aktor		Sistem	
		A1.1	Sistem menampilkan form edit tamu undangan
A1.2	Mengubah data tamu undangan sesuai kebutuhan		
		A1.3	Menyimpan data tamu undangan

A2. Memilih “Delete”

Aktor		Sistem	
		A2.1	Sistem menampilkan form persetujuan hapus data tamu undangan
A2.2	Memilih “Ya”		
		A2.3	Menghapus data tamu undangan

Eksepsi

-



**Gambar 3.14** Diagram Aktivitas Mengelola Data Tamu Undangan

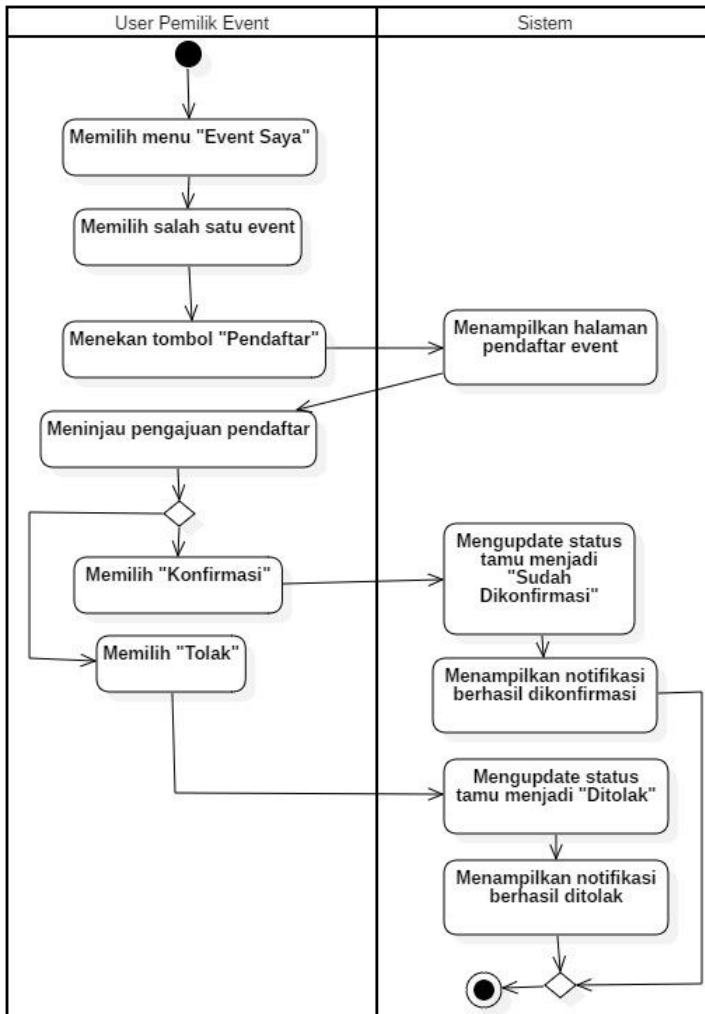
### 3.4.6 UC-06: Mengelola Pengajuan Pendaftar

Pada kasus penggunaan ini, Pemilik Event dapat melakukan pengelolaan terhadap data pengajuan pendaftar. Pengelolaan berupa menyetujui atau menolak data pengajuan pendaftar. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.7 dan diagram aktivitas pada Gambar 3.15

**Tabel 3.7 Spesifikasi Kasus Penggunaan Mengelola Pengajuan Pendaftar**

<b>Use Case</b>	<b>UC-06</b>		
<b>Nama Use Case</b>	Mengelola Pengajuan Pendaftar		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk menolak atau menyetujui tamu yang telah melakukan registrasi ke event.		
<b>Kondisi Awal</b>	Pengajuan pendaftar belum dikonfirmasi		
<b>Kondisi Akhir</b>	Pengajuan pendaftar diproses dan dikonfirmasi		
<b>Alur Normal</b>			
	Aktor		Sistem
1.	Aktor memilih menu “Event Saya”.		
2.	Memilih salah satu <i>event</i>		
3.	Memilih “Pendaftar”		
		4.	Menampilkan halaman pendaftar <i>event</i>
5.	Memilih salah satu data tamu		
6.	Memilih “Konfirmasi” A1. Memilih “Tolak”		
		7.	Mengupdate status tamu menjadi “Sudah Dikonfirmasi”
		8.	Menampilkan notifikasi data tamu berhasil dikonfirmasi
<b>Alur Alternatif</b>			

Aktor		Sistem	
		A1.1	Mengupdate status tamu menjadi "Ditolak"
		A1.2	Menampilkan notifikasi data tamu berhasil ditolak
Eksepsi		-	



**Gambar 3.15 Diagram Aktivitas Mengelola Pengajuan Pendaftar**



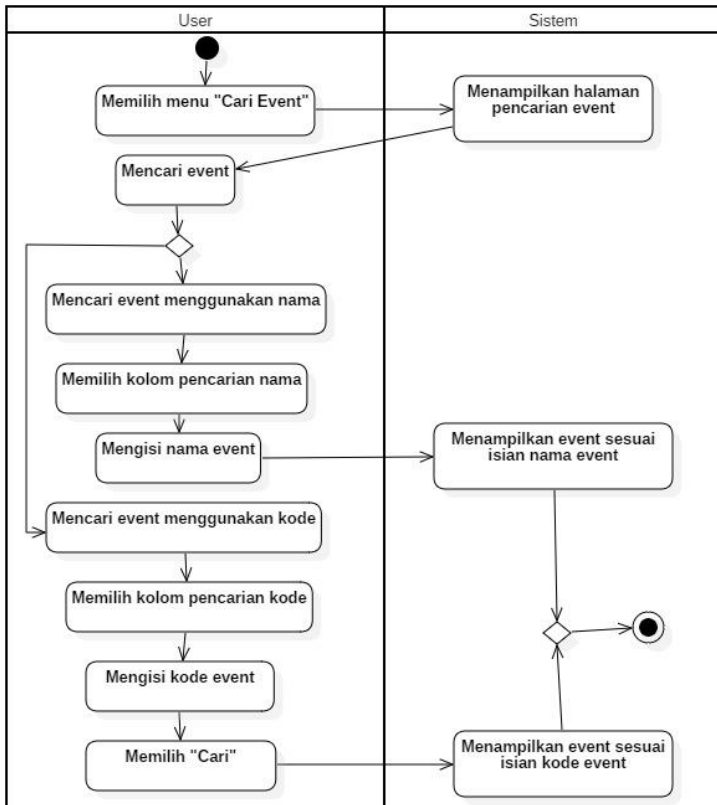
### 3.4.7 UC-07: Mencari Event

Pada kasus penggunaan ini, Tamu dapat melakukan pencarian terhadap *event*. Pengelolaan bisa dilakukan dengan menggunakan nama atau kode *event*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.8 dan diagram aktivitas pada Gambar 3.16

**Tabel 3.8 Spesifikasi Kasus Penggunaan Mencari Event**

<b>Use Case</b>	UC-07		
<b>Nama Use Case</b>	Mencari Event		
<b>Aktor</b>	Pemilik Event dan Tamu		
<b>Deskripsi</b>	Use case ini digunakan untuk mencari event dengan menggunakan nama atau kode event		
<b>Kondisi Awal</b>	Data mengenai event yang dicari belum tampil		
<b>Kondisi Akhir</b>	Data mengenai event yang dicari telah tampil		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Aktor memilih menu “Cari Event”.		
		2.	Menampilkan halaman pencarian <i>event</i> yang terdiri dari field: pencarian menggunakan nama, pencarian menggunakan kode dan daftar event yang ada di Kotakevent.
3.	Mencari <i>event</i> menggunakan nama A1. Mencari <i>event</i> menggunakan kode		

4.	Mengisikan nama <i>event</i> ke kolom pencarian menggunakan nama		
		4.	Menampilkan event sesuai dengan nama yang diisikan pada kolom pencarian.
<b>Alur Alternatif</b>			
Aktor		Sistem	
A1.1	Mengisikan kode <i>event</i> ke kolom pencarian menggunakan kode		
A1.2	Memilih “Cari”		
		A1.3.	Menampilkan event sesuai dengan kode yang diisikan pada kolom pencarian.
Eksepsi		-	



**Gambar 3.16 Diagram Aktivitas Mencari Event**

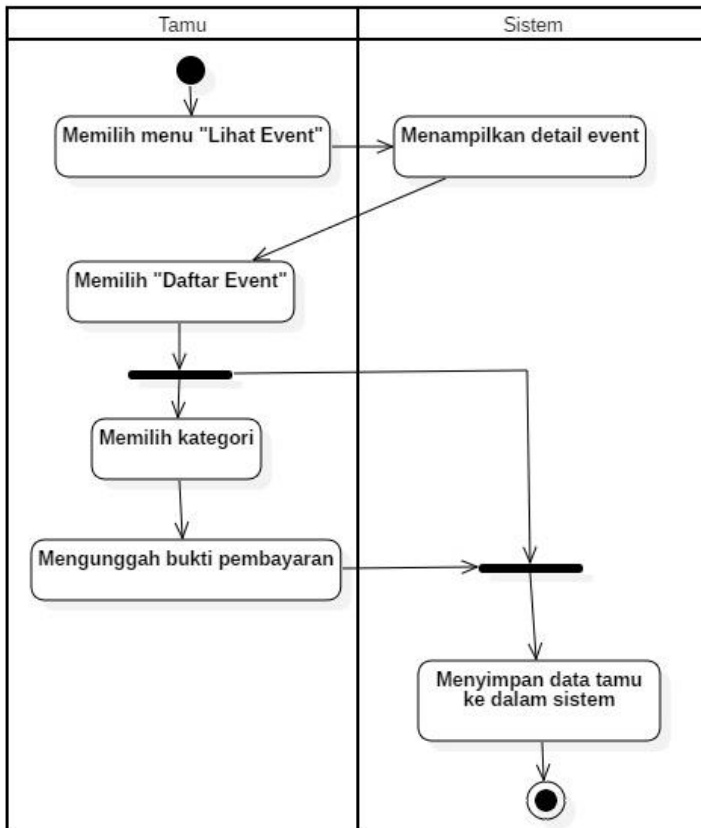
### 3.4.8 UC-08:Melakukan Registrasi Event

Pada kasus penggunaan ini, Tamu dapat melakukan ke *event* yang ada di Kotakevent. Pengelolaan berupa menambah kategori *event* dan mengubah kategori tamu. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.9 dan diagram aktivitas pada Gambar 3.17

**Tabel 3.9 Spesifikasi Kasus Penggunaan Melakukan Registrasi Event**

<b>Use Case</b>	<b>UC-08</b>		
<b>Nama Use Case</b>	Melakukan Registrasi <i>Event</i>		
<b>Aktor</b>	Tamu		
<b>Deskripsi</b>	Use case ini digunakan untuk melakukan registrasi atau mendaftar ke event yang diinginkan		
<b>Kondisi Awal</b>	Data tamu belum terdaftar ke dalam <i>event</i>		
<b>Kondisi Akhir</b>	Data tamu telah terdaftar ke dalam <i>event</i>		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Memilih menu “lihat <i>event</i> ”		
		2.	Menampilkan detail event yang terdiri dari field: foto, nama, tanggal,waktu, lokasi, galeri foto dan deskripsi event.
3.	Memilih “Daftar Event” A1. Event Berbayar		
		4.	Menyimpan data tamu ke dalam sistem
<b>Alur Alternatif</b>			
Aktor		Sistem	
A1.1	Memilih kategori		
A1.2	Mengunggah bukti pembayaran		

		A1.2	Menyimpan data tamu ke dalam sistem
Eksepsi			



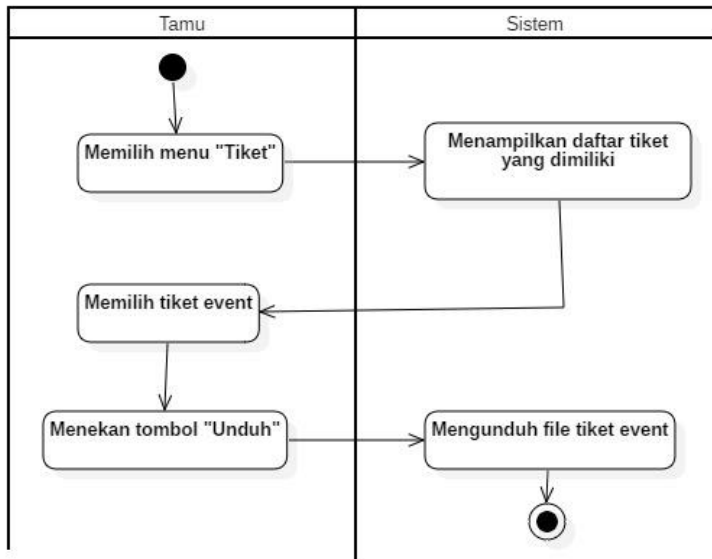
**Gambar 3.17** Diagram Aktivitas Melakukan Registrasi *Event*

### 3.4.9 UC-09: Mengunduh Tiket

Pada kasus penggunaan ini, Tamu dapat mengunduh tiket dari *event* yang telah didaftar. Tiket berbentuk file PDF disertai *qr code* yang berbeda di setiap tiket. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.10 dan diagram aktivitas pada Gambar 3.18

**Tabel 3.10 Spesifikasi Kasus Penggunaan Mengunduh Tiket**

<b>Use Case</b>	<b>UC-09</b>		
<b>Nama Use Case</b>	Mengunduh Tiket		
<b>Aktor</b>	Tamu		
<b>Deskripsi</b>	Use case ini digunakan untuk mengunduh tiket pada <i>event</i> yang telah didaftar.		
<b>Relasi</b>	-		
<b>Kondisi Awal</b>	File tiket <i>event</i> belum terunduh		
<b>Kondisi Akhir</b>	File tiket <i>event</i> telah terunduh		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Aktor memilih menu “Tiket”.		
		2.	Menampilkan semua tiket <i>event</i> yang pernah didaftar oleh aktor.
3.	Memilih <i>event</i> yang ingin diunduh tiketnya		
4.	Menekan tombol “Unduh”		
		5.	Mengunduh file tiket <i>event</i> .
Alur Alternatif		-	
Eksepsi		-	



**Gambar 3.18 Diagram Aktivitas Mengunduh Tiket**

### 3.4.10 UC-10: Mencatat Kehadiran Tamu

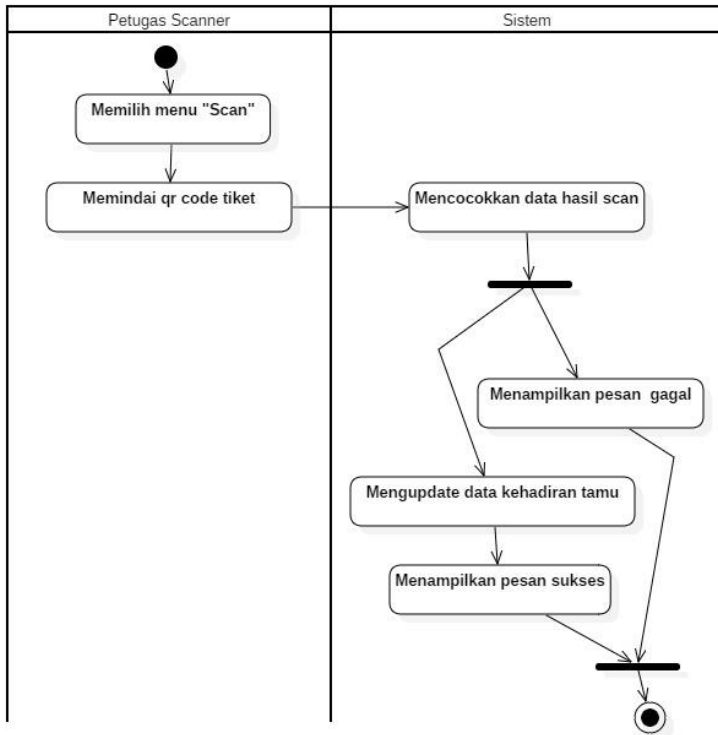
Pada kasus penggunaan ini, Petugas Scanner dapat mencatat kehadiran tamu dengan cara memindai *qr code* pada tiket. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.11 dan diagram aktivitas pada gambar 3.19

**Tabel 3.11 Spesifikasi Kasus Penggunaan Mencatat Kehadiran Tamu**

Use Case	UC-10
Nama Use Case	Mencatat Kehadiran Tamu
Aktor	Petugas Scanner

<b>Deskripsi</b>	Use case ini digunakan untuk mencatat data tamu yang telah hadir dalam acara		
<b>Relasi</b>	-		
<b>Kondisi Awal</b>	Data tamu belum tercatat hadir ke dalam <i>event</i>		
<b>Kondisi Akhir</b>	Data tamu telah tercatat hadir ke dalam <i>event</i>		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Aktor berada pada halaman “Mencatat Kehadiran Tamu”		
2.	Aktor menekan tombol “Scan”		
3.	Aktor memindai <i>qr code</i> pada tiket tamu menggunakan kamera <i>Smartphone Android</i> .		
		4.	Memeriksa hasil scan <i>qr code</i> tiket A1. Data tidak cocok
		5.	Mengupdate data kehadiran tamu ke dalam sistem
		6.	Menampilkan pesan sukses
<b>Alur Alternatif</b>			
Aktor		Sistem	
		A1.1	Sistem menampilkan pesan gagal bahwa data tidak ditemukan dalam sistem
Eksepsi		-	





**Gambar 3.19 Diagram Aktivitas Mencatat Kehadiran Tamu**

### 3.4.11 UC-11: Mengelola Pengajuan Event

Pada kasus penggunaan ini, Admin dapat melakukan pengelolaan terhadap data pengajuan pendaftar. Pengelolaan berupa menyetujui atau menolak data pengajuan pendaftar. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.12 dan diagram aktivitas pada Gambar 3.20

