



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM INFORMASI *EVENT*
KEAGAMAAN BERBASIS WEB MENGGUNAKAN
FRAMEWORK LARAVEL**

SKRIPSI

AUFA BILLAH PUTRA JAZAMA

0110215039

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

SEPTEMBER 2019



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM INFORMASI *EVENT*
KEAGAMAAN BERBASIS WEB MENGGUNAKAN
FRAMEWORK LARAVEL**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer**

**AUFA BILLAH PUTRA JAZAMA
0110215039**

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
SEPTEMBER 2019**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya penulis,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Aufa Billah Putra Jazama
NIM : 0110215039

Depok, 9 September 2019

Aufa Billah Putra Jazama

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Aufa Billah Putra Jazama
NIM : 0110215039
Program Studi : Teknik Informatika
Judul Skripsi : Rancang Bangun Sistem Informasi Event Keagamaan
Berbasis Web Menggunakan Framework Laravel.

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

DEWAN PENGUJI

Pembimbing : Sirojul Munir, S.Si, M.Kom. ()
Penguji I : Ahmad Rio Adriansyah, S.Si. M.Si. ()
Penguji II : Hilmy Abidzar Tawakal, ST., M.Kom ()

Ditetapkan di : Depok

Tanggal : 9 September 2019

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Skripsi ini. Penulisan Skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan Skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi, ST, M.M., M.T selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Bapak Ahmad Rio Adriansyah, S.Si. M.Si selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Indra Hermawan, M.Kom. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama perkuliahan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Sirojul Munir, S.Si, M.Kom selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
8. Teman-teman angkatan 2015 (Vixagraph) yang bersama-sama berjuang dari awal perkuliahan hingga akhir perkuliahan.
9. Setiap orang yang namanya tidak bisa disebutkan satu per satu yang telah memberikan dukungan dalam menyelesaikan penulisan ilmiah ini.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 9 September 2019

Aufa Billah Putra Jazama

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Aufa Billah Putra Jazama
NIM : 0110215039
Program Studi : Teknik Informatika
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty – Free Right*) atas karya ilmiah saya yang berjudul:

Rancang Bangun Sistem Informasi Event Keagamaan Berbasis Web Menggunakan *Framework* Laravel.

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 9 September 2019

Yang menyatakan

(Aufa Billah)

ABSTRAK

Nama : Aufa Billah Putra Jazama
NIM : 0110215039
Program Studi : Teknik Informatika
Judul : Rancang Bangun Sistem Informasi Event Keagamaan Berbasis Web Menggunakan Framework Laravel.

Website saat ini digunakan sebagai sarana penyebaran informasi dan media penghubung antara dua pihak yang membutuhkan. Penyebaran informasi *event* keagamaan masih disebarkan melalui media sosial. Hal ini menyebabkan penyebaran informasi belum merata sehingga informasi tidak tersampaikan dengan baik kepada *user*.

Website diperlukan sebagai sarana penyebaran informasi dan sebagai media penghubung antara dua pihak dalam membuat dan mencari *event* keagamaan. Pembuat *event* dapat membuat *event* secara terpusat sehingga *event* lebih mudah dikelola dan ditemukan melalui platform website. *User* dapat menemukan *event* keagamaan secara terpusat sehingga memudahkan *user* dalam mencari *event* melalui platform website.

Sistem ini dibangun berbasis website menggunakan *framework* Bootstrap untuk *frontend*, *framework* Laravel untuk *backend*, MySQL untuk *database*, dan XAMPP untuk *webserver*. Sistem ini mampu menyajikan *event* keagamaan kepada *user*. *User* dapat menggunakan fitur *join event*, mengelola *event*, mengunduh tiket *event*, mengelola akun, *dashboard user*.

Kata kunci: Sistem Informasi, *Event* Keagamaan, Laravel, Website.

ABSTRACT

Name : Aufa Billah Putra Jazama
NIM : 0110215039
Study Program : Teknik Informatika
Title : Design and Build a Web-Based Religious Event Information System Using the Framework Laravel.

Today website is used as a medium of information dissemination and medium between two parties in need. Information dissemination on religious event still distributed through social media. This causes spread of information not evenly distributed so the information is not conveyed properly to the users.

Website is needed as a information dissemination and as a media liaison between two parties in making and searching for religious event. Event creator can create events are more easily managed and found through the website. Users can find religious events centrally making it easier for users to find events through the website.

This system is built based on a website using Bootstrap framework for frontend, Laravel framework for backend, MySQL for database, and XAMPP for webserver. System is abto to present religious event to users. Users can use event join, manage event, download event tickets, manage account, user dashboard feature.

Key words: Information System, Religious Event, Laravel, Website.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	1
ABSTRAK.....	2
DAFTAR ISI.....	4
DAFTAR GAMBAR.....	7
DAFTAR TABEL.....	9
DAFTAR LAMPIRAN.....	10
BAB I PENDAHULUAN.....	11
1.1 Latar Belakang.....	11
1.2 Perumusan Masalah.....	12
1.3 Tujuan dan Manfaat Penelitian.....	13
1.4 Batasan Masalah.....	13
1.5 Sistematika Penulisan.....	14
BAB II KAJIAN LITERATUR.....	15
2.1 Tinjauan Pustaka.....	15
2.1.1 Teori Sistem Informasi.....	15
2.1.2 Web <i>Framework</i>	15
2.1.3 MVC.....	16
2.1.4 Laravel.....	17
2.1.5 MySQL.....	17
2.1.6 Waterfall.....	18
2.1.7 UML.....	18
2.1.8 Black Box Testing.....	21
2.1.9 Skala Likert.....	22
2.2 Penelitian Terkait.....	23
2.3 Aplikasi Terkait.....	25

BAB III METODOLOGI PENELITIAN.....	26
3.1 Tahapan Penelitian.....	26
3.2 Rancangan Penelitian.....	29
3.2.1 Metode Penelitian.....	29
3.2.2 Jenis Penelitian.....	29
3.2.3 Metode Pengumpulan Data.....	29
3.2.4 Lingkungan Pengembangan.....	30
3.2.4.1 Tempat/Lokasi.....	30
3.2.4.2 Bahan dan Alat.....	31
3.3 Metode Pengembangan Perangkat Lunak.....	32
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	34
4.1 Analisis Sistem.....	34
4.1.1 Istilah Baku.....	36
4.1.2 <i>User Requirement</i>	37
4.1.3 Use Case.....	39
4.2 Perancangan Sistem.....	45
4.2.1 Desain Sistem.....	45
4.2.2 Deployment Diagram.....	60
4.2.3 Antarmuka Sistem.....	61
4.3 Rencana Pengujian.....	68
4.3.1 Pengujian Fungsionalitas: <i>Black Box Testing</i>	68
4.3.2 Pengujian User: <i>User Acceptance Test</i> (UAT).....	71
4.3.3 Kuesioner.....	72
BAB V IMPLEMENTASI DAN EVALUASI.....	73
5.1 Implementasi.....	73
5.1.1 Implementasi antarmuka.....	73
5.2 Evaluasi Sistem.....	85
5.2.1 Hasil <i>Black Box Testing</i>	85
5.2.2 UAT (<i>User Acceptance Test</i>).....	88
5.2.3 Hasil kuesioner.....	90

BAB VI Kesimpulan dan saran.....	94
6.1 Kesimpulan.....	94
6.2 Saran.....	95
DAFTAR PUSTAKA.....	96
LAMPIRAN.....	97

DAFTAR GAMBAR

Gambar 1: Tahapan Penelitian.....	26
Gambar 2: Use Case Diagram.....	39
Gambar 3: Domain Model.....	45
Gambar 4: Sequence Diagram Melihat Event.....	47
Gambar 5: Sequence Diagram Register.....	48
Gambar 6: Sequence Diagram Login.....	49
Gambar 7: Sequence Diagram Join Event.....	50
Gambar 8: Sequence Diagram Menampilkan Tiket.....	51
Gambar 9: Sequence Diagram Mengelola Event.....	52
Gambar 10: Collaboration Diagram Melihat Event.....	54
Gambar 11: Collaboration Diagram Register.....	55
Gambar 12: Collaboration Diagram Login.....	56
Gambar 13: Collaboration Diagram Join Event.....	57
Gambar 14: Collaboration Diagram Menampilkan Tiket.....	58
Gambar 15: Collaboration Diagram Mengelola Event.....	59
Gambar 16: Deployment Diagram.....	60
Gambar 17: Halaman Home.....	61
Gambar 18: Explore.....	62
Gambar 19: Detail Event.....	63
Gambar 20: Halaman Register.....	64
Gambar 21: Halaman Login.....	64
Gambar 22: Halaman Dashboard.....	65
Gambar 23: Create Event.....	65
Gambar 24: Halaman My Event.....	66
Gambar 25: Halaman My Account.....	66
Gambar 26: Halaman Password.....	67
Gambar 27: Halaman Home.....	74
Gambar 28: Halaman Explore.....	75

Gambar 29: Halaman Detail Event.....	76
Gambar 30: Halaman Register.....	77
Gambar 31: Halaman Login.....	78
Gambar 32: Halaman Dashboard.....	79
Gambar 33: Halaman Create Event.....	80
Gambar 34: Halaman My Event.....	81
Gambar 35: Halaman My Ticket.....	82
Gambar 36: Halaman My Account.....	83
Gambar 37: Halaman Password.....	84

DAFTAR TABEL

Table 1: Penelitian Terkait.....	24
Table 2: Penelitian Sebelumnya.....	34
Table 3: Perbandingan Website.....	35
Table 4: Hasil Perbandingan Website.....	35
Table 5: Hasil Analisa Sistem.....	36
Table 6: Istilah Baku.....	36
Table 7: Kebutuhan User.....	37
Table 8: User Aplikasi.....	37
Table 9: Menu Aplikasi.....	38
Table 10: Melihat Event.....	40
Table 11: Mendaftar Sebagai User.....	40
Table 12: Login sebagai User.....	41
Table 13: Join Event.....	42
Table 14: Menampilkan Tiket Event.....	42
Table 15: Mengelola Event.....	43
Table 16: Mengelola Akun.....	44
Table 17: Black Box Testing.....	68
Table 18: UAT.....	71
Table 19: Skala Likert.....	72
Table 20: Kuisisioner.....	72
Table 21: Hasil Black Box Testing.....	85
Table 22: Hasil UAT.....	88
Table 23: Kuisisioner.....	90
Table 24: Skala Likert.....	90
Table 25: Hasil Kuisisioner.....	91
Table 26: Hasil Perhitungan Skala Likert.....	92
Table 27: Hasil Rata-Rata Keseluruhan.....	93

DAFTAR LAMPIRAN

Lampiran 1: Hasil UAT (User Acceptance Test)

Lampiran 2: Hasil Kuesioner

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini website tidak seperti kemunculan pertama kali yang dianggap sebagai *read-only* web dan hanya sebagai sistem kognisi. Perkembangan website menjadikannya sebagai sistem *technosocial* yang saling berinteraksi dengan manusia melalui jaringan teknologi sehingga meningkatkan kognisi, komunikasi, serta kerja sama manusia [1].

Data yang didapat dari *internet live stats* (internetlvestats.com) bahwa jumlah website di seluruh dunia mencapai 1.879.976.800 [2]. Selama tahun 2016 sampai 2017, jumlah website telah tumbuh secara signifikan, dari 900 juta pada Januari 2016 menjadi 1.7 miliar pada 2017. Melihat pertumbuhan website yang sangat signifikan, maka website telah menjadi kebutuhan setiap orang baik untuk kebutuhan pribadi maupun bisnis.

Salah satu permasalahan yang sering terjadi di tengah masyarakat adalah sulitnya menemukan media penghubung antara satu pihak dengan pihak yang lain. Permasalahan menemukan penghubung antara satu pihak dengan pihak lainnya dapat diselesaikan melalui website. Berbagai website penghubung antara satu pihak dengan pihak yang lain mulai muncul seperti kitabisa (kitabisa.com) yang menghubungkan donatur dengan pemilik *campaign*, indorelawan (indorelawan.com) yang menghubungkan organisasi dengan relawan, dan lainnya.

Faktanya, dalam penyebaran informasi *event* keagamaan sebagai sarana penghubung masih belum merata. Hal ini disebabkan karena penyebaran informasi masih disebarkan dalam bentuk teks yang berupa *broadcast* melalui media sosial seperti Whatsapp, Facebook, Telegram, dll. Pada akhirnya muncul permasalahan yang dihadapi mulai penyebaran informasi yang belum merata, penyebaran informasi yang belum terpusat sehingga menyebabkan informasi tidak tersebar dengan baik.

Maka melihat permasalahan tersebut, diperlukan suatu website yang digunakan untuk penyebaran informasi sebagai sarana penghubung antara pembuat *event* keagamaan dan masyarakat, sehingga informasi mengenai *event* keagamaan dapat disebarkan secara terpusat dan masyarakat luas dapat mengetahui informasinya. Oleh karena itu, penulis ingin mengimplementasikan pembuatan website sebagai sarana penghubung antara satu pihak dengan pihak yang lain agar pada akhirnya dapat digunakan dan berguna bagi masyarakat luas.

Adapun judul penelitian yang akan dilakukan adalah **Rancang Bangun Sistem Informasi *Event* Keagamaan Berbasis Web Menggunakan *Framework Laravel*.**

1.2 Perumusan Masalah

Berdasarkan uraian yang telah dijelaskan pada latar belakang, maka akan dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana membangun aplikasi yang menghubungkan dua pihak antara yang membuat dan mencari *event* keagamaan?
2. Apakah fitur-fitur pada aplikasi dapat berjalan secara optimal dan sesuai dengan kebutuhan penyebaran informasi kajian?

1.3 Tujuan dan Manfaat Penelitian

Berdasarkan rumusan masalah yang diuraikan di atas, maka akan didapatkan tujuan sebagai berikut:

1. Membangun sistem informasi *event* keagamaan yang menghubungkan dua pihak dalam membuat dan mencari *event* keagamaan.

Adapun manfaat:

1. Pembuat *event* dapat membuat *event* secara terpusat sehingga event lebih mudah dikelola dan ditemukan melalui platform website.
2. *User* dapat menemukan *Event* keagamaan secara terpusat sehingga memudahkan *user* dalam mencari event.

1.4 Batasan Masalah

Batasan masalah yang dimiliki sebagai berikut:

1. Aplikasi yang dikembangkan berbasis web yang menggunakan *Framework Laravel*.
2. Website yang dibangun hanya berfokus pada fitur: melihat event, join event, mengelola event, ticket, mengelola akun, login dan register ,dashboard user.
3. Website yang dirancang belum menggunakan *web service* baik dari sisi *server* maupun dari sisi *client*.
4. Website memiliki *constraint* lokasi di mana website tidak menampilkan *event* berdasarkan lokasi user saat ini.
5. Dari sisi website tidak memiliki verifikasi terhadap *event* yang dibuat oleh *user* sehingga *user* dapat membuat *event* selain *event* keagamaan.

1.5 Sistematika Penulisan

Penyusunan tugas akhir memiliki sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang yang mendasari tugas akhir, perumusan masalah yang bersifat pertanyaan dari latar belakang, tujuan dan manfaat yang dihasilkan, serta batasan masalah yang bertujuan untuk tetap berada dalam objek tugas akhir.

2. BAB II LANDASAN TEORI

Bab ini menjelaskan teori-teori dan penjelasan yang berkaitan dengan tugas akhir, serta memaparkan tugas akhir terkait yang pernah dilakukan sebelumnya sebagai referensi.

3. BAB III METODOLOGI PENELITIAN

Bab ini memaparkan bagaimana tugas akhir dilakukan menggunakan metode dan teknik penelitian, tahapan apa saja yang dilalui, lingkungan pengembangan sistem, serta jadwal rinci pelaksanaan tugas akhir.

4. BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas analisis dari kebutuhan sistem yang akan dirancang kemudian setelah mendapatkan hasil analisis dilanjutkan dengan merancang sistem yang akan dikembangkan.

5. BAB V IMPLEMENTASI DAN EVALUASI

Bab ini membahas hasil implementasi dan evaluasi dari sistem aplikasi *event* keagamaan yang telah dikembangkan.

6. BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari pelaksanaan tugas akhir serta saran yang dapat dilakukan untuk penelitian selanjutnya.

BAB II

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

2.1.1 Teori Sistem Informasi

Menurut Laudon, sistem informasi adalah komponen-komponen yang saling berkaitan yang saling bekerja-sama untuk mengumpulkan, mengolah, menyimpan, dan menampilkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengaturan, analisis, dan visualisasi pada sebuah organisasi [3].

Suatu sistem informasi tidak hanya terbentuk dari hal-hal yang bersifat fisik, seperti komputer, server, namun juga mencakup hal yang tidak terlihat secara fisik. Sistem informasi memiliki komponen utama berupa [4]:

1. Data
2. Perangkat lunak (*software*)
3. Perangkat keras (*hardware*)
4. Perangkat jaringan (*network*)
5. Orang (*brainware*)

2.1.2 Web Framework

Framework dalam konteks pengembangan website adalah seperangkat kode atau *library* yang menyediakan fungsionalitas umum untuk seluruh *class-class* aplikasi. *Framework* bisa dilihat sebagai dasar atau kerangka untuk membangun suatu website. Dalam kaitannya dengan *library*, *framework* memiliki fungsionalitas yang lebih luas. *Libraries* biasanya berfokus pada pemecahan atau penyelesaian suatu masalah. Sedangkan *framework* menyediakan struktur dari suatu aplikasi. Tujuan menggunakan *framework* pada dasarnya untuk

mempercepat pengembangan website tanpa harus menulis ulang fitur dan struktur website dari awal. Dari pada harus membangun aplikasi web dari awal, lebih baik *developers* menggunakan *framework* yang telah memiliki banyak fungsi, fitur, dll, yang sudah diuji dan berjalan dengan benar. Kebanyakan *framework* hari ini bebas digunakan oleh siapapun, dan juga bersifat *open source* sehingga siapa saja dapat bergabung untuk berkontribusi [5].

2.1.3 MVC

Framework aplikasi web biasanya mengimplementasi pola MVC. Pola MVC memecah suatu aplikasi *framework* menjadi 3 modul: *model*, *view*, dan *controller*. Modul *model* adalah bisnis *logic* dari aplikasi dan inti dari aplikasi. *View* adalah *user interface* dari *controller*, yang mengatur tampilan untuk *user*. *Controller* mengimplementasi alur kontrol antara *view* dan *model* [6].

Model-View-Controller (MVC) adalah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk mengenkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk dipresentasikan pada sebuah *user interface*. *Framework* aplikasi web terdiri dari 3 komponen yang dipisah berdasarkan [7]:

1. *Model*

Digunakan untuk mengelola dan menyimpan informasi, serta memberitahu *user* ketika ada perubahan informasi. Model mengandung data dan fungsi yang berkaitan dengan pemrosesan data.

2. *View*

Bertanggung jawab untuk menampilkan informasi kepada *user* melalui sebuah platform (web, android). *View* melekat pada *model* dan me-render isinya ke permukaan layar untuk ditampilkan ke *user*.

3. *Controller*

Bertugas menerima input dari *user* dan menginstruksikan *model* dan *view* untuk memproses aksi berdasarkan inputan tersebut. Sehingga *Controller* bertanggung jawab atas aksi yang dilakukan oleh *user* terhadap aplikasi.

2.1.4 Laravel

Laravel adalah *Framework MVC* yang memiliki fitur *bundles*, *migrations*, dan *Artisan CLI*. Laravel menyediakan sekumpulan *tools* yang lengkap dan menggabungkan arsitektur aplikasi dengan fitur-fitur terbaik dari berbagai *framework* seperti CodeIgniter, Yii, ASP.NET MVC, Ruby on Rails, Sinatra, dan lainnya.

Laravel adalah *framework Open Source*. Memiliki seperangkat fitur yang sangat banyak yang akan mempercepat pengembangan website. Jika sudah terbiasa dengan *Core PHP* dan *Advance PHP*, maka laravel akan lebih mempermudah pengembangan web, sehingga akan menghemat banyak waktu jika ingin mengembangkan situs dari awal. Tidak hanya itu, website yang dibangun dengan Laravel juga aman. Ini mencegah berbagai serangan yang dapat terjadi di website [8].

2.1.5 MySQL

MySQL adalah aplikasi sistem manajemen *database SQL* bersifat *Open Source* yang paling populer dan banyak digunakan. MySQL dikembangkan, didistribusikan, dan didukung oleh Oracle Corporation.

MySQL merupakan *database relasional* yang menyimpan data dalam tabel yang terpisah. Struktur *database* dikelola ke dalam *file* fisik yang dioptimalkan untuk kecepatan. MySQL menawarkan pemrograman yang fleksibel yang berkaitan dengan objek seperti *database*, *table*, *view*, *rows*, dan *columns*.

MySQL bersifat *Open Source* sehingga setiap orang memungkinkan untuk menggunakan dan memodifikasi MySQL. Siapapun boleh mengunduh MySQL di internet dan menggunakannya tanpa harus membayar apapun. MySQL menyediakan *source code* untuk dipelajari dan bisa diubah sesuai kebutuhan.

MySQL berjalan pada *client* dan *server*. *Server* MySQL sangat cepat, handal, skalabilitas, dan mudah digunakan sehingga bisa meng-*handle database* yang besar dengan lebih cepat. MySQL Server bisa berjalan di desktop atau laptop, web server, dan yang lainnya [9].

2.1.6 Waterfall

Waterfall merupakan metode klasik dalam pengembangan perangkat lunak. Waterfall bersifat sistematis, berurutan dalam membangun *software*. Dalam pengembangan menggunakan metode waterfall, terdapat beberapa tahapan yang berurutan yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *coding* (pengkodean) & *testing* (pengujian), penerapan program, dan pemeliharaan.

2.1.7 UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri bahkan dalam pengembang *software* untuk visualisasi, merancang, dan mendokumentasikan sistem *software*. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML dapat membuat model untuk semua jenis aplikasi *software*, di mana aplikasi tersebut dapat berjalan di *hardware*, sistem operasi, jaringan. UML menggunakan konsep *class* dan *operation* dalam konsep dasarnya sehingga UML lebih cocok untuk bahasa pemrograman berorientasi objek (*Object Oriented Programming*). Walaupun begitu, UML masih bisa digunakan untuk *modelling* bahasa pemrograman berbasis prosedural.

Dalam *modelling* suatu *software*, maka dibutuhkan diagram yang menggambarkan rancangan dari suatu *software*. UML telah mendefinisikan diagram-diagram yang dibutuhkan sebagai berikut [10]:

1. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem. *Use Case Diagram* menjelaskan apa yang dilakukan oleh sistem. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dan sistem. *Use Case Diagram* membantu dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client*, dan merancang *test case* untuk semua fitur yang ada pada sistem.