**Tabel 3.12 Spesifikasi Kasus Penggunaan Mengelola Pengajuan *Event***

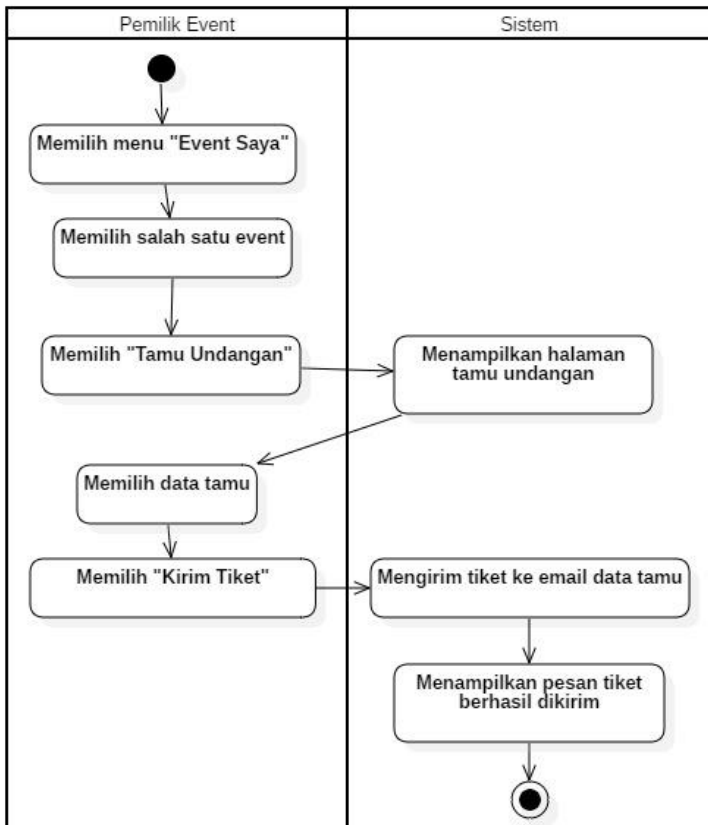
<b>Use Case</b>	<b>UC-11</b>		
<b>Nama Use Case</b>	Mengelola Pengajuan <i>Event</i>		
<b>Aktor</b>	Administrator		
<b>Deskripsi</b>	Use case ini digunakan untuk menolak atau menyetujui event yang telah dibuat oleh Pemilik <i>Event</i> .		
<b>Kondisi Awal</b>	Data mengenai data <i>event</i> masih sesuai dengan kondisi sebelum perubahan dilakukan.		
<b>Kondisi Akhir</b>	Data event yang diubah telah tersimpan ke dalam sistem		
<b>Alur Normal</b>			
Aktor		Sistem	
1.	Aktor memilih menu “Kelola Event”.		
		2.	Menampilkan halaman pengajuan <i>event</i>
3.	Meninjau pengajuan <i>event</i>		
4.	Memilih “Konfirmasi” A1. Memilh “Tolak”		
		5.	Mengupdate status <i>event</i> menjadi “Sudah Dikonfirmasi”
		6.	Menampilkan notifikasi pengajuan event berhasil dikonfirmasi
<b>Alur Alternatif</b>			
Aktor		Sistem	
		A1.1	Mengupdate status <i>event</i> menjadi “Ditolak”
		6.	Menampilkan notifikasi pengajuan event berhasil ditolak



pdf. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.13 dan diagram aktivitas pada Gambar 3.21

**Tabel 3.13 Spesifikasi Kasus Penggunaan Mengirim Tiket**

<b>Use Case</b>	<b>UC-12</b>		
<b>Nama Use Case</b>	Mengirim Tiket Tamu		
<b>Aktor</b>	Pemilik Event		
<b>Deskripsi</b>	Use case ini digunakan untuk mengirim tiket ke email tamu dalam bentuk file pdf		
<b>Relasi</b>	-		
<b>Kondisi Awal</b>	Tiket belum terkirim ke email tamu undangan		
<b>Kondisi Akhir</b>	Tiket telah terkirim ke email tamu undangan		
<b>Alur Normal</b>			
	Aktor		Sistem
1.	Memilih menu “Event Saya”.		
2.	Aktor memilih salah satu event		
3.	Aktor memilih “Tamu Undangan”		
		4.	Menampilkan halaman tamu undangan
5.	Aktor memilih data tamu		
	Aktor memilih “Kirim Tiket”		
		6.	Mengirim tiket ke email data tamu
		7.	Menampilkan pesan tiket berhasil dikirim
Alur Alternatif	-		
Eksepsi	-		

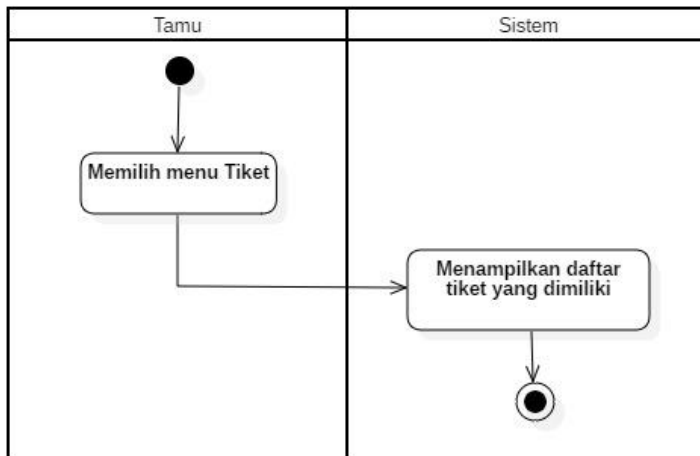


**Gambar 3.21 Diagram Aktivitas Mengirim Tiket**

### 3.4.13 UC-13:Melihat Daftar Tiket

**Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Daftar Tiket**

<b>Use Case</b>	UC-13		
<b>Nama Use Case</b>	Melihat Daftar Tiket		
<b>Aktor</b>	Tamu		
<b>Deskripsi</b>	Use case ini digunakan melihat daftar tiket yang dimiliki		
<b>Kondisi Awal</b>	-		
<b>Kondisi Akhir</b>	Daftar tiket yang dimiliki tampil		
<b>Alur Normal</b>			
	Aktor	Sistem	
1.	Aktor memilih menu "Tiket"		
		2.	Menampilkan daftar tiket yang dimiliki
<b>Alur Alternatif</b>	-		
<b>Eksepsi</b>	-		

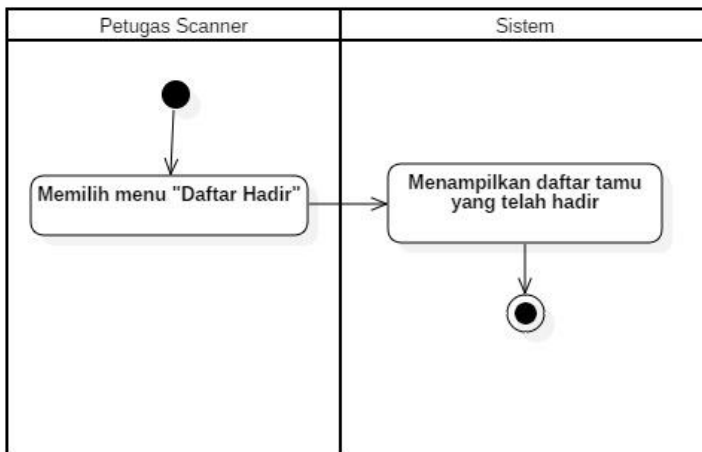


**Gambar 3.22 Diagram Aktivitas Melihat Daftar Tiket**

### 3.4.14 UC-14:Melihat Daftar Hadir

**Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat Daftar Hadir**

<b>Use Case</b>	UC-14												
<b>Nama Use Case</b>	Melihat Daftar Hadir												
<b>Aktor</b>	Petugas Scanner												
<b>Deskripsi</b>	Use case ini digunakan melihat daftar tamu yang telah hadir												
<b>Kondisi Awal</b>	-												
<b>Kondisi Akhir</b>	Daftar hadir tamu yang telah hadir tampil												
<b>Alur Normal</b>													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">Aktor</th> <th colspan="2" style="text-align: center;">Sistem</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.</td> <td>Aktor memilih menu "Daftar Hadir"</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">2.</td> <td>Menampilkan daftar tamu yang telah hadir</td> </tr> </tbody> </table>		Aktor		Sistem		1.	Aktor memilih menu "Daftar Hadir"					2.	Menampilkan daftar tamu yang telah hadir
Aktor		Sistem											
1.	Aktor memilih menu "Daftar Hadir"												
		2.	Menampilkan daftar tamu yang telah hadir										
<b>Alur Alternatif</b>	-												
<b>Eksepsi</b>	-												

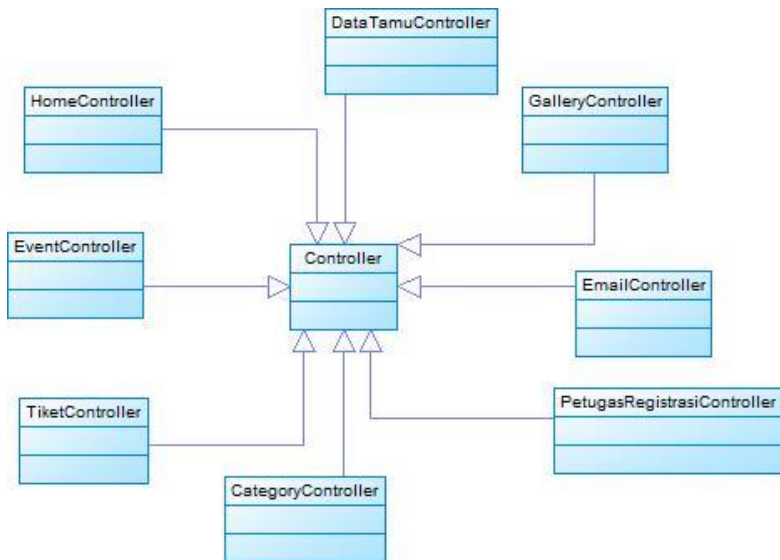


**Gambar 3.23 Diagram Aktivitas Melihat Daftar Hadir**

### 3.5 Perancangan

Pada subbab perancangan akan dijelaskan mengenai perancangan diagram kelas, perancangan basis data, dan perancangan antarmuka.

#### 3.5.1 Perancangan Diagram Kelas



**Gambar 3.24 Diagram Kelas Controller Kotakevent**

Gambar 3.24 menunjukkan model diagram kelas *controller* pada Kotakevent. Penggunaan arsitektur sistem seperti



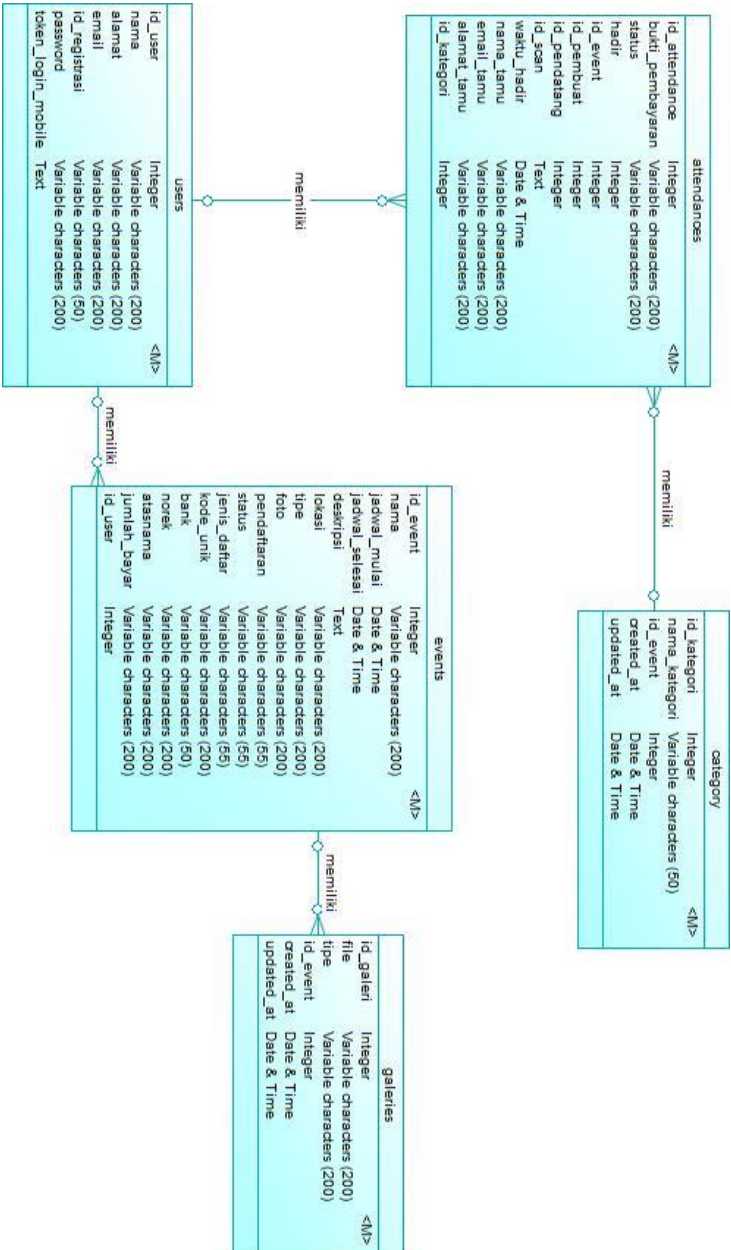
gambar di atas digunakan agar lebih mudah dalam melakukan *maintenance* dan lebih mudah dalam implementasi aplikasi.

### **3.5.2 Perancangan Basis Data**

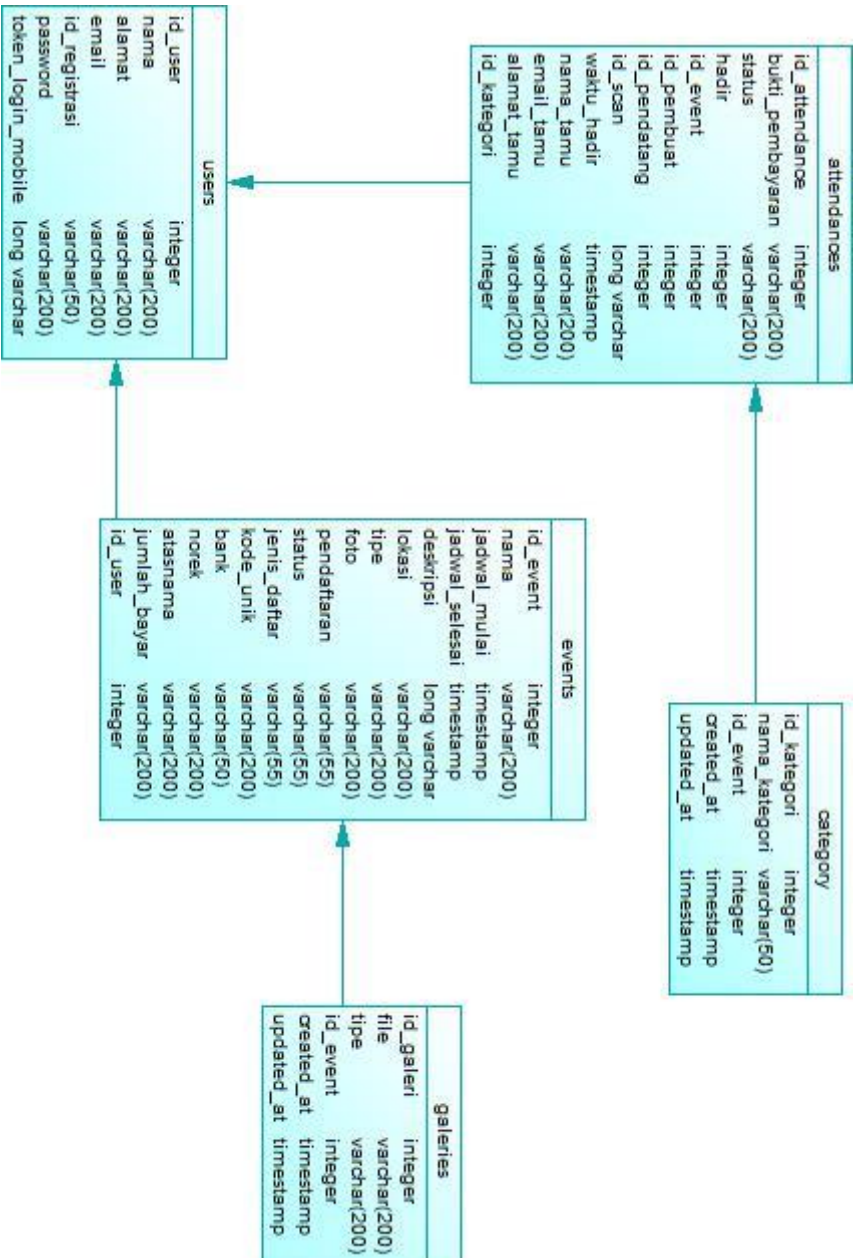
Perancangan Basis Data diperlukan untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung rancangan sistem yang akan dibuat.

Basis data yang akan digunakan adalah MySQL. MySQL dipilih menjadi basis data aplikasi ini karena memiliki sifat RDBMS yang berupa *open source*, mudah digunakan dan memiliki *performance turning*, yaitu menangani *query* sederhana dengan cepat.

Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) yang akan lebih dijelaskan pada Gambar 3.25 dan Gambar 3.26



Gambar 3.25 CDM Kotakevent



Gambar 3.26 PDM Kotakevent

### 3.5.2.1 Penjelasan Tabel

Penjelasan tabel untuk aplikasi Kotakevent adalah sebagai berikut :

1. Tabel Users

Tabel users digunakan untuk menyimpan data pengguna berupa nama, alamat, email, id\_registrasi, password, dan token. Tabel ini memiliki relasi dengan tabel lain yaitu tabel attendances.