2. *Class Diagram*

Class Diagram menggambarkan atribut atau properti dari suatu sistem, sekaligus menyediakan layanan untuk memanipulasi fungsi atau metode tersebut. *Class Diagram* menjelaskan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class* jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

3. *State Diagram*

State Diagram menggambarkan transisi dan perubahan keadaan suatu objek pada sistem sebagai akibat dari interaksi yang diterima. Pada umumnya *State Diagram* menggambarkan *class* tertentu dan satu *class* dapat memiliki lebih dari satu *State Diagram*.

4. *Activity Diagram*

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana sistem berakhir. *Activity Diagram*

juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* menjelaskan proses-proses dan jalur-jalur aktivitas dari level secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

5. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

6. *Collaboration Diagram*

Collaboration Diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi *Collaboration Diagram* menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*,

7. *Component Diagram*

Component Diagram menggambarkan struktur dan hubungan antar komponen *software*, termasuk hal yang berkaitan dengan *software* seperti *dependency* di antaranya. Komponen *Software* adalah modul yang berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*.

8. *Deployment Diagram*

Deployment Diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak, bagaimana infrastruktur jaringan pada komponen tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat *fisikal*.

2.1.8 Black Box Testing

Pengujian *Software* sangat diperlukan untuk memastikan *software* yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian *software* merupakan elemen kritis dari jaminan kualitas *software* dan merupakan bagian yang tak terpisahkan dari siklus hidup *software* yang lain seperti analisis, desain, dan pengkodean [11].

Salah satu jenis pengujian *software* menggunakan metode Black Box Testing. Black Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional *software*, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program [12].

Cara kerja teknik Black Box Testing terdiri dari 3 langkah:

1. Input

Persyaratan dan spesifikasi fungsional dari sistem akan diperiksa. Dokumen desain dan *source code* aplikasi juga diperiksa.

2. *Processing Unit*

Pada proses ini akan dilakukan uji kasus dengan inputan dan juga menjalankannya. Tahap ini juga akan dilakukan *load testing*, *stress testing*, *security review*, dan *globalization testing*. Jika terdapat kekurangan maka akan diperbaiki dan diuji ulang.

3. Output

Setelah melalui semua tahap pengujian, maka akan muncul output yang diharapkan dan menyiapkan laporan akhir dari pengujian.

2.1.9 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Dengan skala likert, variabel yang akan diukur dijabarkan menjadi indikator variabel.

Dalam pengukuran skala likert, terdapat dua bentuk pertanyaan yaitu [13]

1. Pertanyaan Positif

Digunakan untuk mengukur skala positif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1

2. Pertanyaan Negatif

Digunakan untuk mengukur skala negatif. Pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5.

Bentuk jawaban Skala Likert antara lain: Sangat Setuju, Setuju, Ragu-ragu, Tidak Setuju, Sangat Tidak Setuju.

2.2 Penelitian Terkait

Penelitian terkait membahas tentang penelitian-penelitian terdahulu yang memiliki kaitan yang sama dengan penelitian penulis.

1. Pada penelitian Pembuatan “Sistem Informasi Registrasi *Event* dan *ticket event Lagirame*” disimpulkan banyak kegiatan atau event yang dilaksanakan di Kota Surakarta maupun di seluruh Indonesia setiap bulannya. Misal konser musik yang mengajak seluruh penikmat musik, atau event book fair yang mengajak para pembaca. Melihat *event* atau kegiatan tersebut masih ditemukan informasinya tidak terpusat, maka perlu dirancang sebuah sistem informasi manajemen data *event, Lagirame* [14].
2. Pada penelitian “Pengembangan Sistem Informasi *Event* Mahasiswa Berbasis Web di Universitas Negeri Yogyakarta” disimpulkan mahasiswa sering mengadakan berbagai *event* di kampus, namun *event* yang diadakan masih disebarakan tidak terpusat sehingga tidak seluruh mahasiswa mengetahui *event* tersebut maka dibutuhkan sistem informasi kegiatan mahasiswa untuk membantu penyelenggara *event* menyampaikan informasi *event* kepada mahasiswa menggunakan aplikasi web yang dikembangkan menggunakan *framework* Laravel [15].
3. Pada penelitian “Sistem Informasi Kajian dan dan *Event* Kerohanian Islam Berbasis Web” disimpulkan bahwa penyampaian informasi berupa jadwal-jadwal kajian atau jadwal *event* yang dilakukan masih menggunakan media publikasi yang kurang efektif yaitu berupa poster yang hanya disebarakan di tempat tertentu. Maka dibutuhkan sarana untuk mempermudah penyampaian informasi kepada masyarakat melalui media berupa website [16].

Table 1: Penelitian Terkait

Nama Peneliti	Judul	Tahun	Tools	Metodologi	Hasil
Obeth Dewa Wardhana	Sistem Informasi Registrasi <i>Event</i> dan <i>ticket event</i> Lagirame	2017	<i>Framework</i> CodeIgniter, MySQL Database	-	Penelitian ini menghasilkan sistem informasi registrasi manajemen data <i>event</i> , Lagirame. Aplikasi ini memiliki tujuan sebagai sarana promosi <i>event</i> dan informasi bagi masyarakat.
Nanang Wisnu Pambudi	Pengembangan Sistem Informasi <i>Event</i> Mahasiswa Berbasis Web di Universitas Negeri Yogyakarta	2016	<i>Framework</i> Laravel, MySQL Database	Model pengembangan perangkat lunak menggunakan waterfall, terdiri dari 4 tahapan yaitu analisis kebutuhan, desain, implementasi, dan pengujian	Penelitian ini menghasilkan Sistem Informasi <i>Event</i> Mahasiswa di Universitas Negeri Yogyakarta. Aplikasi ini memiliki fungsi untuk membantu penyelenggara <i>event</i> menyampaikan informasi event kepada mahasiswa. Pengembangan aplikasi menggunakan model waterfall dan dikembangkan dengan <i>framework</i> Laravel.
Khoirunnisa Septiani	Sistem Informasi Kajian dan dan <i>Event</i> Kerohanian Islam Berbasis Web	2016	<i>Framework</i> CodeIgniter, <i>Framework</i> JQuery, MySQL Database	Metode yang digunakan dalam penelitian yaitu Observasi, Studi Literatur, Pengembangan Sistem, Implementasi Sistem, Pengujian Sistem, Dokumentasi	Penelitian ini menghasilkan sistem informasi yang mempermudah penyampaian dan penyebaran informasi dari penyelenggara kajian atau <i>event</i> kepada pengunjung, sehingga mempermudah pengunjung mencari <i>event</i> di Yogyakarta secara terstruktur dan akurat.

2.3 Aplikasi Terkait

Pada aplikasi terkait membahas aplikasi-aplikasi web yang berkaitan dengan penelitian yang dilakukan penulis. Aplikasi terkait bertujuan sebagai bahan perbandingan dalam membangun dan merancang aplikasi serupa namun dengan mengembangkan fitur-fitur yang belum ada pada aplikasi terkait atau menyesuaikan kebutuhan user.

1. Eventbrite (eventbrite.com)

Eventbrite merupakan salah satu platform website yang memungkinkan setiap orang untuk membuat *event* dan menyebarkannya kepada orang lain. Melalui eventbrite banyak kegiatan atau *event* diselenggarakan, mulai dari *event* music, konferensi, perlombaan, *hackathon*, diskusi, *talkshow*, penggalangan dana, kompetisi *game*, dan lain-lain. Melalui eventbrite, setiap orang dapat menemukan informasi *event* atau kegiatan secara terpusat, ditambah dengan adanya fitur *ticket event* dan *online registration* sehingga memudahkan *user* dalam mengikuti berbagai *event*.

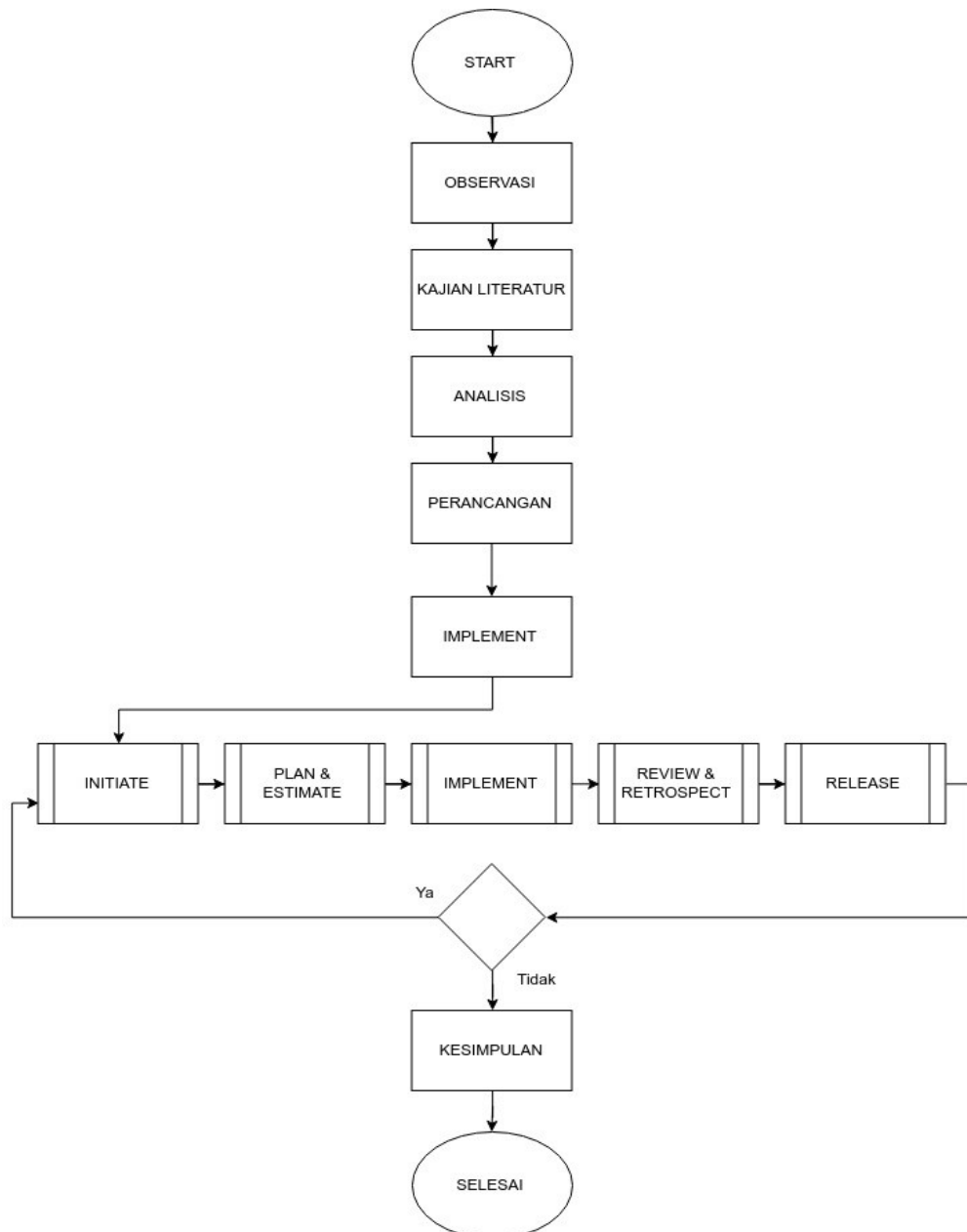
2. Jadwalkajian (Jadwalkajian.com)

Jadwalkajian merupakan platform website yang berfokus pada *event* atau kegiatan keagamaan. Tujuan dari website jadwalkajian agar kaum muslimin mudah mencari kajian sunnah di berbagai daerah indonesia. Pada website ini setiap kajian menampilkan informasi berupa judul kajian, pemateri, informasi, waktu, serta lokasi kajian.

BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian



Gambar 1: Tahapan Penelitian

1. Observasi

Pada tahapan ini dilakukan observasi di internet dalam melihat berbagai website yang digunakan untuk menghubungkan kedua belah pihak. Awalnya pengamatan dilakukan dengan melihat website seperti kitabisa, indorelawan, dan dicoding. Di mana website-website tersebut bertujuan untuk menghubungkan kedua belah pihak yang ingin mengikuti suatu kegiatan.

Setelah melihat dan mengumpulkan informasi tersebut, maka dilakukan pengamatan terhadap website yang menghubungkan kedua belah pihak namun dalam ruang lingkup keagamaan, setelah itu dilakukan pengamatan di internet untuk mencari tahu website-website yang menyediakan informasi-informasi kajian.

Hasil yang didapatkan sudah tersedia beberapa website yang menyampaikan informasi-informasi kajian, namun website tersebut hanya sebatas menyampaikan informasi saja namun tidak adanya fitur *join event* dan *ticket event*.

2. Kajian Literatur

Kajian literatur bertujuan untuk memperkuat hasil observasi yang telah didapatkan. Studi literatur dilakukan dengan cara mencari, mengumpulkan, membaca artikel, jurnal, tugas akhir, *ebook*, website yang berkaitan dengan penelitian penulis.

Hasil yang didapatkan dari kajian literatur berupa perancangan dan implementasi penelitian yang telah dilakukan, mengetahui *Minimum Viable Product* (MVP) yang telah dilakukan oleh peneliti sebelumnya, kemudian mengetahui kekurangan yang pada penelitian sebelumnya sehingga penulis bisa melanjutkan penelitian meningkatkan fitur yang belum ada.

3. Analisis

Setelah mendapatkan gambaran secara umum dari penelitian sebelumnya dan juga gambaran aplikasi yang ingin dibuat, maka penulis berusaha menganalisis hal-hal apa saja yang harus ditambahkan dari penelitian sebelumnya.

Pada tahapan analisis bertujuan untuk mengidentifikasi dan mengevaluasi permasalahan dan kendala yang ada sehingga solusi yang ditawarkan akan disesuaikan dengan hasil analisis. Dengan mengetahui berbagai kendala dan kekurangan, diharapkan nantinya produk yang dikembangkan dapat meningkatkan dan memperbaiki kekurangan yang ada pada produk sebelumnya.

4. Perancangan

Pada tahapan ini dilakukan perancangan sistem berdasarkan hasil analisis yang telah dilakukan pada tahap sebelumnya. Perancangan sistem dirancang menggunakan UML sebagai sebuah bahasa yang menawarkan sebuah standar untuk merancang model sebuah sistem.

Perancangan suatu sistem menggunakan UML terdiri dari perancangan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Collaboration Diagram*, *Class Diagram*, *State Diagram*, *Component Diagram*, dan *Deployment Diagram*.

5. Implementasi

Tahapan selanjutnya yang dilakukan adalah Implementasi berdasarkan hasil perancangan yang sudah dibuat. Implementasi dilakukan dengan pengkodean bahasa pemrograman atau *coding* yang bertujuan menghasilkan suatu produk berupa website yang akan digunakan.

Pengkodean dimulai dari merancang *front end* website yang berguna untuk membuat tampilan website, kemudian akan dilakukan pengkodean pada bagian *back end* agar website bisa berjalan sesuai fungsionalitas.

3.2 Rancangan Penelitian

3.2.1 Metode Penelitian

Dalam penelitian yang dilakukan ini penulis menggunakan metode penelitian deskriptif kualitatif. Metode penelitian deskriptif dipilih karena bertujuan untuk mendeskripsikan suatu gejala, peristiwa, kejadian yang terjadi secara faktual, sistematis, dan akurat. Penelitian deskriptif memusatkan perhatian kepada masalah-masalah yang terjadi pada penelitian berlangsung.

3.2.2 Jenis Penelitian

Jenis penelitian yang digunakan adalah Jenis Penelitian Deskriptif. Jenis Penelitian ini merupakan jenis penelitian yang mendeskripsikan suatu gejala, peristiwa, atau kejadian yang sedang terjadi. Output dari penelitian yang dihasilkan adalah berupa pengembangan aplikasi website yang nantinya akan digunakan oleh dua belah pihak dalam menyebarkan dan mencari informasi seputar keagamaan seperti mabit, kajian, *event* keislaman, dll.

3.2.3 Metode Pengumpulan Data

Dalam mengumpulkan data sebagai penunjang Tugas Akhir, penulis melakukan beberapa metode dalam mengumpulkan data dimana proses untuk mendapatkan data melalui observasi dan studi pustaka.

1. Observasi

Tahapan ini dilakukan proses pengamatan cara kerja website-website yang dijelaskan pada bagian aplikasi. Pengamatan dilakukan untuk melihat cara kerja, mengetahui bagaimana alur tahapan website mulai dari awal sampai akhir, serta mengetahui fitur-fitur yang ada pada website.

2. Studi Pustaka

Tahapan ini dilakukan dengan mengumpulkan informasi penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan penulis. Pengumpulan informasi dilakukan dengan mencari, mengumpulkan, membaca artikel, jurnal, *ebook*, maupun skripsi peneliti lainnya untuk mengkaji dan mengenai pengembangan aplikasi berbasis website, khususnya website yang digunakan untuk membangun sistem informasi.

3.2.4 Lingkungan Pengembangan

3.2.4.1 Tempat/Lokasi

Penulis melakukan penelitian pada Semester 6 dalam rangka menyusun Tugas Akhir. Penelitian dilaksanakan di Kampus B STT Terpadu Nurul Fikri, yang beralamat di Jalan Lenteng Agung Raya No.20, Srengseng Sawah, Lenteng Agung – Jakarta Selatan 12640.

3.2.4.2 Bahan dan Alat

Penelitian yang dilakukan untuk membangun suatu produk website membutuhkan bahan dan alat sebagai sarana dalam pengembangan. Adapun bahan dan alat yang digunakan sebagai berikut.

1. Laptop Acer E1-471

1. Processor: Intel Core i3, 2.20GHz
2. RAM: 8GB
3. System Type: 64 bit Operating System

Laptop digunakan sebagai device yang membantu penyusunan tugas akhir.

2. Xubuntu 14.04.5

Sistem Operasi berbasis Linux yang digunakan untuk membantu menyusun penelitian dalam bentuk tugas akhir. Xubuntu merupakan varian dari sistem operasi Ubuntu, namun Xubuntu memiliki perbedaan dengan Ubuntu yang terletak pada *Desktop Environment* yang digunakan. LibreOffice

Salah satu perangkat lunak perkantoran selain Microsoft Office. LibreOffice adalah perangkat lunak yang bersifat *open source* yang digunakan untuk menulis penelitian dalam bentuk tugas akhir.

3. XAMPP

XAMPP adalah kumpulan dari berbagai perangkat lunak *open source* yang digunakan untuk membantu *programmer* dalam menggunakan *web server*, *database*, dan juga *phpmyadmin*.

4. Google Chrome

Web Browser Google Chrome digunakan untuk melihat hasil aplikasi yang sudah dibuat, serta web browser digunakan sebagai *debugging* aplikasi web yang sedang dikembangkan.

5. *Framework* Laravel

Laravel adalah *framework* PHP yang digunakan untuk mengembangkan aplikasi website. *Framework* Laravel menggunakan konsep MVC dalam mengembangkan suatu website.

3.3 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah *Waterfall*. Dalam pengembangan menggunakan metode *waterfall*, terdapat beberapa tahapan yang berurutan yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *coding* (pengkodean) & *testing* (pengujian), penerapan program, dan pemeliharaan.

Tahapan-tahapan metode pengembangan Waterfall sebagai berikut:

1. *Requirement* Analisis

Pada tahapan ini diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh *user* dan batasan perangkat lunak tersebut. Informasi mengenai kebutuhan dan batasan perangkat lunak bisa didapat melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *System Design*

Pada tahapan ini dilakukan desain sistem yang didapatkan dari kebutuhan sebelumnya. Desain sistem membantu dalam menentukan sistem persyaratan dan mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahapan ini, desain sistem yang telah dirancang akan dilakukan untuk pembuatan sistem atau aplikasi.

4. *Integration & Testing*

Sistem yang telah dikembangkan akan masuk pada tahapan testing. Sistem akan diuji untuk mengecek fungsionalitas sudah berjalan sesuai *requirement*.

5. *Operation & Maintenance*

Pada tahap akhir, perangkat lunak yang sudah jadi, dijalankan dan dilakukan pemeliharaan. Pemeliharaan termasuk memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan diuraikan mengenai proses analisis dan perancangan sistem. Proses analisis dan perancangan sistem akan diuraikan dalam bentuk *Use Case Diagram*, *Use Case Description*, *Domain Model*, *Sequence Diagram*, *Collaboration Diagram*, *Class Diagram*, serta tampilan aplikasi yang diuraikan dalam bentuk struktur menu dan *mock-up* aplikasi web.

4.1 Analisis Sistem

Pada penelitian ini dilakukan proses pengumpulan *User Requirement* melalui studi pustaka dari penelitian sebelumnya yang dilakukan oleh Khoirunnisa Septiani dengan judul “Sistem Informasi Kajian dan dan *Event* Kerohanian Islam Berbasis Web”.

Pengumpulan *User Requirement* juga dilakukan melalui proses pengamatan dan perbandingan yang dilakukan pada website Event Brite (eventrbite.com) dan Jadwal Kajian (jadwalkajian.com), Kedua website tersebut memberikan informasi seputar *event* namun dengan tujuan dan fitur yang berbeda.

Berikut merupakan hasil atau poin-poin utama yang didapatkan dari penelitian sebelumnya yang dilakukan oleh Khoirunnisa Septiani dengan judul “Sistem Informasi Kajian dan dan *Event* Kerohanian Islam Berbasis Web” yaitu:

Table 2: Penelitian Sebelumnya

No	Fitur
1	Menyajikan jadwal <i>event</i> atau kajian Selama 1 Minggu
2	Mengelola jadwal <i>event</i> atau kajian
3	Mengelola data pengelola
4	Informasi berita islam dan jadwal salat

Selanjutnya merupakan hasil atau poin-poin utama yang didapatkan dari hasil pengamatan dan perbandingan pada website Eventbrite dan Jadwalkajian.

Table 3: Perbandingan Website

No	Perbandingan	Event Brite	Jadwal Kajian
1	Segmentasi	Umum	Khusus
2	<i>Join event</i> atau kajian	Ya	Tidak
3	Membuat <i>event</i> atau kajian	Ya	Tidak
4	Pengelola <i>event</i> atau kajian	Personal	Pemilik

Melihat kebutuhan *User Requirement* yang didapatkan dari 2 tahapan di atas yaitu pada penelitian sebelumnya dan perbandingan website *event* maka didapatkan hasil analisis sistem sebagai berikut:

1. Hasil analisis sistem dari perbandingan website.

Table 4: Hasil Perbandingan Website

No	Parameter	Hasil
1	Segmentasi	Agama
2	<i>Join Event</i>	Ya
3	Mengelola <i>Event</i>	Ya
4	Pengelola <i>Event</i>	Personal

2. Hasil analisis sistem dari penelitian sebelumnya.

Table 5: Hasil Analisa Sistem

No	Perbandingan	Penelitian Sebelumnya	Penelitian Sekarang
1	Menyajikan jadwal <i>event</i>	Hanya 1 Minggu	Tidak mengacu pada waktu
2	Mengelola jadwal <i>event</i>	Pada daerah Yogyakarta	Tidak terbatas pada tempat
3	Mengelola data <i>event</i>	Pengelola	Personal
4	Tiket <i>event</i>	Tidak	Ya
5	<i>Join event</i>	Ya	Ya
6	<i>Join event: Autentikasi Login</i>	Tidak	Ya
7	Pencarian <i>event</i>	Tidak	Ya
8	Pencarian kategori <i>event</i>	Tidak	Ya

4.1.1 Istilah Baku

Pada aplikasi web yang akan dirancang terdapat beberapa istilah yang akan dipakai di dalam web, maka oleh karena itu akan dibuatkan istilah baku yang dijadikan acuan dari istilah-istilah yang akan digunakan pada pengembangan aplikasi web.