2. Tabel Events

Tabel events digunakan untuk menyimpan data event berupa nama, jadwal event, deskripsi, lokasi, tipe, foto, jenis pendaftaran, status, kode unik, nama bank, no rekening, nama pemilik rekening, dan jumlah pembayaran. Tabel ini memiliki relasi dengan tabel lain yaitu tabel attendances, galleries, users, dan category.

3. Tabel Galleries

Tabel galleries digunakan sebagai tabel yang menyimpan data foto galeri event. Tabel ini memiliki relasi dengan tabel lain yaitu tabel Events.

4. Tabel Attendances

Tabel Attendances digunakan untuk menyimpan data peserta event berupa bukti pembayaran, status, id event, id pembuat, id pendaftar, id scan, waktu hadir, nama, email, alamat, dan id kategori. Tabel ini memiliki relasi dengan tabel lain yaitu tabel events dan users.

5. Tabel Category

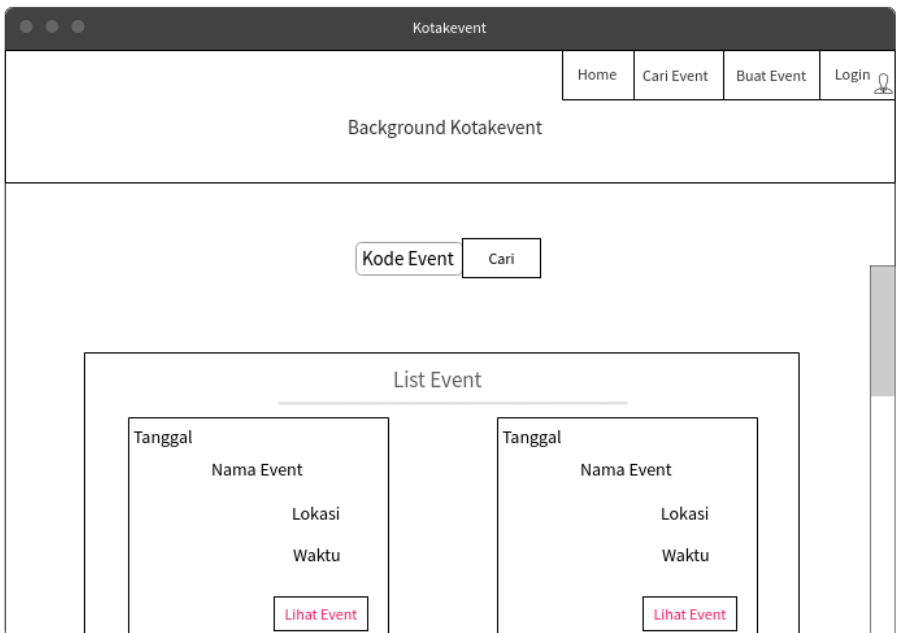
Tabel category digunakan sebagai tabel yang menyimpan data kategori event. Tabel ini memiliki relasi langsung dengan tabel lain yaitu tabel events.

### 3.5.3 Perancangan Antarmuka Web

Pada bagian ini akan dibahas mengenai rancangan antarmuka bagi pengguna untuk memenuhi kasus penggunaan yang sudah dirancang.

#### 3.5.3.1 Halaman Utama Kotakevent

Halaman mendaftar event akan digunakan pada kasus penggunaan melakukan registrasi. Rancangan dan penjelasan terkait antarmuka utama kotakevent dapat dilihat pada Gambar 3.27 dan Tabel 3.17



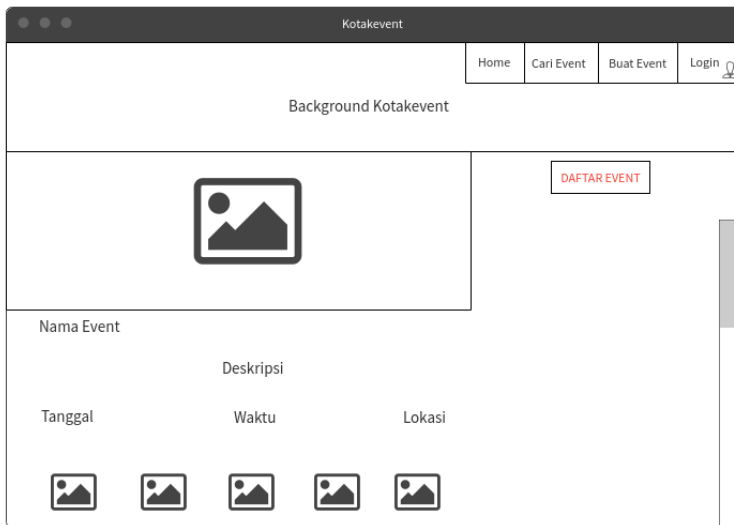
**Gambar 3.27 Rancangan Antarmuka Utama Kotakevent**

**Tabel 3.16 Penjelasan Antarmuka Utama Kotakevent**

<b>No</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Jenis Masukan/ Keluaran</b>
1	<i>loginButton</i>	<i>Button</i>	Tombol aksi untuk menampilkan form login dan register	<i>Button Clicked</i>
2	<i>lihatEvent</i>	<i>Button</i>	Tombol aksi untuk menampilkan detail event	<i>Button Clicked</i>
3	<i>cariEvent</i>	<i>Button</i>	Tombol aksi untuk mencari event menggunakan kode	<i>Button Clicked</i>
4	<i>kodeEvent</i>	<i>Input Text</i>	Field untuk mengisi kode event yang akan dicari	<i>String</i>

### 3.5.3.2 Halaman Detail Event

Halaman mendaftar event akan digunakan pada kasus penggunaan mendaftar event. Halaman ini berisi detail event yaitu nama event, jadwal, deskripsi, galeri foto dan tombol untuk mendaftar event. Pemilihan tombol daftar event akan langsung masuk ke database. Jika jenis pendaftaran event berbayar, maka akan muncul form pilih kategori dan setelah memilih kategori tamu harus mengunggah bukti pembayaran untuk menyelesaikan proses pendaftaran. Setelah data tamu terdaftar, maka tamu harus menunggu persetujuan pemilik *event* untuk mengkonfirmasi tamu yang mendaftar. Setelah dikonfirmasi pemilik *event* maka tamu dapat mengunduh tiket undangan. Rancangan dan penjelasan terkait antarmuka mendaftar event dapat dilihat pada Gambar 3.28 dan Tabel 3.17



**Gambar 3.28 Rancangan Antarmuka Detail Event**

**Tabel 3.17 Penjelasan Antarmuka Mendaftar Event**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1.	<i>daftarEvent</i>	<i>Button</i>	Tombol aksi untuk mendaftar event	<i>Button Clicked</i>
2.	<i>kategori</i>	<i>Button</i>	Tombol aksi untuk memilih kategori	<i>Button Clicked</i>

### 3.5.3.3 Halaman Membuat Event

Halaman membuat event akan digunakan pada kasus penggunaan membuat event. Halaman ini berisi form pengisian event dan tombol buat event. Pemilihan tombol buat event akan langsung masuk ke database dengan status belum dikonfirmasi, event dapat dikelola setelah dikonfirmasi oleh admin. Rancangan dan penjelasan terkait antarmuka membuat event dapat dilihat pada Gambar 3.29 dan Tabel 3.18.

Kotakevent

KOTAK EVENT

Home Cari Event Buat Event Event Saya Tiket User

Foto Event  No File Chosen

Nama Event Lokasi Event

Jadwal Mulai Jadwal Selesai

Jenis Pendaftaran  langsung terdaftar  memilih  berbayar

Pencarian Event  public  private

Kategori

Deskripsi Event

**Gambar 3.29 Rancangan Antarmuka Membuat Event**

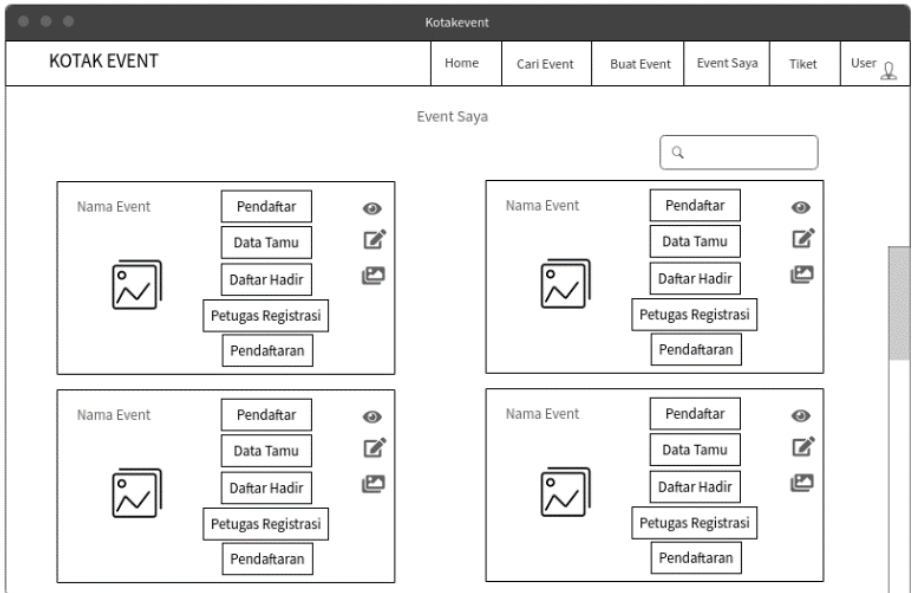


**Tabel 3.18 Penjelasan Antarmuka Membuat Event**

<b>No</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Jenis Masukan/Keluaran</b>
1	<i>fotoEvent</i>	<i>Input File</i>	Tombol aksi untuk mengunggah gambar	<i>File</i>
2	<i>inputPendaftaran</i>	<i>Input Radio</i>	Tombol aksi untuk memilih jenis pendaftaran	<i>String</i>
3	<i>inputPencarian</i>	<i>Input Radio</i>	Tombol aksi untuk memilih jenis pencarian	<i>String</i>
4	<i>tambahKategori</i>	<i>Input Text</i>	Tombol aksi untuk menambah kategori	<i>String</i>

### 3.5.3.4 Halaman Mengelola Event

Halaman mengelola event akan digunakan pada kasus penggunaan mengelola *event*, mengelola data tamu, mengelola pendaftar, melihat daftar *event*, melihat detail *event* dan melihat daftar hadir. Mengelola *event* berupa mengubah data *event* dan menambah foto galeri *event*, mengelola data tamu berupa menambah, mengubah, menghapus, mengekspor dan mengirim tiket data tamu, mengelola pendaftar berupa konfirmasi pendaftar, mengubah kategori tamu, dan mengekspor pdf data tamu, mengelola petugas scanner berupa menambah, mengubah dan menghapus data petugas scanner. Rancangan dan penjelasan terkait antarmuka mengelola event dapat dilihat pada Gambar 3.30 dan Tabel 3.19.



**Gambar 3.30 Rancangan Antarmuka Mengelola Event**

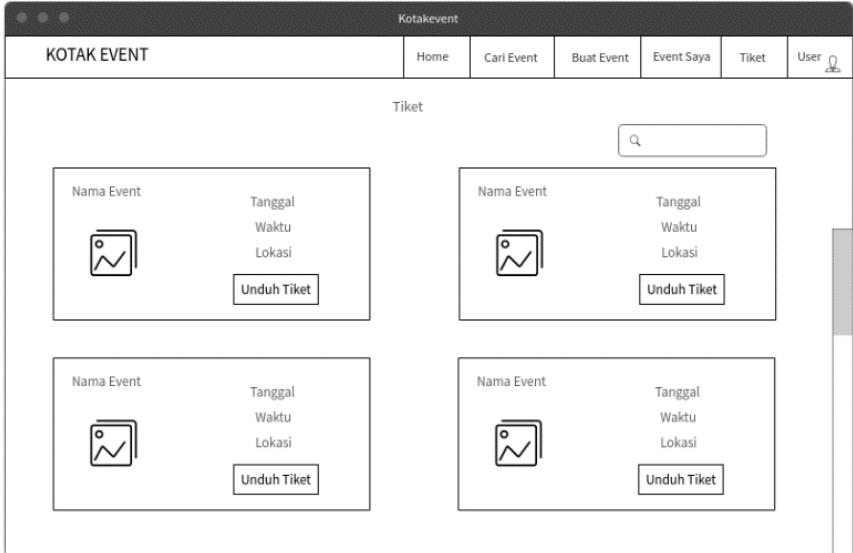
**Tabel 3.19 Penjelasan Antarmuka Mengelola Event**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>pendaftarEvent</i>	<i>Button</i>	Tombol aksi untuk mengelola pendaftar	<i>Button Clicked</i>
2	<i>tamuEvent</i>	<i>Button</i>	Tombol aksi untuk mengelola data tamu	<i>Button Clicked</i>
3	<i>hadirEvent</i>	<i>Button</i>	Tombol aksi untuk mengelola daftar hadir	<i>Button Clicked</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
4	<i>petugasEvent</i>	<i>Button</i>	Tombol aksi untuk mengelola petugas registrasi	<i>Button Clicked</i>
5	<i>pendaftaranEvent</i>	<i>Button</i>	Tombol aksi untuk mengelola pendaftaran	<i>Button Clicked</i>

### 3.5.3.5 Halaman Mengelola Tiket

Halaman mengelola tiket akan digunakan pada kasus penggunaan mengelola tiket dan melihat daftar tiket. Halaman ini berisi daftar tiket yang dimiliki, informasi tiket dan tombol unduh tiket. Rancangan dan penjelasan terkait antarmuka mengelola tiket dapat dilihat pada Gambar 3.31 dan Tabel 3.21



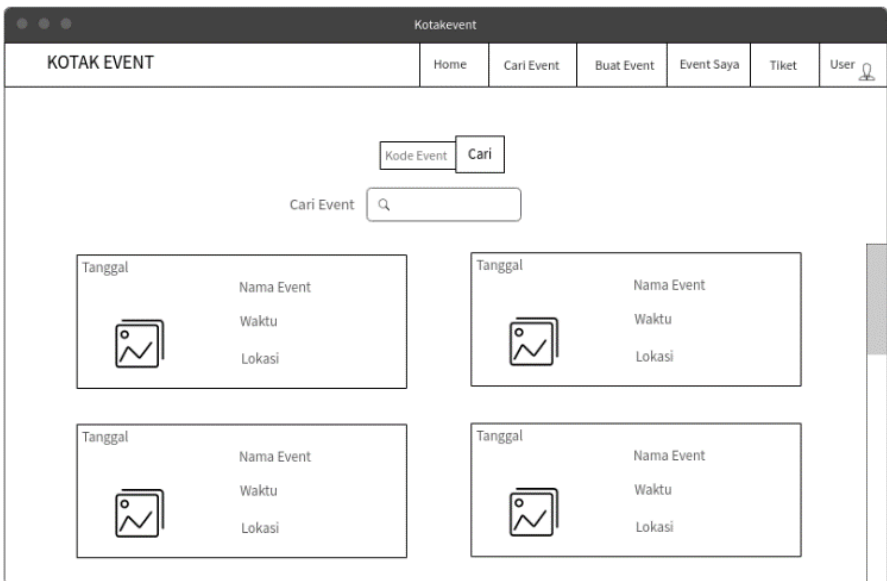
**Gambar 3.31 Rancangan Antarmuka Mengelola Tiket**

**Tabel 3.20 Penjelasan Antarmuka Mengelola Tiket**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>unduhTiket</i>	<i>Button</i>	Tombol aksi untuk mengunduh tiket	<i>Button Clicked</i>
2	<i>uploadBukti</i>	<i>Button</i>	Tombol aksi untuk mengupload bukti pembayaran	<i>Button Clicked</i>

### 3.5.3.6 Halaman Mencari Event

Halaman mencari event akan digunakan pada kasus penggunaan mencari event. Halaman ini berisi semua daftar event yang memiliki status sudah dikonfirmasi, kolom pencarian dengan nama dan kolom pencarian dengan kode. Rancangan dan penjelasan terkait antarmuka mencari event dapat dilihat pada Gambar 3.32 dan Tabel 3.21.

**Gambar 3.32 Rancangan Antarmuka Mencari Event**

**Tabel 3.21 Penjelasan Antarmuka Mencari Event**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>kodeEvent</i>	<i>Input Text</i>	Kolom input untuk mengisi kode event	<i>String</i>
2	<i>cariButton</i>	<i>Button</i>	Tombol aksi untuk mencari event	<i>Button Clicked</i>

### 3.5.3.7 Halaman Mengelola Pengajuan Event

Halaman mengelola pengajuan event akan digunakan pada kasus penggunaan mengelola pengajuan event, melihat data pengajuan event, melihat semua daftar event, dan melihat event selesai. Halaman ini berisi tiga sub tabel yang meliputi tabel pengajuan event, tabel status event, dan tabel event selesai. Rancangan dan penjelasan terkait antarmuka pengajuan event dapat dilihat pada Gambar 3.33.

KELOLA EVENT

Pengajuan Event		Event Dibuat	Event Selesai		
No	Event	Pembuat	Jadwal	Detail	Action
1.	Event1	Garry	1 Mei 2019	Lihat	<input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px;" type="button" value="Konfirmasi"/> <input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px; margin-left: 10px;" type="button" value="Tolak"/>
2.	Event2	Nabila	2 Mei 2019	Lihat	<input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px;" type="button" value="Konfirmasi"/> <input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px; margin-left: 10px;" type="button" value="Tolak"/>
3.	Event3	Anindha	3 Mei 2019	Lihat	<input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px;" type="button" value="Konfirmasi"/> <input style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 10px; margin-left: 10px;" type="button" value="Tolak"/>

**Gambar 3.33 Rancangan Antarmuka Mengelola Pengajuan Event**

### 3.5.4 Perancangan Antarmuka Mobile

#### 3.5.4.1 Halaman Registrasi Event

Halaman registrasi event akan digunakan pada kasus penggunaan menscan qr code tiket dan menambah data tamu yang telah hadir. Halaman ini berisi data daftar hadir, pencarian peserta dan tombol scan event. Rancangan dan penjelasan terkait antarmuka registrasi event dapat dilihat pada Gambar 3.34 dan Tabel 3.22.