Istilah-istilah baku yang digunakan adalah sebagai berikut

Table 6: Istilah Baku

No	Istilah	Arti
1	<i>Event</i>	Kegiatan atau acara yang muncul di aplikasi web
2	<i>Register</i>	Membuat atau mendaftarkan akun di aplikasi web
3	<i>Login</i>	Masuk ke aplikasi web menggunakan akun yang sudah dibuat pada tahapan <i>Register</i>
4	<i>Join Event</i>	Bergabung pada suatu <i>event</i> yang diharuskan login terlebih dahulu
5	<i>Ticket</i>	Tiket yang didapatkan oleh <i>user</i> setelah bergabung pada suatu <i>event</i>

4.1.2 User Requirement

User Requirement berisi pernyataan-pernyataan layanan yang disediakan oleh sistem dan batasan-batasan operasional.

1. Kategori Kebutuhan User

Table 7: Kebutuhan User

Kode	Deskripsi	User
R.001	Melihat <i>Event</i>	<i>Guest</i>
R.002	Mendaftar sebagai <i>User</i>	<i>Guest</i>
R.003	Masuk sebagai <i>User</i>	<i>Guest</i>
R.004	<i>Join Event</i>	<i>User</i>
R.005	Menampilkan Tiket <i>Event</i>	<i>User</i>
R.006	Mengelola <i>Event</i>	<i>User</i>
R.007	Mengelola Akun	<i>User</i>

Pada tabel Kategori Kebutuhan *User*, terdapat 2 kategori user yaitu *Guest* dan *User*. *Guest* dapat melihat *event*, mencari *event*, mendaftar dan login. *User* dengan kategori user yang telah *login* dapat melihat *event*, *join event*, mengelola *event*, mendownload *ticket*, mengelola akun.

2. Kategori User Aplikasi

Table 8: User Aplikasi

User	Deskripsi
<i>Guest</i>	<i>User</i> yang belum <i>login</i> ke Sistem
<i>User</i>	<i>User</i> yang sudah <i>login</i> dapat mengelola <i>event</i>

Pada tabel Kategori *User Aplikasi* dijelaskan tentang kategori *user* yang memiliki hak akses yang berbeda-beda. Kategori *user* dibedakan menjadi 2 *user*: *Guest* dan *User*.

3. Kategori Menu Aplikasi

Pengelompokan menu dapat dibedakan berdasarkan kategori *user* yang dapat dilihat pada tabel berikut.

Table 9: Menu Aplikasi

<i>User</i>	Menu
<i>Guest</i>	<ul style="list-style-type: none">• <i>Melihat Event</i>• <i>Register</i>• <i>Login</i>
<i>User</i>	<ol style="list-style-type: none">1. <i>Login dan Logout</i>2. <i>Dashboard User</i>3. <i>Mengelola Event</i><ul style="list-style-type: none">• <i>Membuat Event</i>• <i>Mengedit Event</i>4. <i>Join Event</i>5. <i>Tiket</i><ul style="list-style-type: none">• <i>My Ticket</i>6. <i>Account</i><ul style="list-style-type: none">• <i>My Account</i>• <i>My Password</i>

User Guest hanya memiliki menu melihat *event*, *register* dan *login*. *User* kategori *user* yang telah *login* memiliki menu untuk *login* dan *logout*, *dashboard user*, *join event*, *mengelola event*, *tiket*, dan *account*.

2. Use Case Description

Berikut daftar *use case description* dari *use case diagram* di atas

1. Use Case Melihat Event

Table 10: Melihat Event

Nama Use Case	Use Case Scenario UC-001 Melihat Event	
Author	Aufa	
Requirement Terkait	R.001 Melihat Event	
Tujuan	User Guest dapat melihat event	
Kondisi Awal	User Guest membuka website	
Kondisi Sukses	User Guest dapat melihat event	
Kondisi Gagal	User Guest tidak dapat melihat event	
Primary Actor	Guest	
Secondary Actor	-	
Trigger	Membuka website dan tombol cari event	
Main Flow	Step	Action
	1	User Membuka Website atau mengklik tombol cari event
	2	User melihat daftar
Extension		

2. Use Case Mendaftar sebagai User

Table 11: Mendaftar Sebagai User

Nama Use Case	Use Case Scenario UC-002 Mendaftar sebagai User	
Author	Aufa	
Requirement Terkait	R.002 Mendaftar sebagai User	
Tujuan	User Guest dapat mendaftar dan mendapatkan akun.	
Kondisi Awal	User Guest membuka halaman Register	
Kondisi Sukses	User Guest dapat mendaftar	
Kondisi Gagal	User Guest tidak terdaftar	
Primary Actor	Guest	
Secondary Actor	-	

<i>Trigger</i>	<i>User Guest</i> mengklik tombol Register	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	<i>User Guest</i> mengklik tombol Register
	2	<i>User Guest</i> mengisi form pendaftaran
	3	Data guest user disimpan
<i>Extension</i>		

3. Use Case Login sebagai User

Table 12: Login sebagai User

Nama Use Case	Use Case Scenario UC-003 Login sebagai User	
Author	Aufa	
Requirement Terkait	R.003 Login sebagai User	
Tujuan	<i>User Guest</i> dapat login sebagai user	
Kondisi Awal	<i>User Guest</i> membuka halaman Login	
Kondisi Sukses	<i>User Guest</i> dapat login	
Kondisi Gagal	<i>User Guest</i> tidak dapat login	
Primary Actor	Guest	
Secondary Actor	-	
Trigger	<i>User Guest</i> mengklik tombol login	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	<i>User Guest</i> mengklik tombol login
	2	<i>User Guest</i> mengisi username dan password
	3	<i>User Guest</i> dapat login
<i>Extension</i>	<i>Step</i>	<i>Action</i>
	1	<i>User Guest</i> salah menginput username dan password
	2	Menampilkan pesan data salah

4. Use Case Join Event

Table 13: Join Event

Nama Use Case	Use Case Scenario UC-004 Join Event	
Author	Aufa	
Requirement Terkait	R.004 Join Event	
Tujuan	User dapat join event	
Kondisi Awal	User telah berhasil login ke sistem	
Kondisi Sukses	User berhasil join event	
Kondisi Gagal	User gagal join event	
Primary Actor	User	
Secondary Actor	-	
Trigger	User mengklik tombol Register Event	
Main Flow	Step	Action
	1	User Guest mengklik tombol Register Event
	2	User diarahkan untuk login jika belum login ke dalam sistem
	3	User berhasil join Event
Extension	Step	Action
	1	User join event namun belum login
	2	User diarahkan ke halaman login
	3	Menampilkan pesan harus login terlebih dahulu.

5. Use Case Menampilkan Tiket Event

Table 14: Menampilkan Tiket Event

Nama Use Case	Use Case Scenario UC-005 Menampilkan Tiket Event	
Author	Aufa	
Requirement Terkait	R.005 Menampilkan Tiket Event	
Tujuan	User berhasil mendapatkan tiket event	
Kondisi Awal	User telah berhasil join event	
Kondisi Sukses	User berhasil mendapatkan tiket event	
Kondisi Gagal	User gagal mendapatkan tiket event	
Primary Actor	User	
Secondary Actor	-	

<i>Trigger</i>	<i>User mengklik tombol Register Event</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	<i>User Guest mengklik tombol Register Event</i>
	2	<i>User berhasil join event</i>
	3	<i>User mendapatkan tiket event</i>

6. Use Case Mengelola Event

Table 15: Mengelola Event

<i>Nama Use Case</i>	<i>Use Case Scenario UC-006 Mengelola Event</i>	
<i>Author</i>	Aufa	
<i>Requirement Terkait</i>	R.006 Mengelola Event	
<i>Tujuan</i>	<i>User dapat mengelola Event</i>	
<i>Kondisi Awal</i>	<i>User telah berhasil login ke sistem</i>	
<i>Kondisi Sukses</i>	<i>User berhasil mengelola event</i>	
<i>Kondisi Gagal</i>	<i>User gagal mengelola event</i>	
<i>Primary Actor</i>	<i>User</i>	
<i>Secondary Actor</i>	-	
<i>Trigger</i>	<i>User mengklik menu Event</i>	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1	<i>User Guest mengklik menu Event</i>
	2	<i>User mengisi form pembuatan event</i>
	3	<i>Menampilkan event yang sudah dibuat</i>
<i>Extension</i>	<i>Step</i>	<i>Action</i>
	1	<i>User salah dalam mengisi form pembuatan event</i>
	2	<i>Menampilkan pesan kesalahan</i>

7. Use Case Mengelola Akun

Table 16: Mengelola Akun

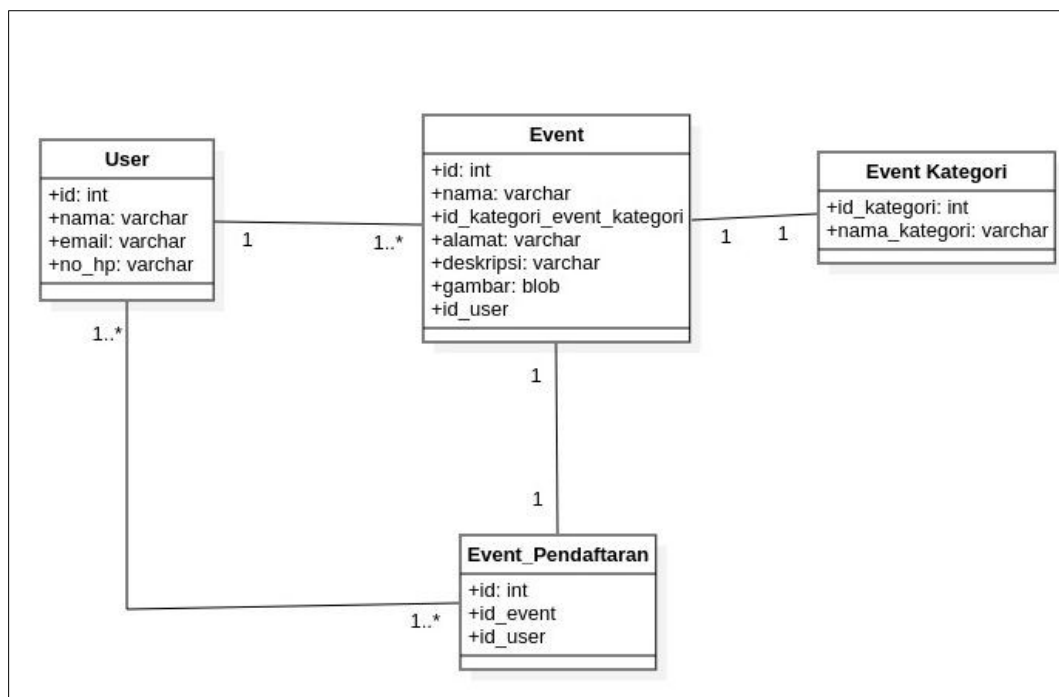
Nama Use Case	Use Case Scenario UC-007 Mengelola Akun	
Author	Aufa	
Requirement Terkait	R.007 Mengelola Akun	
Tujuan	User dapat mengelola akun	
Kondisi Awal	User telah berhasil login ke sistem	
Kondisi Sukses	User berhasil mengelola akun	
Kondisi Gagal	User gagal mengelola akun	
Primary Actor	User	
Secondary Actor	-	
Trigger	User mengklik menu Akun	
Main Flow	Step	Action
	1	User Guest mengklik menu Akun
	2	Menampilkan akun user

4.2 Perancangan Sistem

4.2.1 Desain Sistem

1. Domain Model

Domain model berfungsi untuk menunjukkan hubungan yang mungkin ada di antara objek-objek yang sangat kompleks, secara umum objek diagram berisi dengan objek dan *link*.