**Gambar 3.34 Rancangan Antarmuka Registrasi Event**

**Tabel 3.22 Penjelasan Antarmuka Registrasi Event**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>scanData</i>	<i>Button</i>	Tombol aksi untuk menscan tiket	<i>Button Clicked</i>
2	<i>tambahData</i>	<i>Button</i>	Tombol aksi untuk menambah data tamu	<i>Button Clicked</i>
3	<i>daftarHadir</i>	<i>Button</i>	Tombol aksi untuk melihat daftar hadir	<i>Button Clicked</i>

### 3.5.4.2 Halaman Melihat Daftar Hadir

Halaman melihat daftar hadir akan digunakan pada kasus penggunaan melihat daftar hadir. Halaman ini berisi nama dan kategori tamu yang telah teregistrasi. Rancangan dan penjelasan terkait antarmuka melihat daftar hadir dapat dilihat pada Gambar 3.35



The image shows a mobile application interface for viewing event attendees. It features a green header with the text 'KOTAK EVENT'. Below the header is a white navigation bar containing a blue back arrow, the text 'Kotak Event', and a blue magnifying glass icon. The main content area is a table with five rows, each representing an attendee. The first two rows have 'VIP' as their category, while the last three rows have 'Umum' (General) as their category.

KOTAK EVENT	
← Kotak Event	🔍
Gerry	VIP
Nabila	VIP
Adib	Umum
Anindha	Umum
Sabhira	Umum

**Gambar 3.35 Rancangan Antarmuka Melihat Daftar Hadir**

### 3.5.5 Perancangan API Endpoint Petugas Scanner

Pada Perancangan API Endpoint untuk **Aplikasi Registrasi Tamu Pada Perangkat Bergerak Berbasis Android** bermula pada pembuatan *web* menggunakan *framework laravel* yang sudah tersedia library API REST pada *controller*. Perancangan API *Endpoint* Petugas Registrasi untuk merelasikan antara *web* dengan *android*, dijelaskan pada Tabel 3.23

**Tabel 3.23 Perancangan API Endpoint Petugas Scanner**

No	Endpoint URL	Method	Deskripsi
1	<i>/api_getListEvent</i>	<i>GET</i>	Mendapatkan data tamu yang telah melakukan registrasi
2	<i>/api_scan</i>	<i>POST</i>	Mendapatkan konten qr code dan meregistrasi tamu
3	<i>/api_search</i>	<i>POST</i>	Mencari data tamu yang telah melakukan registrasi



## BAB IV IMPLEMENTASI

Bab ini membahas mengenai implementasi sistem sesuai dengan analisis dan perancangan proses bisnis pada aplikasi Kotakevent yang telah dijelaskan pada bab sebelumnya.

### 4.1 Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1

**Table 4.1 Lingkungan Implementasi**

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-5200U CPU @ 2.20GHz (4 CPUs) , ~2.2GHz Memori: 8192 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows 10 Home 64-bit Perangkat Pengembang: Laravel, Android Studio, Postman Perangkat Perancang Diagram: Sybase Power Designer 16, StarUML Perangkat Database: MySQL

### 4.2 Implementasi Antarmuka Sistem

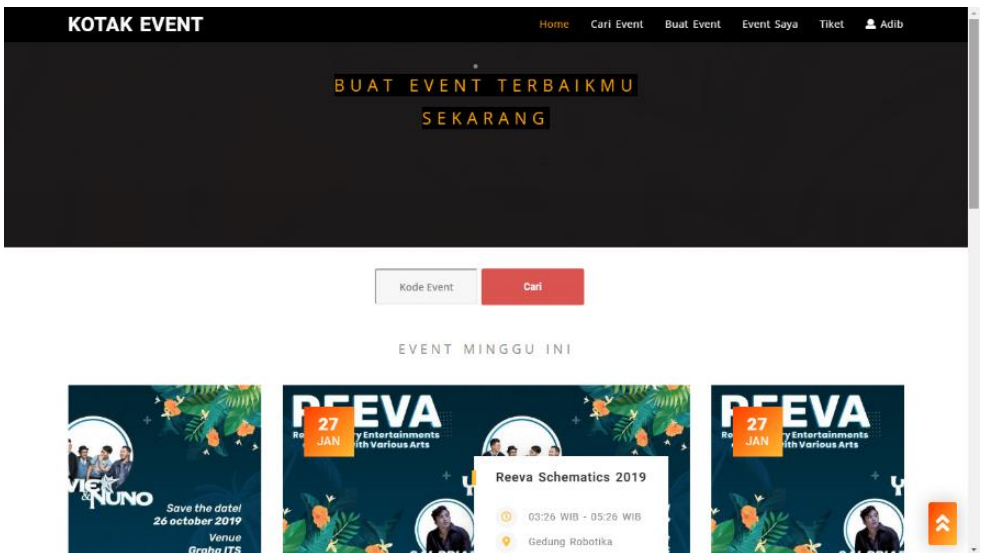
Implementasi antarmuka sistem dilakukan dengan menggunakan *file* dengan ekstensi *.blade.php* untuk setiap halaman web dan menggunakan file *activity* pada halaman *mobile*. Berikut ini akan dijelaskan mengenai implementasi antarmuka sistem yang sudah direalisasikan.

### 4.2.1 Antarmuka Web

Implementasi antarmuka system dengan platform web. Antarmuka web terdiri dari halaman utama, halaman mendaftar event, halaman membuat event, halaman mengelola event, halaman mengelola tiket, halaman mencari event, dan halaman mengelola pengajuan event.

### 4.2.2 Halaman Utama Kotakevent

Halaman ini merupakan halaman utama Kotakevent. Di dalam halaman ini berisi list event dan pencarian event menggunakan kode. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.1



Gambar 4.1 Halaman Utama Kotakevent

### 4.2.3 Halaman Registrasi *Event*

Halaman ini merupakan halaman registrasi *event*. Di dalam halaman ini berisi detail event berupa nama event, jadwal event, lokasi event, deskripsi event, foto galeri event dan tombol daftar event. Jika pengguna mengklik tombol daftar event, maka data pengguna akan masuk ke dalam database dengan status sudah dikonfirmasi, belum dikonfirmasi atau menunggu pembayaran sesuai dengan jenis pendaftaran event. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.2

The screenshot shows the 'KOTAK EVENT' website interface. At the top, there is a navigation bar with links for Home, Cari Event, Buat Event, Event Saya, Tiket, and a user profile icon labeled 'Bisma'. The main content area features a large banner for 'REEVA Revolutionary Entertainments and Expo with Various Arts'. The banner includes the names 'YOVIE & NUNO' and 'SAL PRIADI', the date '26 October 2019', and the venue 'Graha ITS'. Below the banner, the event title 'Reeva Schematics 2019' is displayed, followed by an orange 'DAFTAR EVENT' button. A 'Deskripsi' section contains three event details cards: 'Tanggal Event 27 JAN 2020', 'Waktu Event 03:26 WIB - 05:26 WIB', and 'Lokasi Event GEDUNG ROBOTIKA'. To the right of the banner, there is a payment information section showing the account 'BRI 5115100036 Adeb Kuncoro' and ticket prices: 'Platinum Rp 75.000 | Gold Rp 50.000 | Silver Rp 25.000'. A vertical scrollbar is visible on the right side of the page.

Gambar 4.2 Halaman Mendaftar Event

#### 4.2.4 Halaman Membuat Event

Halaman ini merupakan halaman membuat Event. Di dalam halaman ini terdapat form event yang berisi foto, nama, lokasi, tanggal mulai & selesai, jenis pendaftaran, pencarian event, kategori dan deskripsi.. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.3

The screenshot displays the 'KOTAK EVENTS' web application interface for creating an event. The navigation bar at the top includes 'Home', 'Cari Event', 'Buat Event', 'Event', 'Undangan', and a user profile 'Adib Wahyu Kuncoro'. The main form contains the following elements:

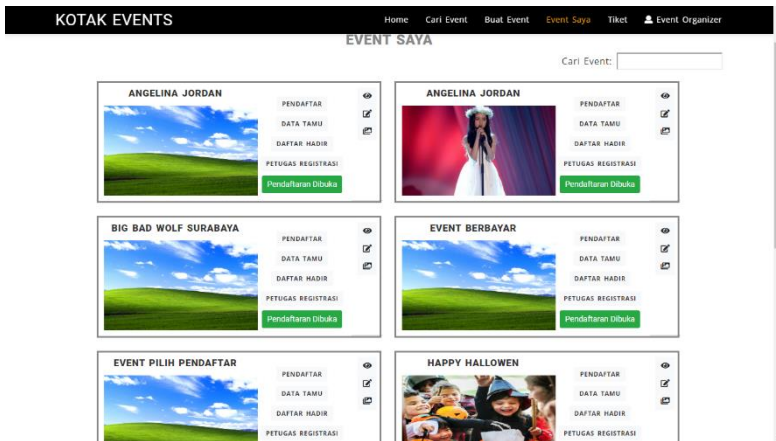
- Photo Event:** A file upload field with a 'Choose File' button and the text 'No file chosen'.
- Form Fields:** Input boxes for 'Nama Event', 'Lokasi Event', 'Jadwal Mulai', and 'Jadwal Selesai'. A tooltip 'Please fill out this field.' is visible over the 'Jadwal Selesai' field.
- Jenis Pendaftaran:** Three radio button options:
  - siapapun yang mendaftar akan langsung terverifikasi dan mendapatkan QRCode undangan
  - saya akan memilih pendftar yang dapat mengikuti event ini
  - event ini berbayar dan pendftar harus mengunggah bukti pembayaran
- Pencarian Event:** Two radio button options:
  - Siapapun dapat mencari acara ini
  - Acara ini hanya dapat dicari menggunakan kode
- Kategori:** An input field with a green 'Tambah' button.
- Deskripsi:** A large text area with the placeholder 'Masukkan deskripsi acara'.

An orange arrow icon is located in the bottom right corner of the page.

**Gambar 4.3 Halaman Membuat Event**

## 4.2.5 Halaman Mengelola Event

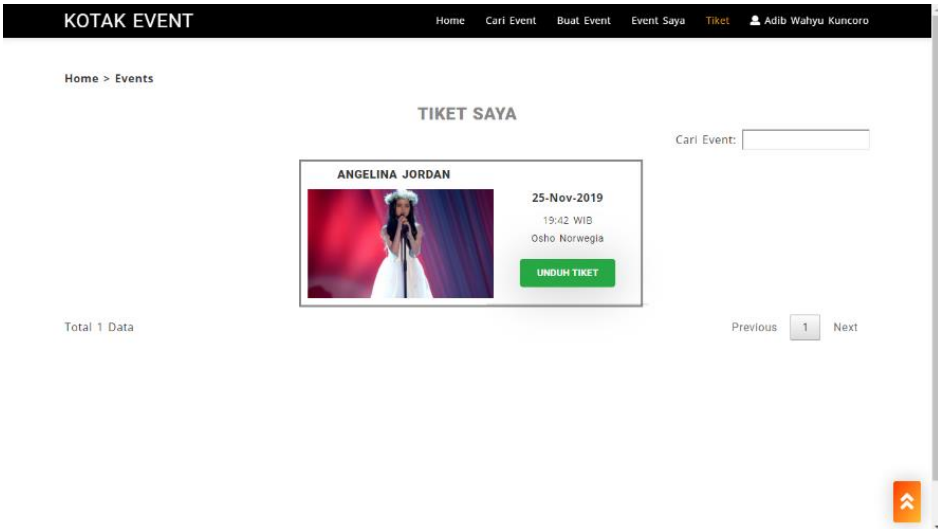
Halaman ini merupakan halaman mengelola event, mengelola data tamu, mengelola pendaftar, melihat daftar event, melihat detail event dan melihat daftar hadir. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.4



**Gambar 4.4** Halaman Mengelola Event

## 4.2.6 Halaman Mengelola Tiket

Halaman ini merupakan halaman mengelola tiket. Halaman ini berisi daftar tiket yang dimiliki, informasi tiket dan tombol unduh tiket. Pemilihan tombol unduh tiket akan langsung mengunduh tiket dalam bentuk pdf. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.5.



**Gambar 4.5 Halaman Mengelola Tiket**

#### **4.2.7 Halaman Mengelola Pengajuan Event**

Halaman ini merupakan mengelola pengajuan event, melihat data pengajuan event, melihat semua daftar event, dan melihat event selesai. Halaman ini berisi tiga sub tabel yang meliputi tabel pengajuan event, tabel status event, dan tabel event selesai. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.6

**KOTAK EVENTS** Home Kelola Event Admin

### KELOLA EVENT

Pengajuan Event **Status Event** Event Selesai

Show  entries Search:

No	Nama Event	Pembuat	Jadwal	Jenis Event	Jenis Pendaftaran	Detail Event	Action
No data available in table							

Showing 0 to 0 of 0 entries Previous Next

**KOTAK EVENT** Home Kelola Event Admin

### KELOLA EVENT

Pengajuan Event **Status Event** Event Selesai

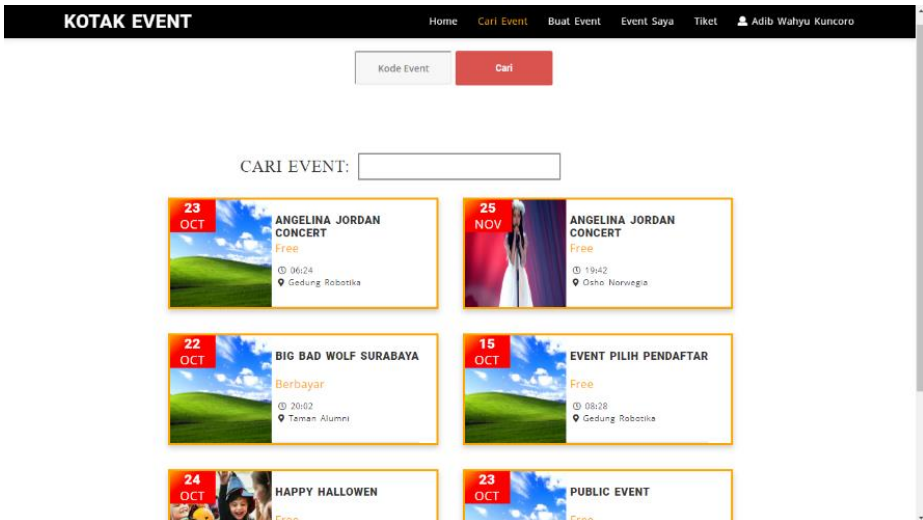
Show  entries Search:

No	Nama Event	Pembuat	Detail	Status
1	Surabaya Maraton 140K 2020	Adib		SUDAH DIKONFIRMASI
2	Public Event	Adib		SUDAH DIKONFIRMASI
3	Event Pilih Pendaftar	Adib		SUDAH DIKONFIRMASI
4	Event Berbayar	Adib		SUDAH DIKONFIRMASI
5	Big Bad Wolf Surabaya	Adib		SUDAH DIKONFIRMASI
6	Happy Halloween	Adib		SUDAH DIKONFIRMASI
7	Angelina Jordan Concert	Adib		SUDAH DIKONFIRMASI
8	Opera Kecoa	Adib		SUDAH DIKONFIRMASI
9	Sampek Engtay	Adib		SUDAH DIKONFIRMASI
10	Wedding Resepsi	Adib		SUDAH DIKONFIRMASI

**Gambar 4.6 Halaman Mengelola Pengajuan Event**

## 4.2.8 Halaman Mencari Event

Halaman ini merupakan halaman mencari event. Halaman ini berisi semua daftar event yang memiliki status sudah dikonfirmasi, kolom pencarian dengan nama dan kolom pencarian dengan kode. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.7



Gambar 4.7 Halaman Mencari Event

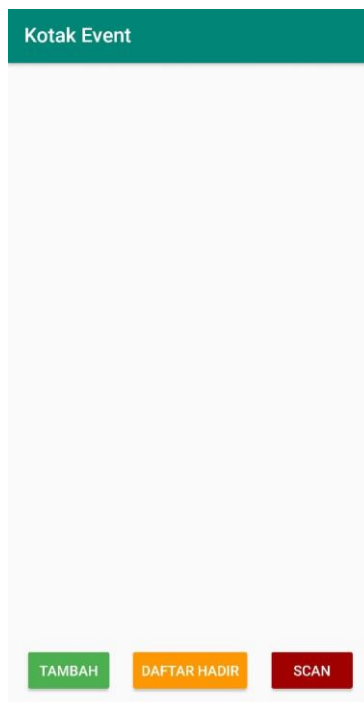


#### 4.2.9 Antarmuka Mobile

Implementasi antarmuka system dengan platform mobile. Antarmuka mobile terdiri dari halaman registrasi event, halaman menambah data tamu teregistrasi dan halaman melihat daftar hadir.