Gambar 3: Domain Model

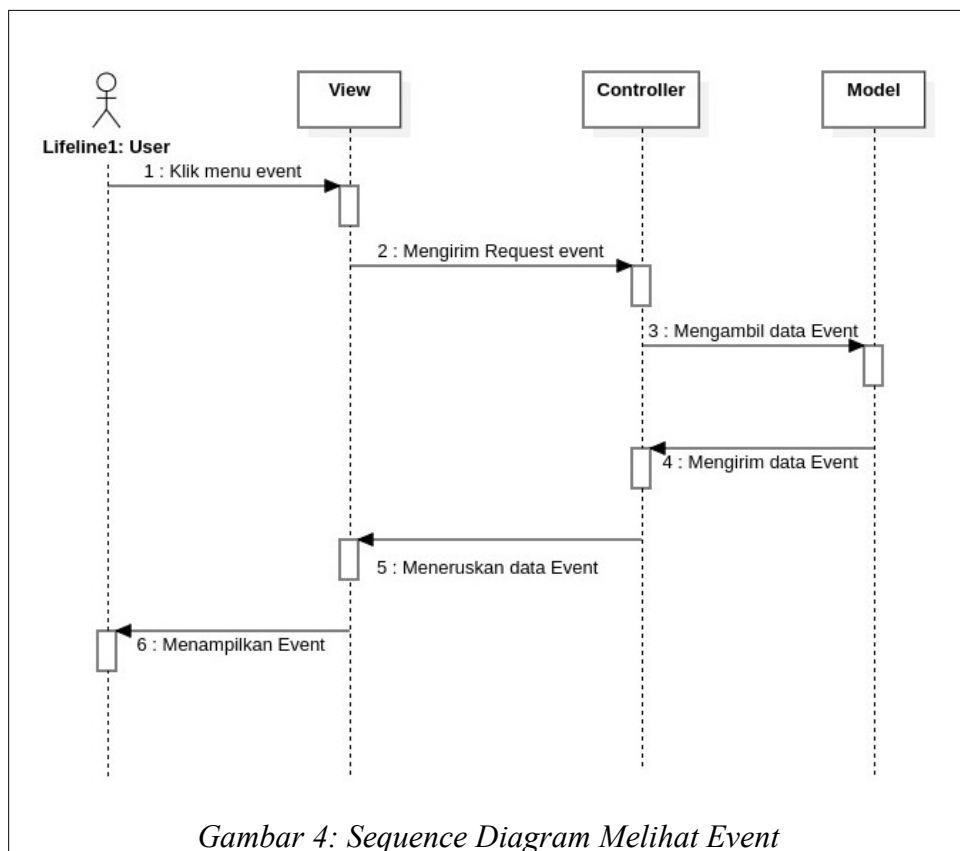
2. *Sequence System Diagram*

System Sequence Diagram digunakan untuk menggambarkan perilaku pada skenario yang telah dibuat. Pada *sequence diagram* ini memiliki komponen utama yang terdiri atas objek yang digambarkan dengan kotak persegi yang bernama, *message* (pesan) yang digambarkan dengan garis tanda panah, serta waktu yang ditunjukkan dengan garis terputus vertikal.

Skenario yang sebelumnya telah dibuat pada *use case scenario*, lalu dikembangkan lagi ke dalam bentuk *sequence diagram*. Jika pada *use case scenario* tidak dideskripsikan objek-objek apa saja yang terlibat dalam sistem, maka pada *sequence diagram* bisa terlihat objek-objek apa saja yang terlibat dalam skenario yang telah dibuat.

a) ***Sequence Diagram Melihat Event***

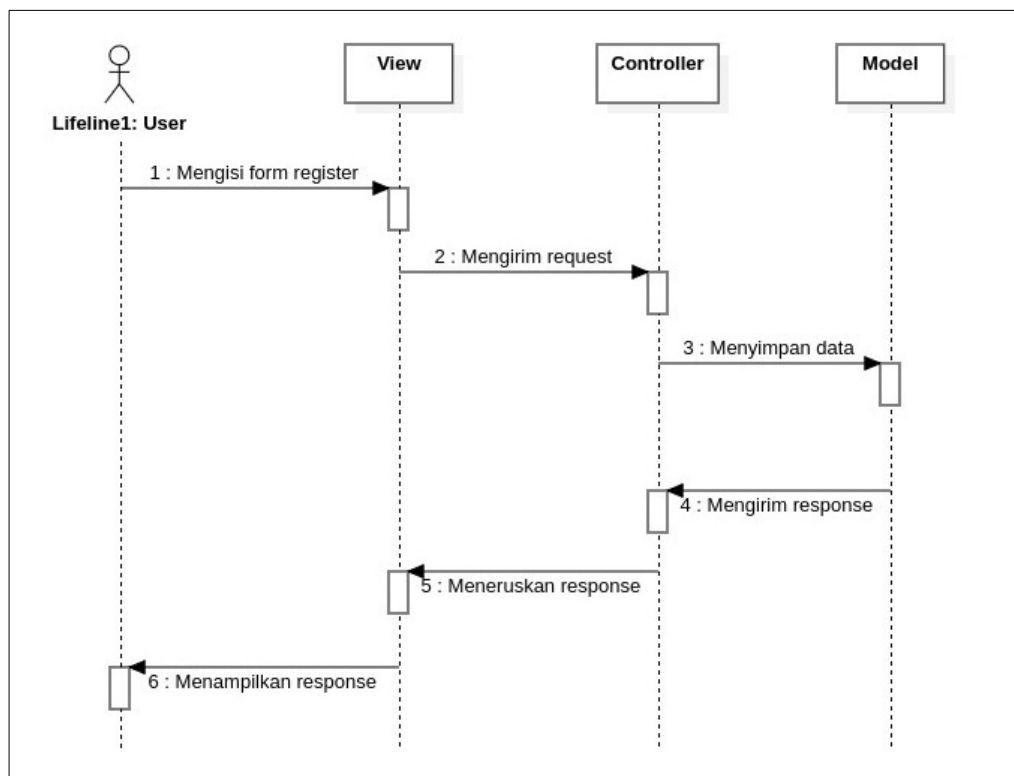
Pada *Sequence Diagram Melihat Event* dijelaskan bahwa untuk melihat *event*, *User* terlebih dahulu mengakses menu *event*, lalu menu tersebut mengirim *request* ke *Controller*. *Controller* akan mengambil data *event* dari Model lalu merender halaman *View* yang ditampilkan ke *User*.



Gambar 4: Sequence Diagram Melihat Event

b) Sequence Diagram Register

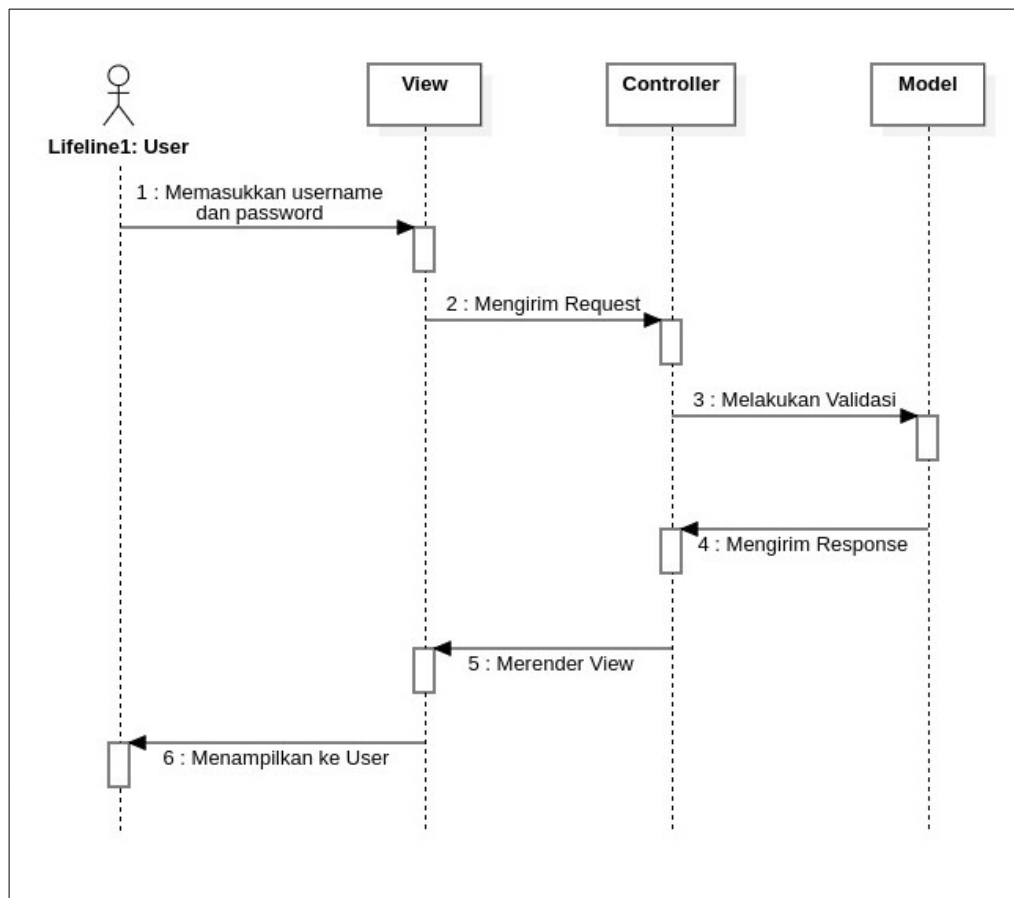
Pada *Sequence Diagram Register* dijelaskan bahwa *User* dapat melakukan Registrasi melalui menu Register. *User* terlebih dahulu mengakses menu Register, lalu mengisi data di halaman Register dan dikirim ke *Controller*. *Controller* akan meneruskan data *user* ke Model untuk disimpan di *Database*.