#### 4.2.10 Halaman Mencatat Kehadiran

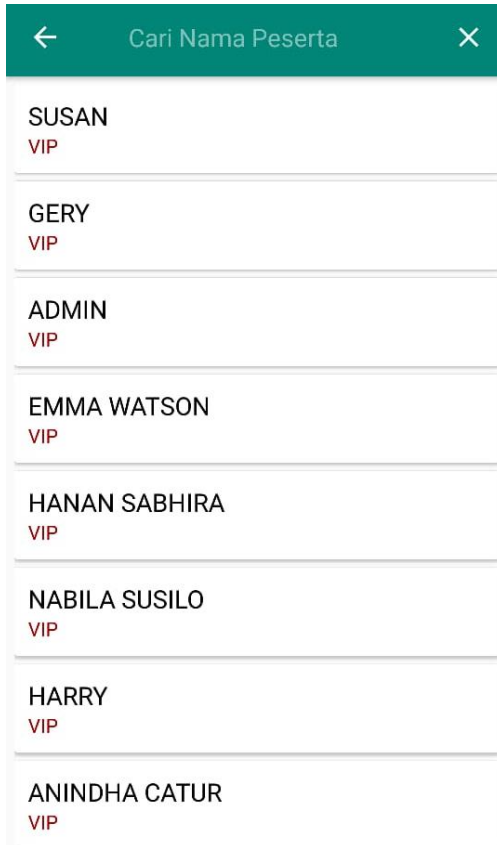
Halaman ini merupakan halaman menscan qr code tiket event dan menambah data tamu yang telah hadir. Halaman ini berisi data daftar hadir, pencarian peserta dan tombol scan event. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.8



**Gambar 4.8 Halaman Registrasi Event**

#### 4.2.11 Halaman Melihat Daftar Hadir

Halaman ini merupakan halaman melihat daftar hadir. Halaman ini berisi data daftar hadir, pencarian peserta dan tombol scan event. Realisasi dari halaman ini dapat ditunjukkan pada Gambar 4.9



**Gambar 4.9** Halaman Melihat Daftar Hadir

#### 4.2.12 Implementasi API Endpoint Petugas Scanner

Implementasi API *endpoint* terhadap sistem aplikasi yang telah dibuat. Implementasi ini digunakan untuk mendapatkan data tamu yang telah hadir, mencari data tamu yang telah hadir dan mencatat kehadiran tamu.

```
1. @FormUrlEncoded
2. @POST("insert")
3. Call<Value> daftar(
4.     @Field("nama") String nama,
5.     @Field("email") String email,
6.     @Field("alamat") String alamat,
7.     @Field("token") String token);
8.
9. @FormUrlEncoded
10. @POST("scan")
11. Call<Value> scan(
12.     @Field("id_scan") String id_scan,
13.     @Field("token") String token);
14.
15. @FormUrlEncoded
16. @POST("getListEvent")
17. Call<Value> view(
18.     @Field("token") String token);
19.
20. @FormUrlEncoded
21. @POST("search")
22. Call<Value> search(
23.     @Field("token") String token);
24.     @Field("search") String search);
25.
```

**Kode Sumber 4.1 Fungsi pada service Petugas Scanner untuk Memanggil Data**

**Tabel 4.1 Penjelasan Implementasi API Endpoint Petugas Registrasi**

No	Endpoint URI	Method	Deskripsi
1	<i>/scan</i>	<i>POST</i>	Mendapatkan konten qr code dan meregistrasi tamu
	Post Parameter: <i>JSON: id_scan, token</i>		
2	<i>/getListEvent</i>	<i>POST</i>	Mendapatkan data tamu yang telah melakukan registrasi
	Post Parameter: <i>JSON: token</i>		
3	<i>/search</i>	<i>POST</i>	Mencari data tamu yang telah melakukan registrasi
	Post Parameter: <i>JSON: token, search</i>		

## **BAB V**

### **UJI COBA DAN EVALUASI**

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi aplikasi Kotakevent.

#### **5.1 Lingkungan Uji Coba**

Lingkungan uji coba adalah kombinasi antara perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba. Pengujian dilakukan dengan menggunakan tiga buah lingkungan pengujian yaitu lingkungan pengujian aplikasi, *server* basis data, dan *server* aplikasi *web*. Adapun rincian dari masing-masing lingkungan pengujian tersebut secara berturut-turut ditunjukkan pada Tabel 5.1, Tabel 5.2.

**Tabel 5.1 Lingkungan Pengujian Aplikasi Web**

<b>Spesifikasi</b>	<b>Deskripsi</b>
CPU	Intel® Core™ i5-5200U CPU @ 2.20GHz
RAM	8.00 GB
Sistem Operasi	Windows 10 Home 64-bit
Browser	Chrome versi 78
Server Web	Apache 2.4.29

**Tabel 5.2 Lingkungan Pengujian Aplikasi Mobile**

<b>Spesifikasi</b>	<b>Deskripsi</b>
Jenis Perangkat	<i>Smartphone</i>
Merk Perangkat	Xiaomi Redmi Note 5
Sistem Operasi	Android 8.0 Oreo
Memori internal	32 GB
RAM	2GB

## 5.2 Skenario Pengujian

Pada bagian ini akan dibahas mengenai proses uji coba yang digunakan. Pengujian dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang sudah dirancang pada sistem. Metode *black box* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya.

Selain pengujian dengan metode *black box*, juga dilakukan pengujian dengan aktor yang berperan langsung di aplikasi untuk menilai fungsionalitas aplikasi

### 5.2.1 Pengujian Terhadap Fungsionalitas Sistem

Pada proses uji coba, pengujian dilakukan dengan menjalankan serangkaian perintah terhadap beberapa skenario berdasarkan tipe *event* yang selanjutnya akan disebut sebagai kasus pengujian. Kasus pengujian ini berkorelasi dengan kasus-kasus penggunaan dan kebutuhan fungsional yang sebelumnya sudah dirancang dan dijelaskan pada Bab III.

#### 5.2.1.1 Kasus Pengujian Tipe Event Open Berbayar Umum

Pada kasus uji ini, pengujian yang dilakukan yaitu membuat, mencari, dan mendaftar event dengan tipe open berbayar umum.

**Tabel 5.3 Kasus Membuat Event Open Berbayar Umum**

<b>ID</b>	UJ-001
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event Open Berbayar Umum
<b>Nama</b>	Pengujian Membuat Event Open Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu membuat event tipe open berbayar umum

<b>ID</b>	UJ-001
<b>Kondisi Awal</b>	Berada di halaman Buat Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Buat Event”</li> <li>2. Sistem akan menampilkan halaman buat event yang berisi <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, jadwal mulai, jadwal selesai, jenis <i>event</i>, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i></li> <li>3. Pemilik <i>event</i> mengisi <i>field</i> yang tersedia dengan memilih jenis <i>event</i> : open, jenis pendaftaran :berbayar, jenis pencarian : umum.</li> <li>4. Pemilik event menekan tombol “Buat Event”</li> <li>5. Sistem menyimpan data event ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat membuat <i>event</i> dengan tipe open berbayar umum
<b>Hasil yang diperoleh</b>	<i>Event</i> dengan tipe open berbayar umum berhasil dibuat
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.4 Kasus Mencari Event Open Berbayar Umum**

<b>ID</b>	UJ-002
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mencari Event Open Berbayar Umum
<b>Nama</b>	Pengujian Mencari Event Open Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mencari event tipe open berbayar umum
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>:</li> </ol>

<b>ID</b>	UJ-002
	<p>foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</p> <ol style="list-style-type: none"> <li>3. Tamu menekan kolom pencarian menggunakan nama <i>event</i></li> <li>4. Tamu mengisikan nama <i>event</i> yang ingin dicari</li> <li>5. Sistem menampilkan <i>event</i> sesuai dengan inputan nama <i>event</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan daftar <i>event</i> sesuai dengan nama <i>event</i> yang dimasukkan
<b>Hasil yang diperoleh</b>	Sistem telah menampilkan daftar <i>event</i> sesuai dengan nama <i>event</i> yang dimasukkan
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.5 Kasus Mendaftar Event Open Berbayar Umum**

<b>ID</b>	UJ-003
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mendaftar Event Open Berbayar Umum
<b>Nama</b>	Pengujian Mendaftar Event Open Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah tamu dapat mendaftar ke event tipe open berbayar umum
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu mencari dan memilih event menggunakan nama event</li> <li>4. Sistem menampilkan detail <i>event</i></li> <li>5. Tamu menekan tombol “Daftar Event”</li> <li>6. Tamu memilih kategori <i>event</i></li> <li>7. Tamu menekan tombol “Daftar”</li> </ol>



<b>ID</b>	UJ-003
	8. Sistem menyimpan data tamu ke dalam basis data 9. Tamu menekan tombol “Upload Bukti Pembayaran” 10. Tamu memilih file bukti pembayaran untuk diunggah 11. Tamu menekan tombol simpan 12. Sistem menyimpan data tamu ke dalam basis data
<b>Hasil yang diharapkan</b>	Tamu dapat mendaftar ke <i>event</i> dengan tipe open berbayar umum
<b>Hasil yang diperoleh</b>	Tamu telah mendaftar ke <i>event</i> dengan tipe open berbayar umum
<b>Hasil Pengujian</b>	Berhasil

Jenis Event



Open (tamu dapat registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event)



Close (tamu hanya dapat ditambahkan oleh pemilik event)

Jenis Pendaftaran



siapapun yang mendaftar akan langsung terverifikasi dan mendapatkan Tiket



saya akan memilih pendaftar yang dapat mengikuti event ini (tiket didapat setelah disetujui)



event ini berbayar dan pendaftar harus mengunggah bukti pembayaran untuk mendapatkan tiket

Bank

No Rekening

Atas Nama

Harga Tiket

Pencarian Event

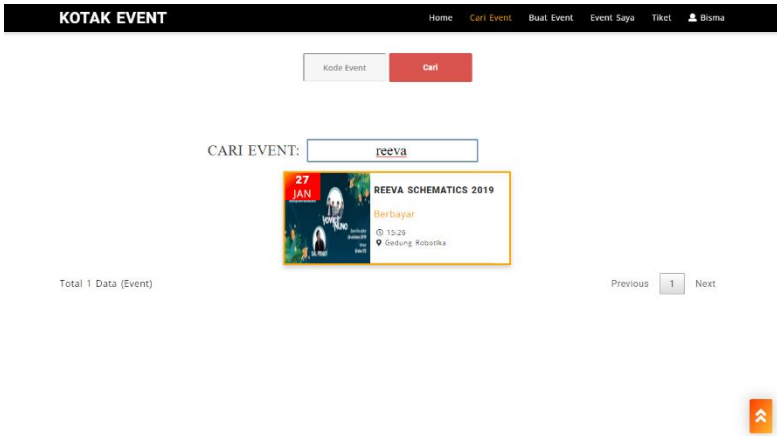


Siapapun dapat mencari acara ini

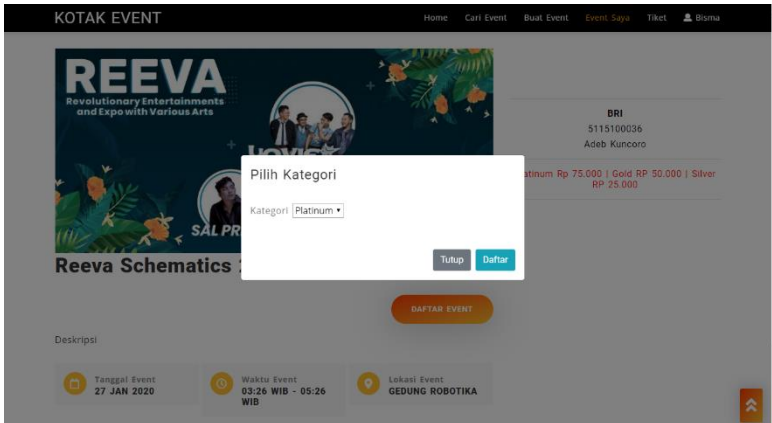


Acara ini hanya dapat dicari menggunakan kode

**Gambar 5.1 Membuat Event Tipe Open Berbayar Umum**



Gambar 5.2 Mencari *Event* Menggunakan Nama



Gambar 5.3 Mendaftar *Event* Berbayar

### 5.2.1.2 Kasus Pengujian Tipe Event Open Berbayar Kode

Pada kasus uji ini, pengujian yang dilakukan yaitu membuat, mencari, dan mendaftar event dengan tipe open berbayar kode.

**Tabel 5.6 Kasus Membuat Event Open Berbayar Kode**

<b>ID</b>	UJ-004
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event Open Berbayar Kode
<b>Nama</b>	Pengujian Membuat Event Open Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu membuat event tipe open berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Buat Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Buat Event”</li> <li>2. Sistem akan menampilkan halaman buat event yang berisi <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, jadwal mulai, jadwal selesai, jenis <i>event</i>, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i></li> <li>3. Pemilik <i>event</i> mengisi <i>field</i> yang tersedia dengan memilih jenis <i>event</i> : open, jenis pendaftaran :berbayar, jenis pencarian : kode dan mengisikan kode <i>event</i>.</li> <li>4. Pemilik event menekan tombol “Buat Event”</li> <li>5. Sistem menyimpan data event ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat membuat <i>event</i> dengan tipe open berbayar kode
<b>Hasil yang diperoleh</b>	<i>Event</i> dengan tipe open berbayar kode berhasil dibuat

<b>ID</b>	UJ-004
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.7 Kasus Mencari Event Open Berbayar Kode**

<b>ID</b>	UJ-005
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mencari Event Open Berbayar Kode
<b>Nama</b>	Pengujian Mencari Event Open Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mencari event tipe open berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu menekan kolom pencarian menggunakan kode <i>event</i></li> <li>4. Tamu mengisikan kode <i>event</i> yang ingin dicari</li> <li>5. Sistem menampilkan <i>event</i> sesuai dengan inputan kode <i>event</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan daftar <i>event</i> sesuai dengan kode <i>event</i> yang dimasukkan
<b>Hasil yang diperoleh</b>	Sistem telah menampilkan daftar <i>event</i> sesuai dengan kode <i>event</i> yang dimasukkan
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.8 Kasus Mendaftar Event Open Berbayar Kode**

<b>ID</b>	UJ-006
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mendaftar Event Open Berbayar Kode

<b>ID</b>	UJ-006
<b>Nama</b>	Pengujian Mendaftar Event Open Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah tamu dapat mendaftar ke event tipe open berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu mencari dan memilih <i>event</i> menggunakan kode <i>event</i>.</li> <li>4. Sistem menampilkan detail <i>event</i></li> <li>5. Tamu menekan tombol “Daftar Event”</li> <li>6. Tamu memilih kategori <i>event</i></li> <li>7. Tamu menekan tombol “Daftar”</li> <li>8. Sistem menyimpan data tamu ke dalam basis data</li> <li>9. Tamu menekan tombol “Upload Bukti Pembayaran”</li> <li>10. Tamu memilih file bukti pembayaran untuk diunggah</li> <li>11. Tamu menekan tombol simpan</li> <li>12. Sistem menyimpan data tamu ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Tamu dapat mendaftar ke <i>event</i> dengan tipe open berbayar kode
<b>Hasil yang diperoleh</b>	Tamu telah mendaftar ke <i>event</i> dengan tipe open berbayar kode
<b>Hasil Pengujian</b>	Berhasil

Jenis Event

Open (tamu dapat registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event)

Close (tamu hanya dapat ditambahkan oleh pemilik event)

Jenis Pendaftaran

- siapapun yang mendaftar akan langsung terverifikasi dan mendapatkan Tiket
- saya akan memilih pendaftar yang dapat mengikuti event ini (tiket didapat setelah disetujui)
- event ini berbayar dan pendaftar harus mengunggah bukti pembayaran untuk mendapatkan tiket

Bank

No Rekening

Atas Nama

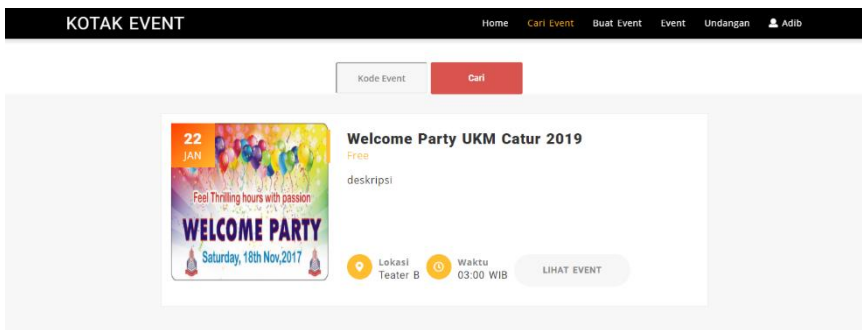
Harga Tiket

Pencarian Event

- Siapapun dapat mencari acara ini
- Acara ini hanya dapat dicari menggunakan kode

Kode unik

**Gambar 5.4 Membuat *Event* Tipe Open Berbayar Kode**



**Gambar 5.5 Mencari *Event* Menggunakan Kode**

### 5.2.1.3 Kasus Pengujian Tipe Event Open Tidak Berbayar Umum

Pada kasus uji ini, pengujian yang dilakukan yaitu membuat, mencari, dan mendaftar event dengan tipe open tidak berbayar umum.

**Tabel 5.9 Kasus Membuat Event Open Tidak Berbayar Umum**

<b>ID</b>	UJ-007
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event Open Tidak Berbayar Kode
<b>Nama</b>	Pengujian Membuat Event Open Tidak Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu membuat event tipe open tidak berbayar umum
<b>Kondisi Awal</b>	Berada di halaman Buat Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Buat Event”</li> <li>2. Sistem akan menampilkan halaman buat event yang berisi <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, jadwal mulai, jadwal selesai, jenis <i>event</i>, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i></li> <li>3. Pemilik <i>event</i> mengisi <i>field</i> yang tersedia dengan memilih jenis <i>event</i> : open, jenis pendaftaran :tidak berbayar, jenis pencarian : umum.</li> <li>4. Pemilik event menekan tombol “Buat Event”</li> <li>5. Sistem menyimpan data event ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat membuat <i>event</i> dengan tipe open tidak berbayar umum
<b>Hasil yang diperoleh</b>	<i>Event</i> dengan tipe open tidak berbayar umum berhasil dibuat
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.10 Kasus Mencari Event Open Tidak Berbayar Umum**

<b>ID</b>	UJ-008
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mencari Event Open Tidak Berbayar Umum
<b>Nama</b>	Pengujian Mencari Event Open Tidak Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mencari event tipe open tidak berbayar umum
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu menekan kolom pencarian menggunakan nama <i>event</i></li> <li>4. Tamu mengisikan nama <i>event</i> yang ingin dicari</li> <li>5. Sistem menampilkan <i>event</i> sesuai dengan inputan nama <i>event</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan daftar <i>event</i> sesuai dengan nama <i>event</i> yang dimasukkan
<b>Hasil yang diperoleh</b>	Sistem telah menampilkan daftar <i>event</i> sesuai dengan nama <i>event</i> yang dimasukkan
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.11 Kasus Mendaftar Event Open Tidak Berbayar Umum**

<b>ID</b>	UJ-009
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mendaftar Event Open Tidak Berbayar Umum



<b>ID</b>	UJ-009
<b>Nama</b>	Pengujian Mendaftar Event Open Tidak Berbayar Umum
<b>Tujuan Pengujian</b>	Menguji apakah tamu dapat mendaftar ke event tipe open tidak berbayar umum
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu mencari dan memilih event menggunakan nama event</li> <li>4. Sistem menampilkan detail <i>event</i></li> <li>5. Tamu menekan tombol “Daftar Event”</li> <li>6. Sistem menyimpan data tamu ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Tamu dapat mendaftar ke <i>event</i> dengan tipe open tidak berbayar umum
<b>Hasil yang diperoleh</b>	Tamu telah mendaftar ke <i>event</i> dengan tipe open tidak berbayar umum
<b>Hasil Pengujian</b>	Berhasil

Jenis Event



Open (tamu dapat registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event)



Close (tamu hanya dapat ditambahkan oleh pemilik event)

Jenis Pendaftaran



siapapun yang mendaftar akan langsung terverifikasi dan mendapatkan Tiket



saya akan memilih pendaftar yang dapat mengikuti event ini (tiket didapat setelah disetujui)



event ini berbayar dan pendaftar harus mengunggah bukti pembayaran untuk mendapatkan tiket

Pencarian Event

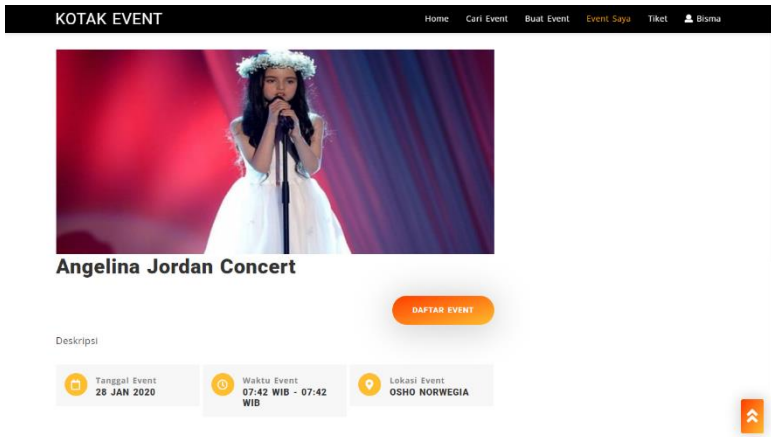


Siapapun dapat mencari acara ini



Acara ini hanya dapat dicari menggunakan kode

**Gambar 5.6 Mendaftar *Event* Tipe Open Tidak Berbayar Umum**



**Gambar 5.7** Mendaftar *Event* Tidak Berbayar

### 5.2.1.4 Kasus Pengujian Tipe Event Open Tidak Berbayar Kode

Pada kasus uji ini, pengujian yang dilakukan yaitu membuat, mencari, dan mendaftar event dengan tipe open berbayar kode.

**Tabel 5.12** Kasus Membuat Event Open Tidak Berbayar Kode

<b>ID</b>	UJ-010
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event Open Tidak Berbayar Kode
<b>Nama</b>	Pengujian Membuat Event Open Tidak Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu membuat event tipe open tidak berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Buat Event Kotakevent

<b>ID</b>	UJ-010
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Buat Event”</li> <li>2. Sistem akan menampilkan halaman buat event yang berisi <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, jadwal mulai, jadwal selesai, jenis <i>event</i>, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i></li> <li>3. Pemilik <i>event</i> mengisi <i>field</i> yang tersedia dengan memilih jenis <i>event</i> : open, jenis pendaftaran :berbayar, jenis pencarian : kode dan mengisikan kode <i>event</i>.</li> <li>4. Pemilik event menekan tombol “Buat Event”</li> <li>5. Sistem menyimpan data event ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat membuat <i>event</i> dengan tipe open tidak berbayar kode
<b>Hasil yang diperoleh</b>	<i>Event</i> dengan tipe open tidak berbayar kode berhasil dibuat
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.13 Kasus Mencari Event Open Tidak Berbayar Kode**

<b>ID</b>	UJ-011
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mencari Event Open Tidak Berbayar Kode
<b>Nama</b>	Pengujian Mencari Event Open Tidak Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mencari event tipe open tidak berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>:</li> </ol>

<b>ID</b>	UJ-011
	<p>foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</p> <ol style="list-style-type: none"> <li>3. Tamu menekan kolom pencarian menggunakan kode <i>event</i></li> <li>4. Tamu mengisikan kode <i>event</i> yang ingin dicari</li> <li>5. Sistem menampilkan <i>event</i> sesuai dengan inputan kode <i>event</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan daftar <i>event</i> sesuai dengan kode <i>event</i> yang dimasukkan
<b>Hasil yang diperoleh</b>	Sistem telah menampilkan daftar <i>event</i> sesuai dengan kode <i>event</i> yang dimasukkan
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.14 Kasus Mendaftar Event Open Tidak Berbayar Kode**

<b>ID</b>	UJ-012
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mendaftar Event Open Tidak Berbayar Kode
<b>Nama</b>	Pengujian Mendaftar Event Open Tidak Berbayar Kode
<b>Tujuan Pengujian</b>	Menguji apakah tamu dapat mendaftar ke event tipe open tidak berbayar kode
<b>Kondisi Awal</b>	Berada di halaman Cari Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Sistem akan menampilkan halaman cari event yang berisi daftar event dengan <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, dan jadwal <i>event</i>.</li> <li>3. Tamu mencari dan memilih event menggunakan kode event</li> <li>4. Sistem menampilkan detail <i>event</i></li> <li>5. Tamu menekan tombol “Daftar Event”</li> <li>6. Sistem menyimpan data tamu ke dalam basis data</li> </ol>

<b>ID</b>	UJ-012
<b>Hasil yang diharapkan</b>	Tamu dapat mendaftar ke <i>event</i> dengan tipe open berbayar umum
<b>Hasil yang diperoleh</b>	Tamu telah mendaftar ke <i>event</i> dengan tipe open berbayar umum
<b>Hasil Pengujian</b>	Berhasil

Jenis Event



Open (tamu dapat registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event)



Close (tamu hanya dapat ditambahkan oleh pemilik event)

Jenis Pendaftaran



siapapun yang mendaftar akan langsung terverifikasi dan mendapatkan Tiket



saya akan memilih pendaftar yang dapat mengikuti event ini (tiket didapat setelah disetujui)



event ini berbayar dan pendaftar harus mengunggah bukti pembayaran untuk mendapatkan tiket

Pencarian Event



Siapapun dapat mencari acara ini



Acara ini hanya dapat dicari menggunakan kode

Kode unik

### Gambar 5.8 Membuat *Event* Tipe Open Tidak Berbayar Kode

#### 5.2.1.5 Kasus Pengujian Tipe Event Close Event

Pada kasus uji ini, pengujian yang dilakukan yaitu membuat, menambah atau mengimport data tamu, dan mengirim tiket dengan tipe close event.

**Tabel 5.15 Kasus Membuat Event Tipe Close Event**

<b>ID</b>	UJ-013
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event Tipe Close Event

<b>ID</b>	UJ-013
<b>Nama</b>	Pengujian Membuat Event Tipe Close Event
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu membuat event tipe close event
<b>Kondisi Awal</b>	Berada di halaman Buat Event Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Buat Event”</li> <li>2. Sistem akan menampilkan halaman buat event yang berisi <i>field</i>: foto, nama <i>event</i>, lokasi <i>event</i>, jadwal mulai, jadwal selesai, jenis <i>event</i>, jenis pendaftaran, jenis pencarian, kategori dan deskripsi <i>event</i></li> <li>3. Pemilik <i>event</i> mengisi <i>field</i> yang tersedia dengan memilih jenis <i>event</i> : close</li> <li>4. Pemilik event menekan tombol “Buat Event”</li> <li>5. Sistem menyimpan data event ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat membuat <i>event</i> dengan tipe open tidak berbayar kode
<b>Hasil yang diperoleh</b>	<i>Event</i> dengan tipe open tidak berbayar kode berhasil dibuat
<b>Hasil Pengujian</b>	Berhasil

Tabel 5.16 Kasus Menambah Data Tamu

<b>ID</b>	UJ-014
<b>Nama Skenario Pengujian</b>	Fungsionalitas Menambah Data Tamu
<b>Nama</b>	Pengujian Menambah Data Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambah data tamu
<b>Kondisi Awal</b>	Berada di halaman Event Saya Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Event Saya”</li> <li>2. Pemilik event memilih <i>event</i> dan menekan tombol “Tamu Undangan”</li> </ol>

<b>ID</b>	UJ-014
	<ol style="list-style-type: none"> <li>3. Sistem menampilkan halaman daftar tamu undangan dari event yang terpilih. Halaman memuat format file import csv, field import file untuk memasukkan data tamu melalui file, tombol “kategori”, tombol “data tamu”, tombol “PDF tamu” untuk mengunduh daftar tamu dalam bentuk PDF, field cari tamu, dan tabel yang berisi daftar tamu.</li> <li>4. Pemilik <i>event</i> menekan tombol “Data Tamu”</li> <li>5. Sistem menampilkan form tambah data tamu</li> <li>6. Pemilik event mengisi form tambah data tamu</li> <li>7. Pemilik event menekan tombol “Simpan”</li> <li>8. Sistem menyimpan data tamu ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menambahkan data tamu sesuai keinginan
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah menambahkan data tamu sesuai keinginan
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.17 Kasus Mengimport Data Tamu**

<b>ID</b>	UJ-015
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengimport Data Tamu
<b>Nama</b>	Pengujian Mengimport Data Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mengimport data tamu
<b>Kondisi Awal</b>	Berada di halaman Event Saya Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Event Saya”</li> <li>2. Pemilik event memilih <i>event</i> dan menekan tombol “Tamu Undangan”</li> </ol>

<b>ID</b>	UJ-015
	<ol style="list-style-type: none"> <li>3. Sistem menampilkan halaman daftar tamu undangan dari event yang terpilih. Halaman memuat format file import csv, field import file untuk memasukkan data tamu melalui file, tombol “kategori”, tombol “data tamu”, tombol “PDF tamu” untuk mengunduh daftar tamu dalam bentuk PDF, field cari tamu, dan tabel yang berisi daftar tamu.</li> <li>4. Pemilik <i>event</i> menekan tombol “Choose File” dan memilih file excel yang ingin diimport</li> <li>5. Pemilik event menekan tombol “Import Data Tamu”</li> <li>6. Sistem menyimpan data tamu ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat mengimport data tamu <i>event</i> menggunakan file excel
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah mengimport data tamu <i>event</i> menggunakan file excel
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.18 Kasus Mengirim Tiket Tamu**

<b>ID</b>	UJ-016
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengirim Tiket Tamu
<b>Nama</b>	Pengujian Mengirim Tiket Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mengirim tiket tamu
<b>Kondisi Awal</b>	Berada di halaman Event Saya Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik event memilih menu “Event Saya”</li> <li>2. Pemilik event memilih <i>event</i> dan menekan tombol “Tamu Undangan”</li> <li>3. Sistem menampilkan halaman daftar tamu undangan dari event yang terpilih. Halaman memuat format file import csv, field import</li> </ol>



<b>ID</b>	UJ-016
	<p>file untuk memasukkan data tamu melalui file, tombol “kategori”, tombol “data tamu”, tombol “PDF tamu” untuk mengunduh daftar tamu dalam bentuk PDF, field cari tamu, dan tabel yang berisi daftar tamu.</p> <ol style="list-style-type: none"> <li>4. Pemilik event memilih data tamu yang ingin dikirim undangan</li> <li>5. Pemilik event menekan tombol “Kirim Tiket”</li> <li>6. Sistem men-generate tiket yang disertai QR Code dalam bentuk file pdf</li> <li>7. Sistem mengirim tiket ke email data tamu yang dipilih</li> <li>8. Sistem menampilkan notifikasi tiket berhasil dikirim</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat mengirim tiket ke email data tamu yang dipilih
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah mengirim tiket ke email data tamu yang dipilih
<b>Hasil Pengujian</b>	Berhasil

Jenis Event



Open (tamu dapat registrasi melalui akun Kotakevent dan dapat ditambahkan oleh pemilik event)



Close (tamu hanya dapat ditambahkan oleh pemilik event)

Kategori

Umum

**Gambar 5.9 Membuat *Event Tipe Close Event***

Choose File data-tamuz.xlsx Import Data Tamu

**ANGELINA JORDAN CONCERT**

→ Kategori → Data Tamu
Cari Event:  PDF Tamu

No	Nama	Email	Alamat	Kategori	Action
1	adib	adib@gmail.com86	Jalan yg lurus	Umum	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
2	Adib Wahyu K	adebwahyu@gmail.com	Jalan yg lurus	VVIP	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
3	adib	adib@gmail.com84	Jalan yg lurus	VIP	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
4	adib	adib@gmail.com83	Jalan yg lurus	Umum	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
5	adib	adib@gmail.com82	Jalan yg lurus	VVIP	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>

**Gambar 5.11 Mengimport Data Tamu Undangan**

Choose File data-tamuz.xlsx Import Data Tamu

**Sukses!**  
Tiket Berhasil Dikirim!

→ Kategori → Data Tamu
Cari Event:  PDF Tamu

No	Nama	Email	Alamat	Kategori	Action
1	adib	adib@gmail.com86	Jalan yg lurus	Umum	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
2	Adib Wahyu K	adebwahyu@gmail.com	Jalan yg lurus	VVIP	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input checked="" type="button" value="Kirim Tiket"/>
3	adib	adib@gmail.com84	Jalan yg lurus	VIP	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>
4	adib	adib@gmail.com83	Jalan yg lurus	Umum	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="button" value="Kirim Tiket"/>

**Gambar 5.10 Mengirim Tiket Ke Email Tamu Undangan**

### 5.2.1.6 Kasus Pengujian Mengelola Pengajuan Pendaftar

Pada kasus uji ini, pengujian yang dilakukan yaitu menyetujui dan menolak pengajuan pendaftar.

**Tabel 5.19 Kasus Menyetujui Pengajuan Pendaftar**

<b>ID</b>	UJ-017
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Pengajuan Pendaftar
<b>Nama</b>	Pengujian Menyetujui Pengajuan Pendaftar
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menyetujui pengajuan pendaftar
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih menu “Pendaftar”</li> <li>2. Sistem menampilkan halaman pendaftar</li> <li>3. Pemilik <i>event</i> memilih data tamu</li> <li>4. Pemilik <i>event</i> menekan tombol konfirmasi</li> <li>5. Sistem menyimpan data tamu ke dalam basis data</li> <li>6. Sistem menampilkan notifikasi konfirmasi berhasil</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menyetujui pengajuan pendaftar
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah menyetujui pengajuan pendaftar
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.20 Kasus Menolak Pengajuan Pendaftar**

<b>ID</b>	UJ-018
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Pengajuan Pendaftar
<b>Nama</b>	Pengujian Menolak Pengajuan Pendaftar
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menolak pengajuan pendaftar
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih menu “Pendaftar”</li> <li>2. Sistem menampilkan halaman pendaftar</li> <li>3. Pemilik <i>event</i> memilih data tamu</li> <li>4. Pemilik <i>event</i> menekan tombol tolak</li> </ol>

<b>ID</b>	UJ-018
	5. Sistem menyimpan data tamu ke dalam basis data 6. Sistem menampilkan notifikasi berhasil ditolak
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menolak pengajuan pendaftar
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah menolak pengajuan pendaftar
<b>Hasil Pengujian</b>	Berhasil

PENDAFTAR EVENT  
REEVA SCHEMATICS 2019

Belum Dikonfirmasi	Semua Pendaftar					
Show	10 entries	Search:	<input type="text"/>			
No	Nama	Email	Status	Bukti_Pembayaran	Kategori	Action
1	tamu1	tamu1@gmail.com	Belum Dikonfirmasi	<a href="#">Lihat</a>	Gold	<a href="#">Konfirmasi</a> <a href="#">Tolak</a>

**Gambar 5.12 Mengelola Pengajuan Pendaftar**

### 5.2.1.7 Kasus Pengujian Mengelola Kategori Tamu

Pada kasus uji ini, pengujian yang dilakukan yaitu menambah dan mengubah kategori.