Gambar 5: Sequence Diagram Register

c) *Sequence Diagram Login sebagai User*

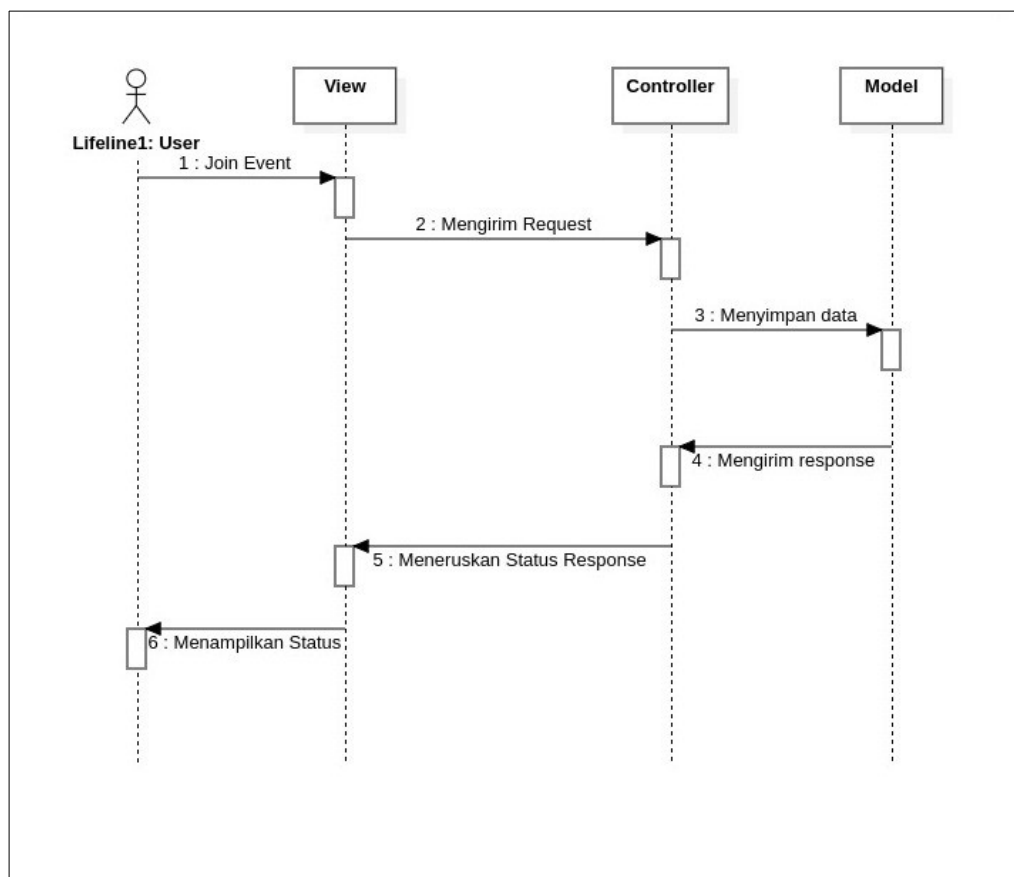
Pada *Sequence Diagram Login* dijelaskan bahwa *User* dapat melakukan *Login*. *User* terlebih dahulu mengakses menu *Login*, lalu mengisi *username* dan *password* dan dikirim ke *Controller*. *Controller* akan meneruskan data *user* ke *Model* untuk memverifikasi data *user*.



Gambar 6: *Sequence Diagram Login*

d) Sequence Diagram Join Event

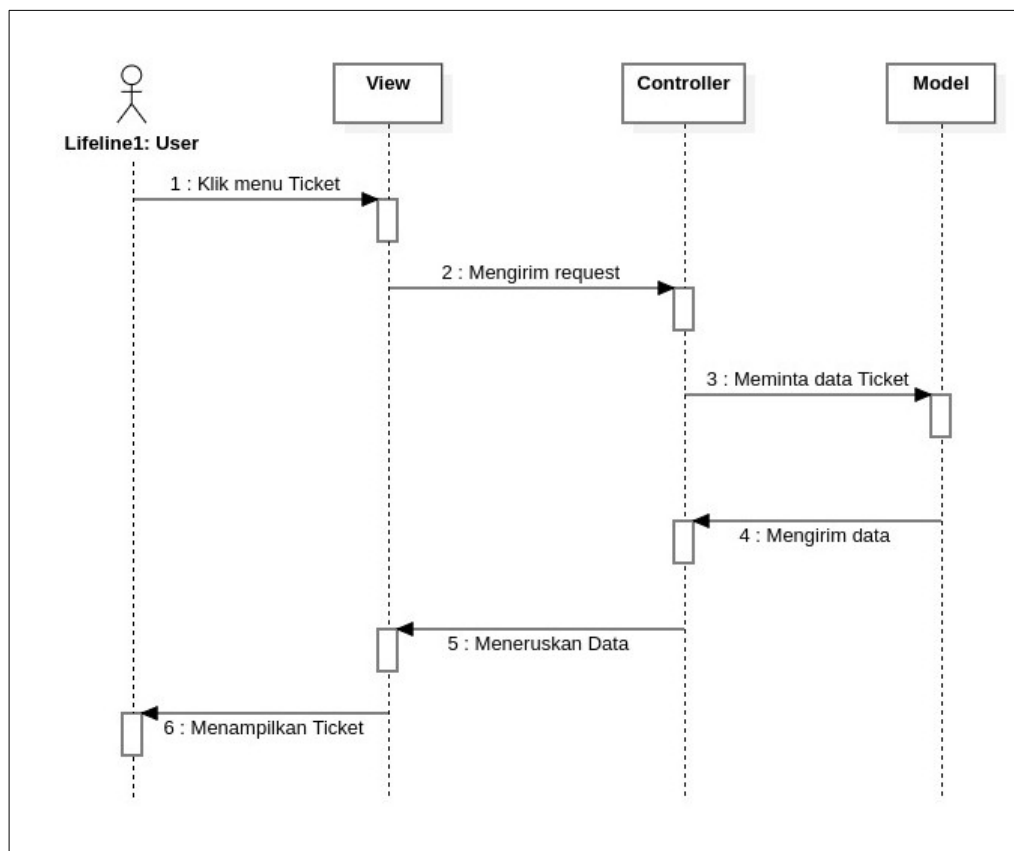
Pada *Sequence Diagram join Event* dijelaskan bahwa *User* dapat *join event*. *User* mengakses *event* terlebih dahulu dan *join* pada *event* tersebut. Data dikirim ke *Controller* yang selanjutnya diteruskan ke *Model* untuk disimpan di *Database*.



Gambar 7: Sequence Diagram Join Event

e) ***Sequence Diagram Menampilkan Tiket***

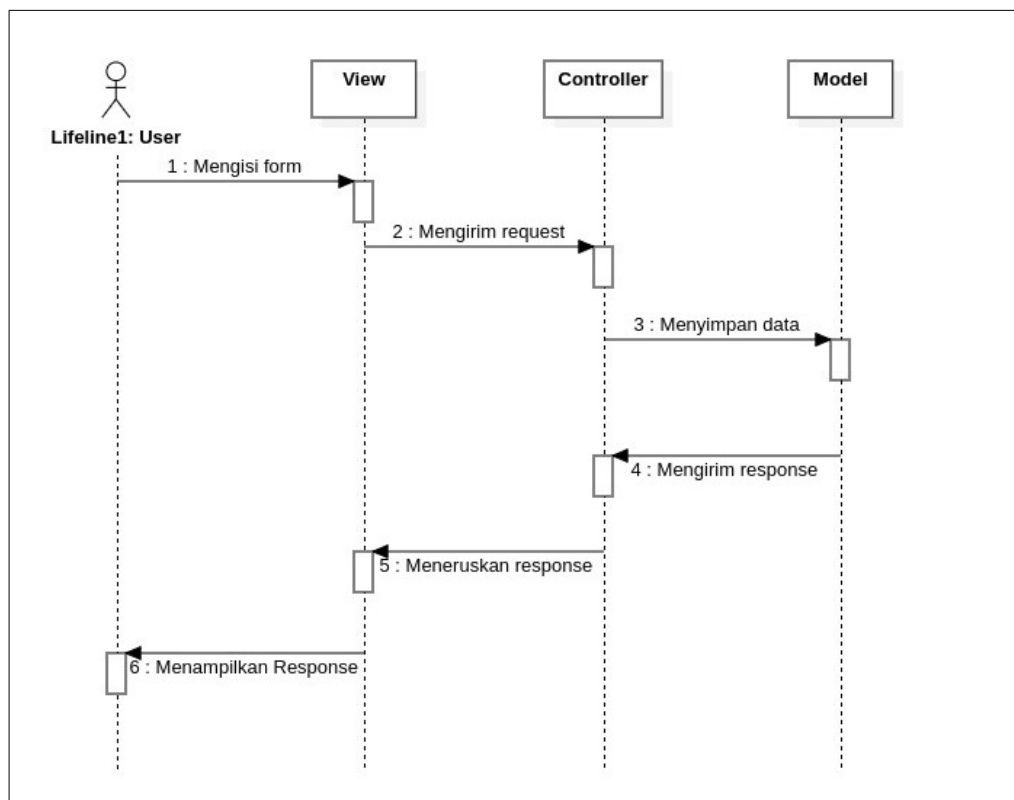
Pada *Sequence Diagram Menampilkan Tiket* dijelaskan bahwa User dapat melihat tiket pada *event* yang sudah di-join. User mengakses *event* terlebih dahulu dan mengakses menu tiket. User akan mengirim *request Controller*, lalu *Controller* akan mengambil data dari *Model* dan *Controller* akan mengirimkan *response* ke User berupa tiket.



Gambar 8: *Sequence Diagram Menampilkan Tiket*

f) *Sequecne Diagram Mengelola Event*

Pada *Sequence Diagram* mengelola *Event* dijelaskan bahwa *User* dapat mengelola *event* seperti membuat *event*. *User* mengakses menu *crete event* terlebih dahulu dan mengisi data pada *form* yang disediakan. Data dikirim ke *Controller* yang selanjutnya diteruskan ke *Model* untuk disimpan di *Database*.



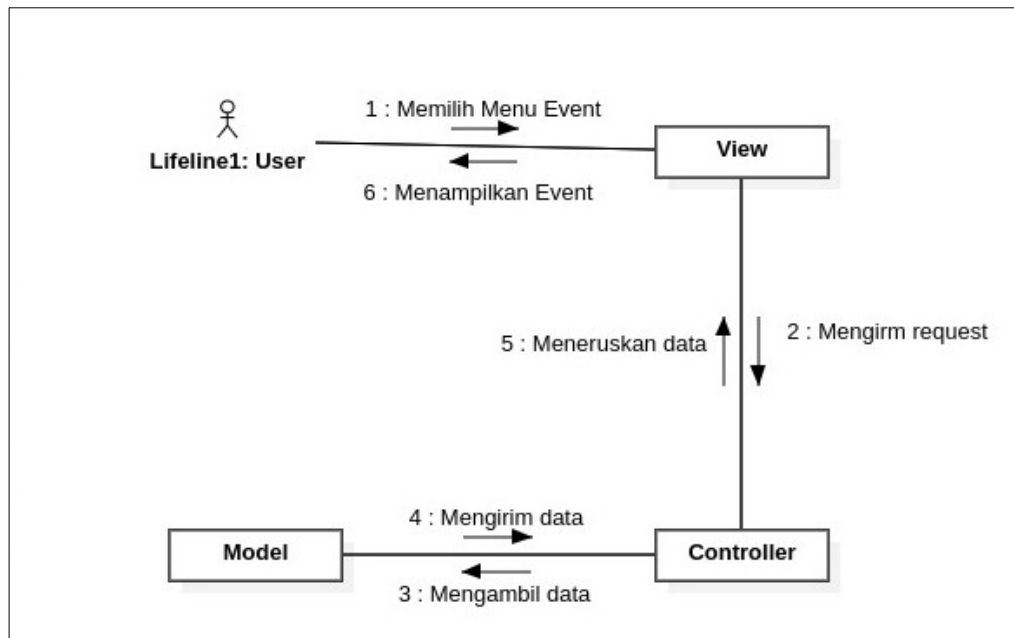
Gambar 9: Sequence Diagram Mengelola Event