**Tabel 5.21 Kasus Menambah Kategori Kategori Tamu**

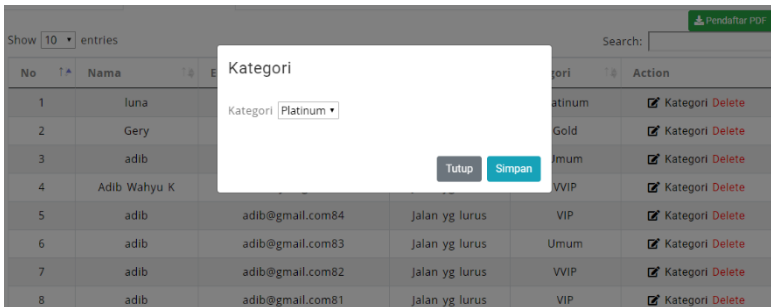
<b>ID</b>	UJ-019
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Kategori Tamu
<b>Nama</b>	Pengujian Menambah Kategori Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambahkan kategori tamu
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”

<b>ID</b>	UJ-019
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih menu “TamU undangan”</li> <li>2. Sistem menampilkan halaman tamu undangan</li> <li>3. Pemilik <i>event</i> menekan tombol “Kategori”</li> <li>4. Sistem menampilkan form tambah kategori</li> <li>5. Pemilik <i>event</i> mengisi nama kategori</li> <li>6. Pemilik <i>event</i> menekan tombol simpan</li> <li>7. Sistem menyimpan data tamu ke dalam basis data</li> <li>8. Sistem menampilkan notifikasi kategori berhasil ditambahkan</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menambah kategori tamu
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> dapat mengubah kategori tamu
<b>Hasil Pengujian</b>	Berhasil

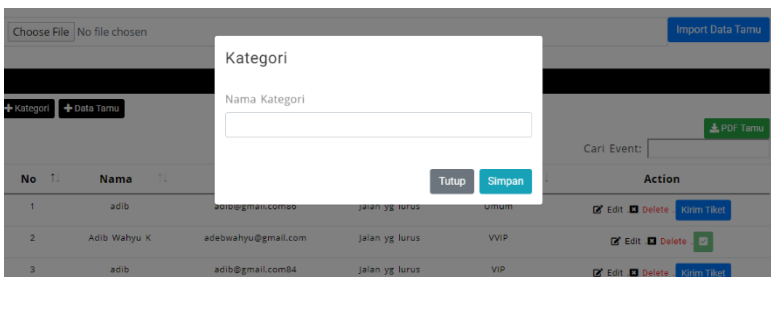
**Tabel 5.22 Kasus Mengubah Kategori Tamu**

<b>ID</b>	UJ-020
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Kategori Tamu
<b>Nama</b>	Pengujian Mengubah Kategori Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambahkan kategori tamu
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih menu “TamU Undangan”</li> <li>2. Sistem menampilkan halaman tamu undangan</li> <li>3. Pemilik <i>event</i> menekan tombol “Kategori”</li> <li>4. Sistem menampilkan form tambah kategori</li> <li>5. Pemilik <i>event</i> mengisi nama kategori</li> </ol>

<b>ID</b>	UJ-020
	<ol style="list-style-type: none"> <li>6. Pemilik <i>event</i> menekan tombol simpan</li> <li>7. Sistem menyimpan data tamu ke dalam basis data</li> <li>8. Sistem menampilkan notifikasi kategori berhasil ditambahkan</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menambah kategori tamu
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> dapat mengubah kategori tamu
<b>Hasil Pengujian</b>	Berhasil



**Gambar 5.14 Mengubah Kategori Tamu**



**Gambar 5.13 Menambah Kategori Tamu**

### 5.2.1.8 Kasus Pengujian Mengelola Event

Pada kasus uji ini, pengujian yang dilakukan yaitu mengubah data *event* dan menambah foto galeri *event*.

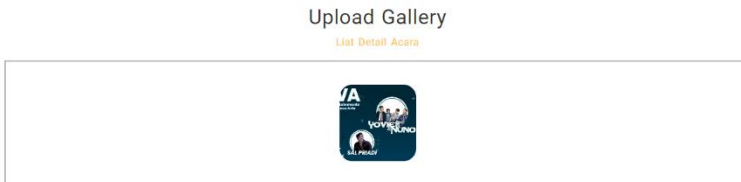
**Tabel 5.23 Kasus Mengubah Data *Event***

<b>ID</b>	UJ-021
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Event
<b>Nama</b>	Pengujian Mengubah Data <i>Event</i>
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mengubah data <i>event</i>
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>2. Pemilik <i>event</i> menekan icon “Edit”</li> <li>3. Sistem menampilkan halaman edit data <i>event</i></li> <li>4. Pemilik <i>event</i> menambah atau mengubah data <i>event</i> sesuai <i>field</i> yang tertera</li> <li>5. Pemilik <i>event</i> menekan tombol simpan</li> <li>6. Sistem menyimpan data <i>event</i> ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat mengubah data <i>event</i>
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah mengubah data <i>event</i>
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.24 Kasus Menambah Foto Galeri *Event***

<b>ID</b>	UJ-022
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Event
<b>Nama</b>	Pengujian Menambah Foto Galeri <i>Event</i>
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambah foto galeri <i>event</i>

<b>ID</b>	UJ-022
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>2. Pemilik <i>event</i> menekan icon “Galeri Foto”</li> <li>3. Sistem menampilkan halaman galeri <i>event</i></li> <li>4. Pemilik <i>event</i> menambah foto <i>event</i> ke <i>field</i> yang tertera</li> <li>5. Sistem menyimpan data <i>event</i> ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menambah foto galeri <i>event</i>
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> dapat menambah foto galeri <i>event</i>
<b>Hasil Pengujian</b>	Berhasil



**Gambar 5.15 Menambah Foto Galeri *Event***

### 5.2.1.9 Kasus Pengujian Mengelola Petugas Scanner

Pada kasus uji ini, pengujian yang dilakukan yaitu menambah dan mengubah dan menghapus data petugas scanner. Kasus pengujian dijelaskan pada tabel



Tabel 5.25 Kasus Menambah Petugas Scanner

<b>ID</b>	UJ-023
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Petugas Scanner
<b>Nama</b>	Pengujian Menambah Petugas Scanner
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menambah petugas scanner
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>2. Pemilik <i>event</i> menekan tombol “Petugas Scanner”</li> <li>3. Sistem menampilkan halaman petugas scanner</li> <li>4. Pemilik <i>event</i> menekan tombol “Petugas Scanner”</li> <li>5. Sistem menampilkan form tambah petugas scanner</li> <li>6. Pemilik <i>event</i> mengisi data petugas scanner</li> <li>7. Pemilik <i>event</i> menekan tombol simpan</li> <li>8. Sistem menyimpan data petugas scanner ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menambah petugas scanner
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah menambah petugas scanner
<b>Hasil Pengujian</b>	Berhasil

Tabel 5.26 Kasus Mengubah Petugas Scanner

<b>ID</b>	UJ-024
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Petugas Scanner
<b>Nama</b>	Pengujian Mengubah Petugas Scanner
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mengubah data petugas scanner

<b>ID</b>	UJ-024
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>2. Pemilik <i>event</i> menekan tombol “Petugas Scanner”</li> <li>3. Sistem menampilkan halaman petugas scanner</li> <li>4. Pemilik <i>event</i> menekan tombol “Edit”</li> <li>5. Sistem menampilkan form edit petugas scanner</li> <li>6. Pemilik <i>event</i> mengisi atau mengubah data petugas scanner</li> <li>7. Pemilik <i>event</i> menekan tombol simpan</li> <li>8. Sistem menyimpan data petugas scanner ke dalam basis data</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat mengubah petugas scanner
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah mengubah petugas scanner
<b>Hasil Pengujian</b>	Berhasil

**Tabel 5.27 Kasus Menghapus Petugas Scanner**

<b>ID</b>	UJ-025
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mengelola Petugas Scanner
<b>Nama</b>	Pengujian Menghapus Petugas Scanner
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menghapus data petugas scanner
<b>Kondisi Awal</b>	Berada di halaman “Event Saya”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>2. Pemilik <i>event</i> memilih salah satu <i>event</i></li> <li>3. Pemilik <i>event</i> menekan tombol “Petugas Scanner”</li> <li>4. Sistem menampilkan halaman petugas scanner</li> </ol>

<b>ID</b>	UJ-025
	<ol style="list-style-type: none"> <li>5. Pemilik <i>event</i> menekan tombol “Delete”</li> <li>6. Sistem menampilkan modal persetujuan hapus data</li> <li>7. Pemilik <i>event</i> memilih tombol “Ya”</li> <li>8. Sistem menghapus data petugas scanner</li> </ol>
<b>Hasil yang diharapkan</b>	Pemilik <i>event</i> dapat menghapus data petugas scanner
<b>Hasil yang diperoleh</b>	Pemilik <i>event</i> telah menghapus data petugas scanner
<b>Hasil Pengujian</b>	Berhasil

ANGELINA JORDAN CONCERT								
+ Petugas Scanner								
No	Ti	Nama	Ti	Email	Ti	Alamat	Ti	Action
1		Sisca		sisca@gmail.com		solo		<a href="#">Edit</a> <a href="#">Delete</a>

Total 1 Data Previous **1** Next

**Gambar 5.16 Tabel Mengelola Petugas Registrasi**

### 5.2.1.10 Kasus Pengujian Mencatat Kehadiran Tamu

Pada kasus uji ini, pengujian yang dilakukan yaitu mencatat kehadiran tamu dengan cara memindai *qr code* tiket. Kasus pengujian dijelaskan pada tabel

**Tabel 5.28 Kasus Mencatat Kehadiran Tamu**

<b>ID</b>	UJ-026
<b>Nama Skenario Pengujian</b>	Fungsionalitas Mencatat Kehadiran Tamu
<b>Nama</b>	Pengujian Mencatat Kehadiran Tamu
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu mencatat kehadiran tamu

<b>ID</b>	UJ-026
<b>Kondisi Awal</b>	Berada di halaman “Mencatat Kehadiran”
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pemilik <i>event</i> menekan tombol “Scan”</li> <li>2. Sistem menampilkan halaman memindai <i>qr code</i></li> <li>3. Pemilik <i>event</i> memindai <i>qr code</i> tiket</li> <li>4. Sistem mengupdate data kehadiran tamu</li> <li>5. Sistem menampilkan pesan kehadiran tamu berhasil dicatat</li> </ol>
<b>Hasil yang diharapkan</b>	Petugas scanner dapat mencatat kehadiran tamu
<b>Hasil yang diperoleh</b>	Petugas scanner telah mencatat kehadiran tamu
<b>Hasil Pengujian</b>	Berhasil



**Gambar 5.17 Mencatat Kehadiran Tamu**

## 5.2.2 Kasus Pengujian Penanganan Error

Pada kasus uji ini, pengujian yang dilakukan yaitu mendaftar pada event yang telah selesai dan petugas scanner melakukan login di luar waktu *event*. Kasus pengujian mendaftar pada event yang telah selesai dijelaskan pada Tabel 5.29

### 5.2.2.1 Tamu Mendaftar Di Event Yang Telah Selesai

Tabel 5.29 Mendaftar Ke *Event* Yang Telah Selesai

<b>ID</b>	UJ-003
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event
<b>Nama</b>	Pengujian mendaftar event yang telah selesai
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menangani waktu pendaftaran <i>event</i>
<b>Kondisi Awal</b>	User tamu belum melakukan registrasi ke <i>event</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Tamu memilih menu “Cari Event”</li> <li>2. Tamu memilih <i>event</i> yang ingin didaftar</li> <li>3. Tamu menekan <i>event</i> yang dipilih</li> <li>4. Sistem menampilkan detail <i>event</i></li> <li>5. Sistem memberikan notifikasi dengan pesan “Event Telah Selesai”</li> </ol>
<b>Hasil yang diharapkan</b>	Tamu tidak bisa mendaftar ke <i>event</i> dan terdapat pesan bahwa <i>event</i> telah selesai
<b>Hasil yang diperoleh</b>	Tamu tidak bisa mendaftar ke <i>event</i> dan terdapat pesan “Event Telah Selesai”
<b>Hasil Pengujian</b>	Berhasil



**Konser Musik**

Deskripsi

Tanggal Event  
16 01 2020 - 20 01 2020

Waktu Event  
08:04 WIB - 11:04 WIB

Lokasi Event  
GEDUNG ROBOTIKA

MANDIRI  
5115100036  
Adib Wahyu Kuncoro

VIP 50.000 | Festival 30.000

**Gambar 5.18 Mendaftar Ke *Event* Yang Telah Selesai**

### 5.2.2.2 Petugas Scanner Login Atau Memindai Tiket Di Luar Waktu *Event*

**Tabel 5.30 Melakukan Login Di Luar Waktu *Event***

<b>ID</b>	UJ-004
<b>Nama Skenario Pengujian</b>	Fungsionalitas Membuat Event
<b>Nama</b>	Pengujian petugas scanner melakukan login di luar waktu <i>event</i>
<b>Tujuan Pengujian</b>	Menguji apakah sistem sudah mampu menangani waktu login atau waktu memindai tiket di luar waktu <i>event</i>
<b>Kondisi Awal</b>	Petugas scanner belum melakukan login ke aplikasi <i>mobile</i> Kotakevent
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Petugas scanner berada di halaman login <i>mobile</i> Kotakevent</li> <li>2. Petugas scanner mengisikan email dan password pada kolom yang tersedia</li> <li>3. Petugas scanner menekan tombol “Login”</li> </ol>

<b>ID</b>	UJ-004
	4. Petugas scanner tidak bisa melakukan login 5. Sistem memberikan notifikasi dengan pesan “Bukan Waktu Acara!”
<b>Hasil yang diharapkan</b>	Petugas scanner tidak dapat melakukan login di luar waktu <i>event</i>
<b>Hasil yang diperoleh</b>	Petugas scanner tidak dapat melakukan login di luar waktu <i>event</i> dan mendapatkan notifikasi dengan pesan “Bukan Waktu Acara!”
<b>Hasil Pengujian</b>	Berhasil

The image shows a login interface for 'K3'. At the top, the logo 'K3' is displayed in yellow. Below it, there are two input fields: 'Email' with the value 'susan@gmail.com' and 'Password' with the value 'adeb007'. A red underline is visible under the email field. Below the password field is an orange 'LOGIN' button. At the bottom of the form, there is a dark grey button with the text 'Bukan Waktu Acara!'.

**Gambar 5.19 Melakukan Login Di Luar Waktu *Event***

### 5.2.3 Kasus Pengujian Response Body

Pengujian response body dilakukan pada endpoint di modul-modul REST API. Setiap endpoint diuji kesesuaian response body berdasarkan skenario yang telah ditentukan. Rincian pengujian terdapat pada Tabel

### 5.2.3.1 Skenario Pengujian Response Body

Terdapat dua skenario pengujian yang diterapkan pada setiap endpoint yaitu skenario A dan B, yang didesain untuk menguji parameter yang dikirim ketika mengakses endpoint. Rincian scenario terdapat pada Tabel 5.43

**Tabel 5.31 Skenario Pengujian Response Body**

Kode Skenario	Nama Skenario	Deskripsi
A	Parameter Lengkap	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan mengirim seluruh parameter yang dibutuhkan oleh <i>endpoint</i>.</li> <li>• Pengujian berhasil jika <i>response</i> berisikan data</li> </ul>
B	Parameter Tidak Lengkap	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan mengirim sebagian parameter yang dibutuhkan oleh <i>endpoint</i>.</li> <li>• Pengujian berhasil jika <i>response</i> bernilai null</li> </ul>



**Tabel 5.32 Pengujian API Endpoint**

Kode	Endpoint URI	Method	Deskripsi	Skenario	
				A	B
RA01	<i>/getListEvent</i>	<i>POST</i>	Mendapatkan data tamu yang telah melakukan registrasi	✓	✓
RA02	<i>/scan</i>	<i>POST</i>	Mendapatkan konten qr code dan memperbarui data kehadiran	✓	✓
RA03	<i>/search</i>	<i>POST</i>	Mencari data tamu yang telah melakukan registrasi	✓	✓

#### 5.2.4 Pengujian Terhadap Pengguna

Pengujian ini dilakukan untuk mengetahui penilaian dan tanggapan dari pengguna terhadap sejumlah aspek dari aplikasi Kotakevent. Pengujian ini dilakukan dengan menggunakan sistem Likert, suatu skala psikometrik yang umum digunakan dalam kuisisioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Ada dua bentuk pertanyaan yang menggunakan Likert yaitu pertanyaan positif untuk mengukur minat positif, dan untuk pertanyaan negatif untuk mengukur minat negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1, sedangkan pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5. Kemudian disediakan lima pilihan skala dengan format sebagai berikut :

Keterangan pengisian kuisioner.

1. Sangat Tidak Setuju (STS)
2. Tidak Setuju (TS)
3. Ragu-ragu (R)
4. Setuju (S)
5. Sangat Setuju (SS)

Dalam pelaksanaan kuisioner ini melibatkan Ketua BEM FTIK ITS 2019/2020 dan Ketua UKM Catur ITS 2015/2016 yang merupakan orang yang telah berpengalaman dalam mengadakan sebuah acara/*event*. Pengujian dilakukan dengan *menghosting* sistem agar dapat diakses melalui jaringan internet. Pengguna membuka tautan untuk mencoba aplikasi kemudian pengguna mengisi kuisioner dari penulis dalam *form* kuisioner yang disediakan. Tabel 5.33 merupakan data responden yang telah mengisi kuisioner mengenai sistem yang telah dibangun.

**Tabel 5.33 Data Pengisi Kuisioner**

No	Nama	Keterangan
1	Aldinata Rizky Revanda	Ketua BEM FTIK ITS 2019/2020
2	Rifqi Maula Iqbal	Ketua UKM Catur ITS 2015/2016

Pada Tabel 5.34 menunjukkan hasil kuisioner dari responden untuk pemilik *event* dan petugas scanner, dan Tabel 5.35 menunjukkan hasil kuisioner untuk tamu. Kemudian dari hasil kuisioner data diolah untuk mengetahui persentase dari masing-masing alternatif jawaban pada setiap soal yang diberikan kepada masing-masing responden.

**Tabel 5.34 Hasil Rekapitulasi Kuisioner Pemilik *Event* dan Petugas Scanner**

No	Pertanyaan	Nilai				
		STS	TS	R	S	SS
1	Apakah informasi yang disediakan oleh aplikasi ini mudah dipahami?				2	
2	Apakah penggunaan menu aplikasi mudah digunakan?				2	
3	Apakah aplikasi ini mudah dioperasikan?				1	1
4	Apakah aplikasi ini mudah dipelajari?				1	1
5	Apakah aplikasi ini mudah digunakan untuk mengelola sebuah event?			1	1	
6	Apakah tampilan menu dalam aplikasi mudah untuk dikenali?				2	
7	Apakah aplikasi ini sesuai dengan kebutuhan?				2	
8	Apakah aplikasi ini mempunyai kemampuan dan fungsi sesuai yang diharapkan?				2	
9	Apakah proses membuat event mudah dipahami oleh pemilik event?				2	
10	Apakah proses mengelola pendaftar mudah dipahami oleh pemilik event?			1	1	
11	Apakah proses mengelola data tamu mudah dipahami oleh pemilik event?				2	
12	Apakah proses mengelola kategori tamu mudah dipahami oleh pemilik event?			1	1	
13	Apakah proses mengelola petugas scanner mudah dipahami oleh pemilik event?				2	
14	Apakah proses mencatat kehadiran tamu mudah dipahami oleh petugas scanner?				2	
15	Apakah aplikasi ini jarang error (bermasalah)?				1	1
16	Apakah aplikasi ini jarang not responding?					2

No	Pertanyaan	Nilai				
		STS	TS	R	S	SS
17	Secara keseluruhan, apakah aplikasi ini memuaskan?				2	

**Tabel 5.35 Hasil Rekapitulasi Kuisiонер Tamu**

No	Pertanyaan	Nilai				
		STS	TS	R	S	SS
1	Apakah informasi yang disediakan oleh aplikasi ini mudah dipahami?				2	
2	Apakah penggunaan menu aplikasi mudah digunakan?				2	
3	Apakah aplikasi ini mudah dioperasikan?				1	1
4	Apakah aplikasi ini mudah dipelajari?				2	
5	Apakah tampilan menu dalam aplikasi mudah untuk dikenali?				1	1
6	Apakah aplikasi ini sesuai dengan kebutuhan?				1	1
7	Apakah aplikasi ini mempunyai kemampuan dan fungsi sesuai yang diharapkan?			2		
8	Apakah aplikasi ini mudah digunakan untuk mendaftar sebuah event?				2	
9	Apakah proses mendaftar event tidak berbayar mudah dipahami oleh tamu?				2	
10	Apakah proses mendaftar event berbayar mudah dipahami oleh tamu?				2	
11	Apakah proses mengunggah bukti pembayaran mudah dipahami oleh tamu?				1	1
12	Apakah proses mengunduh tiket mudah dipahami oleh tamu?					2
13	Apakah aplikasi ini jarang error (bermasalah)?				2	

No	Pertanyaan	Nilai				
		STS	TS	R	S	SS
14	Apakah aplikasi ini jarang not responding?				2	
15	Secara keseluruhan, apakah aplikasi ini memuaskan?				2	

**Tabel 5.36 Persentase Alternatif Jawaban Kuisioner Pemilik *Event* Dan Petugas Scanner**

Pertanyaan	Penilaian				
	STS	TS	R	S	SS
1	0	0	0	2	0
Persentase	0	0	0	100%	0
2	0	0	0	2	0
Persentase	0	0	0	100%	0
3	0	0	0	1	1
Persentase	0	0	0	50%	50%
4	0	0	0	1	1
Persentase	0	0	0	50%	50%
5	0	0	1	1	0
Persentase	0	0	50%	50%	0
6	0	0	0	2	0
Persentase	0	0	0	100%	0
7	0	0	0	2	0
Persentase	0	0	0	100%	0
8	0	0	0	2	0
Persentase	0	0	0	100%	0
9	0	0	0	2	0
Persentase	0	0	0	100%	0
10	0	0	1	1	0
Persentase	0	0	50%	50%	0
11	0	0	0	2	0
Persentase	0	0	0	100%	0
12	0	0	1	1	0
Persentase	0	0	50%	50%	0

Pertanyaan	Penilaian				
	STS	TS	R	S	SS
13	0	0	0	2	0
Persentase	0	0	0	100%	0
14	0	0	0	2	0
Persentase	0	0	0	100%	0
15	0	0	0	1	1
Persentase	0	0	0	50%	50%
16	0	0	0		2
Persentase	0	0	0		100%
17	0	0	0	2	
Persentase	0	0	0	100%	

**Tabel 5.37 Persentase Alternatif Jawaban Kuisisioner Tamu**

Pertanyaan	Penilaian				
	STS	TS	R	S	SS
1	0	0	0	2	0
Persentase	0	0	0	100%	0
2	0	0	0	2	0
Persentase	0	0	0	100%	0
3	0	0	0	1	1
Persentase	0	0	0	50%	50%
4	0	0	0	2	0
Persentase	0	0	0	100%	0
5	0	0	0	1	1
Persentase	0	0	0	50%	50%
6	0	0	0	1	1
Persentase	0	0	0	50%	50%
7	0	0	1	1	0
Persentase	0	0	50%	50%	0
8	0	0	0	2	0
Persentase	0	0	0	100%	0
9	0	0	0	2	0
Persentase	0	0	0	100%	0

Pertanyaan	Penilaian				
	STS	TS	R	S	SS
10	0	0	0	2	0
Persentase	0	0	0	100%	0
11	0	0	0	1	1
Persentase	0	0	0	50%	50%
12	0	0	1	0	2
Persentase	0	0	50%	0	100%
13	0	0	0	2	0
Persentase	0	0	0	100%	0
14	0	0	0	2	0
Persentase	0	0	0	100%	0
15	0	0	0	2	0
Persentase	0	0	0	100%	0

Kemudian pada Tabel 5.38 ditunjukkan Persentase interpretasi penilaian kuisioner untuk setiap pertanyaan kuisioner. Untuk mendapatkan inerpretasi penilaian maka dilakukan langkah-langkah sebagai berikut :

1. Mencari skor ideal untuk pertanyaan yang diajukan kepada responden sebagai berikut :
  - a. Skor tertinggi : (SS) 5x2 responden = 10
  - b. Skor terendah : (STS) 1x2 responden = 2
2. Menghitung hasil interpretasi dengan: Total skor / Skor tertinggi x 100%
3. Melihat hasil interpretasi hasil penelitian dengan melihatnya dari persentase skala Likert

**Tabel 5.38 Persentasi Interpretasi Penilaian Kuisisioner Pemilik Event dan Petugas Scanner**

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	R (3)	S (4)	SS (5)		
1	0	0	0	2	0	8	80%
2	0	0	0	2	0	8	80%
3	0	0	0	1	1	9	90%
4	0	0	0	1	1	9	90%
5	0	0	1	1	0	7	70%
6	0	0	0	2	0	8	80%
7	0	0	0	2	0	8	80%
8	0	0	0	2	0	8	80%
9	0	0	0	2	0	8	80%
10	0	0	1	1	0	7	70%
11	0	0	0	2	0	8	80%
12	0	0	1	1	0	7	70%
13	0	0	0	2	0	8	80%
14	0	0	0	2	0	8	80%
15	0	0	0	1	1	9	90%
16	0	0	0	0	2	10	100%
17	0	0	0	2	0	8	80%

**Tabel 5.39 Persentasi Interpretasi Penilaian Kuisisioner Tamu**

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	R (3)	S (4)	SS (5)		
1	0	0	0	2	0	8	80%
2	0	0	0	2	0	8	80%
3	0	0	0	1	1	9	90%



No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	R (3)	S (4)	SS (5)		
4	0	0	0	2	0	8	80%
5	0	0	0	1	1	9	90%
6	0	0	0	1	1	9	90%
7	0	0	1	1	0	7	70%
8	0	0	0	2	0	8	80%
9	0	0	0	2	0	8	80%
10	0	0	0	2	0	8	80%
11	0	0	0	1	1	9	90%
12	0	0	0	0	2	10	100%
13	0	0	0	2	0	8	80%
14	0	0	0	2	0	8	80%
15	0	0	0	2	0	8	80%

**Tabel 5.40 Interpretasi Skor Hasil Penilaian Kuisisioner Pemilik *Event* Dan Petugas Scanner**

No	STS 20%	TS 40%	R 60%	S 80%	SS 100%
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

No	STS 20%	TS 40%	R 60%	S 80%	SS 100%
15					
16					
17					

**Tabel 5.41 Interpretasi Skor Hasil Penilaian Kuisisioner Tamu**

No	STS 20%	TS 40%	R 60%	S 80%	SS 100%
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

Dari Tabel 5.40 dapat disimpulkan bahwa sebanyak 100% responden setuju terhadap kemudahan pengoperasian sistem, kemudahan tampilan dan kemudahan dalam mengelola *event* yaitu mengelola data tamu, mengelola kategori tamu dan mencatat kehadiran tamu. Dan pada Tabel 5.41 dapat disimpulkan bahwa sebanyak 100% responden setuju terhadap kemudahan pengoperasian sistem, kemudahan tampilan dan kemudahan dalam melakukan registrasi ke sebuah *event*. Kemudian pada Tabel 5.42 merupakan hasil kuisisioner penilaian sistem.

**Tabel 5.42 Hasil Kuisisioner Penilaian Sistem**

No	Nama	Kritik Saran
1	Aldinata Rizky Revanda	<ol style="list-style-type: none"> <li>1. Untuk aplikasi <i>mobile</i> mungkin bisa ditambahkan fitur untuk mengisi <i>report</i> atau keterangan tamu, misalnya tamu tertangkap membawa senjata tajam maka bisa dimasukkan di <i>report</i>.</li> <li>2. Untuk tampilan <i>website</i> bisa dibuat lebih menarik lagi sehingga baik pemilik <i>event</i> maupun calon peserta yang ingin mencari <i>event</i> lebih tertarik.</li> </ol>
2	Rifqi Maula Iqbal	<ol style="list-style-type: none"> <li>1. Aplikasi sudah bagus dan sudah bisa digunakan, tetapi dari segi keamanan <i>web</i> masih minim karena belum ada <i>enkripsi url</i> pada <i>web</i> sehingga <i>url</i> bisa diakses sembarang orang.</li> <li>2. Bisa dibuat lebih menarik lagi, dan bisa ditambahkan aplikasi <i>mobile</i> khusus untuk pemilik <i>event</i>.</li> </ol>

### 5.3 Rekapitulasi Kasus Pengujian

Rekapitulasi dari kasus pengujian yang telah dilakukan. Hasil rekapitulasi diantaranya adalah hasil pengujian fungsional, dan hasil pengujian response body.

#### 5.3.1 Hasil Pengujian Fungsional

Terdapat tiga puluh fungsional yang dimiliki Kotakevent. Pengujian fungsional dijelaskan pada Tabel 5.43.

**Tabel 5.43 Rekapitulasi Hasil Pengujian Aplikasi Kotakevent**

No	Fungsionalitas	Kode Uji	Kasus Uji	Terpenuhi
1	Skenario pengujian tipe event open berbayar umum	UJ-001	Membuat Event	✓
		UJ-002	Mencari Event	✓
		UJ-003	Mendaftar Event	✓
2	Skenario pengujian tipe event open berbayar kode	UJ-004	Membuat Event	✓
		UJ-005	Mencari Event	✓
		UJ-006	Mendaftar Event	✓
3	Skenario pengujian tipe event open tidak berbayar umum	UJ-007	Membuat Event	✓
		UJ-008	Mencari Event	✓
		UJ-009	Mendaftar Event	✓
4	Skenario pengujian tipe event open tidak berbayar kode	UJ-010	Membuat Event	✓
		UJ-011	Mencari Event	✓
		UJ-012	Mendaftar Event	✓
5	Skenario pengujian tipe event close event	UJ-013	Membuat Event	✓
		UJ-014	Menambah data tamu	✓
		UJ-015	Mengimport data tamu	✓
		UJ-016	Mengirim tiket tamu	✓
6		UJ-017	Menyetujui pengajuan pendaftar	✓

No	Fungsionalitas	Kode Uji	Kasus Uji	Terpenuhi
	Skenario pengujian mengelola pengajuan pendaftar	UJ-018	Menolak pengajuan pendaftar	✓
7	Skenario pengujian mengelola kategori tamu	UJ-019	Menambah kategori tamu	✓
		UJ-020	Mengubah kategori tamu	✓
8	Skenario pengujian mengelola <i>event</i>	UJ-021	Mengubah data <i>event</i>	✓
		UJ-022	Menambah foto galeri <i>event</i>	✓
9	Skenario pengujian mengelola petugas scanner	UJ-023	Menambah petugas scanner	✓
		UJ-024	Mengubah petugas scanner	✓
		UJ-025	Menghapus petugas scanner	✓
10	Skenario pengujian mencatat kehadiran tamu	UJ-026	Mencatat kehadiran tamu	✓

### 5.3.2 Hasil Pengujian Penanganan Error

Tabel 5.44 Rekapitulasi Hasil Pengujian Penanganan Error

No	Kasus Uji	Kode Uji	Terpenuhi
1	Mendaftar ke <i>event</i> yang telah selesai	UJ-017	✓
2	Melakukan login atau memindai tiket di luar waktu <i>event</i>	UJ-018	✓

### 5.3.3 Hasil Pengujian Response Body

Pengujian response body dilakukan pada endpoint di modul-modul REST API. Setiap endpoint diuji kesesuaian

response body berdasarkan skenario yang telah ditentukan. Rincian rekapitulasi hasil pengujian terdapat pada Tabel 5.45.

**Tabel 5.45 Rekapitulasi Hasil Pengujian API Endpoint**

No	Endpoint URL	Method	Deskripsi	Terpenuhi
1.	<i>/getListEvent</i>	<i>POST</i>	Mendapatkan data tamu yang telah hadir dalam <i>event</i>	✓
2.	<i>/scan</i>	<i>POST</i>	Mendapatkan konten qr code dan mencatat data kehadiran	✓
3.	<i>/search</i>	<i>POST</i>	Mencari data tamu yang telah hadir	✓

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini berdasarkan hasil pengujian dan hal lainnya yang telah dilakukan. Selain itu, juga terdapat beberapa saran terhadap tugas akhir yang penulis ajukan terhadap pengembangan Tugas Akhir ini untuk pengembangan kedepannya.

#### **6.1 Kesimpulan**

Berdasarkan pengerjaan Tugas Akhir ini dan hasil yang didapatkan pada tahap uji coba aplikasi, penulis mengambil beberapa kesimpulan sebagai berikut.

1. Aplikasi Kotakevent dapat men-generate QR Code pada undangan/tiket dalam bentuk file pdf. QR Code berisi enkripsi dari id\_user dan id\_event yang akan menjadi kode unik saat melakukan registrasi.
2. Aplikasi Kotakevent dapat memverifikasi tamu yang hadir dengan cara menscan QR Code pada undangan/tiket. Proses registrasi dilakukan oleh Petugas Scanner, dimana pemilik event dapat memilih siapa saja yang dapat menjadi Petugas Scanner.
3. Aplikasi Kotakevent dapat mengelola data tamu undangan berdasarkan kategorinya dengan cara menambah atau mengubah kategori tamu undangan sesuai keinginan pemilik event.

#### **6.2 Saran**

Terdapat beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Perlu penggalian kebutuhan lebih lanjut terhadap pengguna terkait untuk pengembangan sistem.
2. Perlu penambahan aplikasi mobile khusus untuk pemilik event agar dapat lebih mudah dalam memantau atau mengelola event.



## DAFTAR PUSTAKA

- [1] “How to create QRcode.” [Online]. Available: [http://www.swetake.com/qrcode/qr1\\_en.html](http://www.swetake.com/qrcode/qr1_en.html). [Accessed: 24-Dec-2019].
- [2] “PHP QR Code - QR code generator, an LGPL PHP library.” [Online]. Available: <http://phpqrcode.sourceforge.net/>. [Accessed: 23-Jan-2020].
- [3] “ZXing – About.” [Online]. Available: <https://zxing.github.io/zxing/>. [Accessed: 24-Dec-2019].
- [4] “ZBar bar code reader.” [Online]. Available: <http://zbar.sourceforge.net/>. [Accessed: 24-Dec-2019].
- [5] “QRCodeReader (ZXing 3.4.0 API).” [Online]. Available: <https://zxing.github.io/zxing/apidocs/>. [Accessed: 24-Dec-2019].
- [6] “ZBar Bar Code Reader Library: Main Page.” [Online]. Available: <http://zbar.sourceforge.net/api/>. [Accessed: 24-Dec-2019].
- [7] “Android - Overview - Tutorialspoint.” [Online]. Available: [https://www.tutorialspoint.com/android/android\\_overview.htm](https://www.tutorialspoint.com/android/android_overview.htm). [Accessed: 23-Jan-2020].
- [8] M. M. Luthfi, “Pengertian dan Keunggulan Framework Laravel,” *IDCloudHost*, 07-Jun-2016. [Online]. Available: <https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>. [Accessed: 07-Jan-2020].
- [9] P. A. R. Novina, “Pengenalan Framework Laravel,” *Laboratorium Enterprise Application*, 19-Sep-2014. [Online]. Available: <http://lea.si.fti.unand.ac.id/2014/09/pengenalan-framework-laravel/>. [Accessed: 07-Jan-2020].
- [10] “Composer.” [Online]. Available: <https://getcomposer.org/>. [Accessed: 26-Dec-2019].
- [11] “Pengertian dan Fungsi PHP dalam Pemrograman Web | Duniaikom,” 22-Jul-2019. [Online]. Available: <https://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/>. [Accessed: 26-Dec-2019].

- [12]“The Java Language Environment.” [Online]. Available: <https://www.oracle.com/technetwork/java/intro-141325.html>. [Accessed: 24-Dec-2019].
- [13]“MVC Framework - Introduction - Tutorialspoint.” [Online]. Available: [https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm). [Accessed: 26-Dec-2019].
- [14]G. M. Rama dan A. Kak, “Some structural measures of API usability,” *Software: Practice and Experience*, vol. 45, no. 1, hlm. 75–110, 2015.
- [15]“JSON.” [Online]. Available: <https://www.json.org/json-en.html>. [Accessed: 26-Dec-2019].

## BIODATA PENULIS



Penulis lahir di Sukoharjo, 23 Oktober 1997. Penulis telah menempuh pendidikan dasar di MIN Jetis Sukoharjo, kemudian untuk pendidikan menengah pertama di SMPN 1 Sukoharjo dan di jenjang menengah atas di SMAN 1 Sukoharjo. Sejak kecil, penulis suka dengan hal-hal terkait perkembangan teknologi dan perkembangan komputer. Hal tersebut juga yang mendasari penulis melanjutkan pendidikan sarjana S1 di Jurusan Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Rekayasa Perangkat Lunak serta memiliki ketertarikan di bidang Sistem dan Manajemen Basis Data, Pemrograman *Web*, serta Pemrograman Android. Untuk komunikasi, penulis dapat dihubungi melalui surel: [adibkuncoro8757@gmail.com](mailto:adibkuncoro8757@gmail.com).