3. *Collaboration Diagram*

Collaboration Diagram merupakan perluasan dari objek diagram, *Collaboration Diagram* menunjukkan *message-message* objek yang dikirim satu sama lain. *Collaboration Diagram* dan *Sequence Diagram* bisa saling mengisi, pada *Collaboration Diagram* bisa ditambahkan nomor urut pada label sebuah *message* untuk menunjukkan urutan informasi yang disampaikan. Untuk mengirim pesan pada objek, buat tanda panah pada garis asosiasi (*link*) yang menghubungkan dua objek. Arah panah menunjukkan objek yang menerima pesan, sedangkan label di dekat panah merupakan pesan yang dikirim, tipikal dari *message* meminta kepada objek yang menerimanya untuk melakukan suatu operasi.

a) ***Collaboration Diagram Melihat Event***

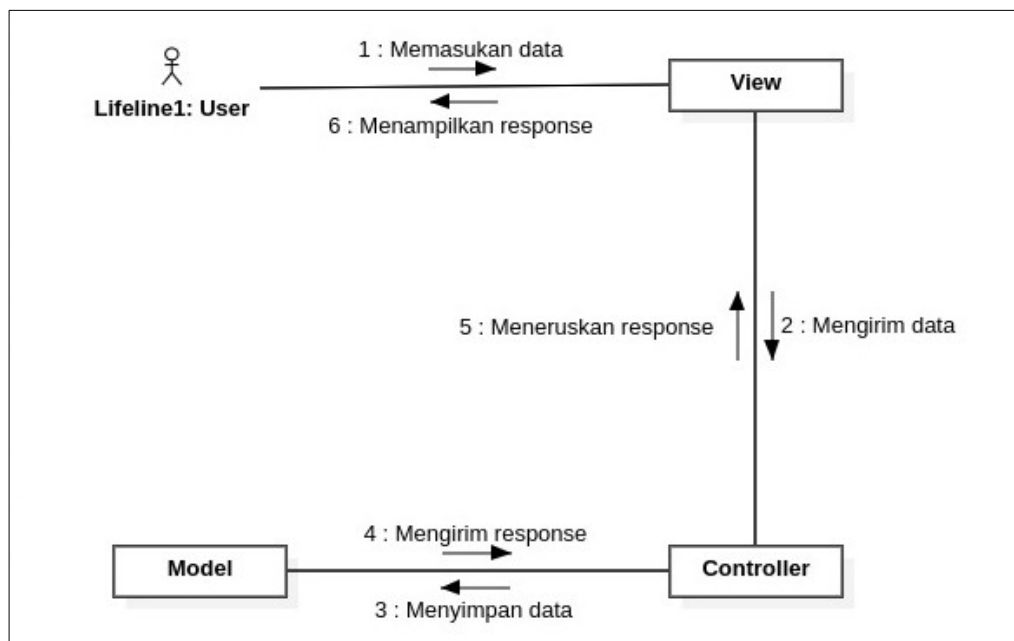
Pada *Collaboration Diagram Melihat Event* menggambarkan proses menampilkan halaman *event*. Dimulai dari *user* merequest halaman menampilkan *event* ke *Controller*. Selanjutnya *Controller* akan meminta data *event* dari *Model*, kemudian *Model* akan mengambil data dari *database* dan mengirimkan ke *Controller*. *Controller* akan merender *view* beserta data yang didapatkan dari *Model* dan ditampilkan ke *User*.



Gambar 10: Collaboration Diagram Melihat Event

b) Collaboration Diagram Register

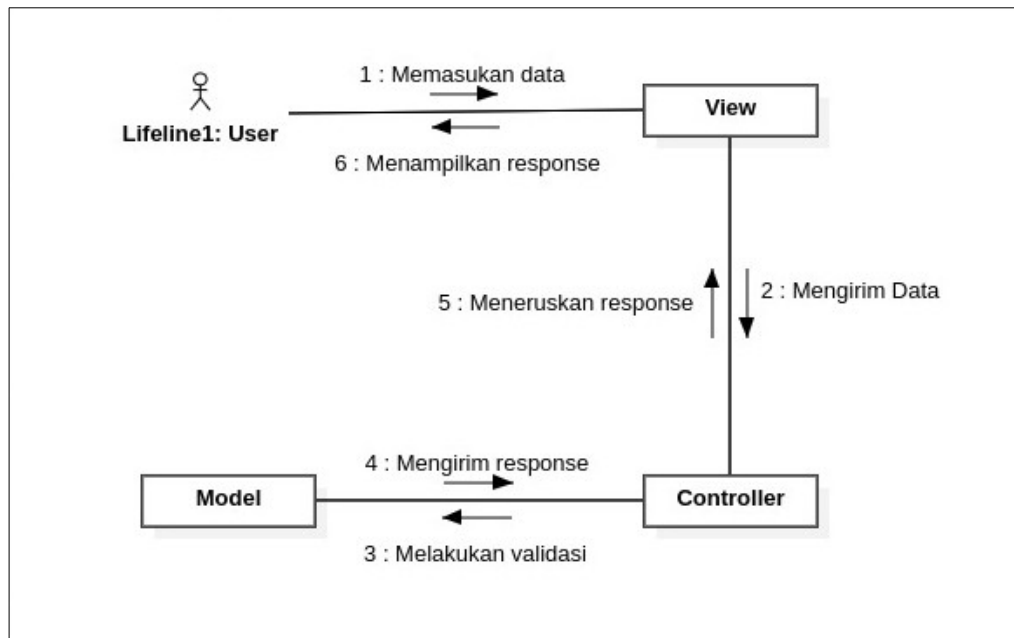
Pada *Collaboration Diagram Register* menggambarkan proses *register User*. Dimulai dari *User* mengisi data di *form* kemudian disubmit dan dikirimkan ke *Controller*. *Controller* meneruskan data ke *Model* yang nantinya akan disimpan ke dalam *database*. *Model* mengirimkan *response* ke *Controller* dan *Controller* menampilkan *response* ke *user*.



Gambar 11: Collaboration Diagram Register

c) Collaboration Diagram Login

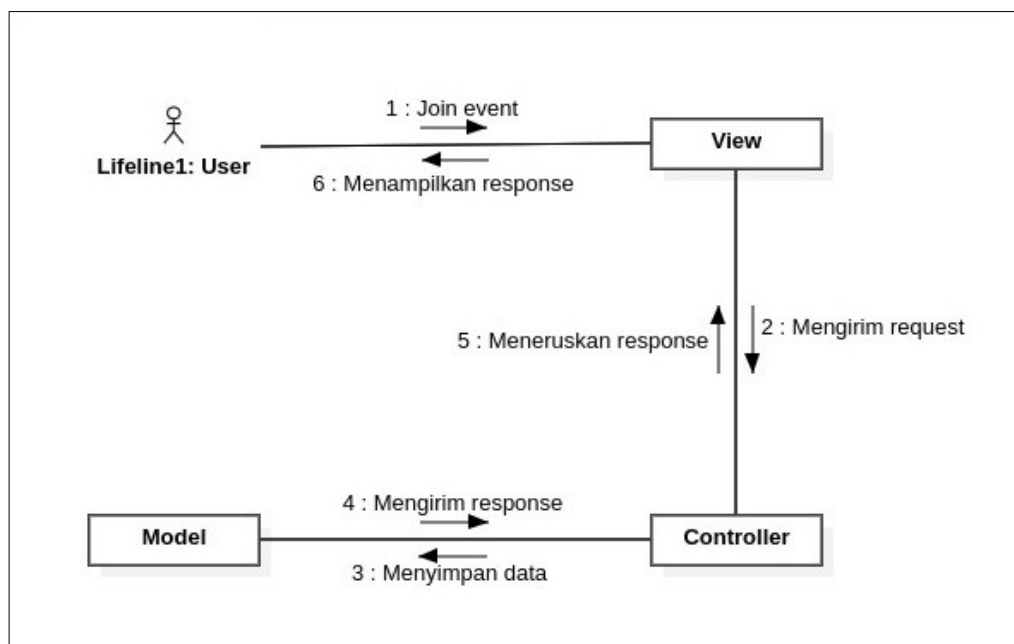
Pada *Collaboration Diagram Login* menggambarkan proses *Login User* ke Sistem. Dimulai dari *User* menginput *username* dan *password* di *form* kemudian disubmit dan dikirimkan ke *Controller*. *Controller* meneruskan data ke *Model* yang nantinya akan dilakukan verifikasi sesuai data yang terdapat di *Database*. Setelah mendapatkan hasil verifikasi, *Model* mengirimkan *response* ke *Controller* dan *Controller* menampilkan *response* ke *user*.



Gambar 12: Collaboration Diagram Login

d) Collaboration Diagram Join Event

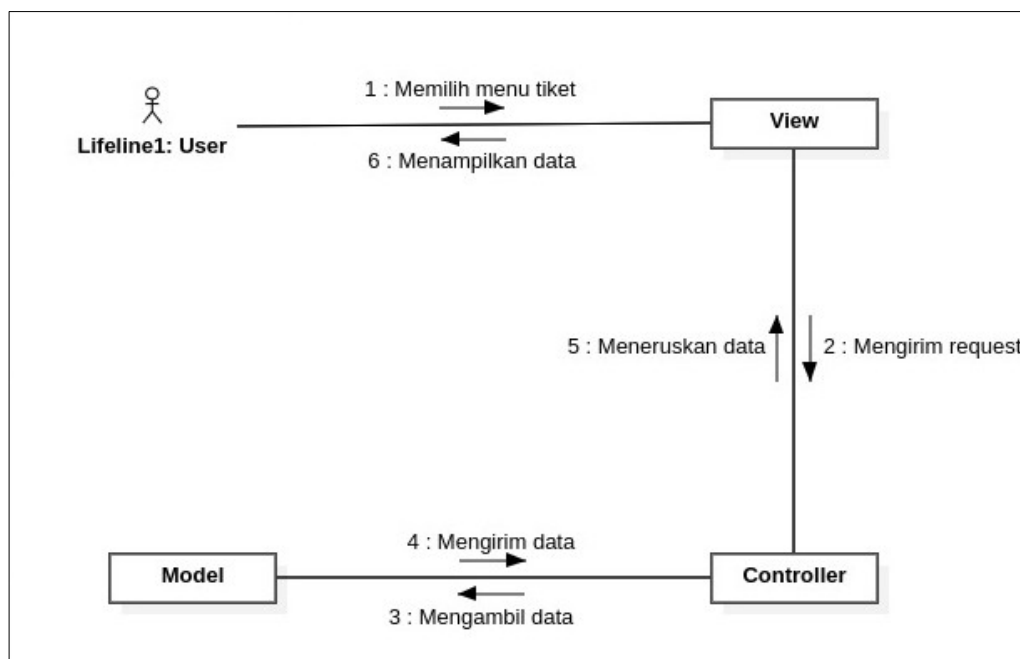
Pada *Collaboration Diagram Join Event* menggambarkan proses *join* pada sebuah Event. Dimulai dari *User* mengakses *event* dan mengklik *join* pada *Event*. Hasil *join* akan dikirim ke *Controller*. *Controller* meneruskan data ke Model yang nantinya akan disimpan ke dalam *database*. Model mengirimkan *response* ke *Controller* dan *Controller* menampilkan *response* ke *user*.



Gambar 13: Collaboration Diagram Join Event

e) **Collaboration Diagram Menampilkan Tiket**

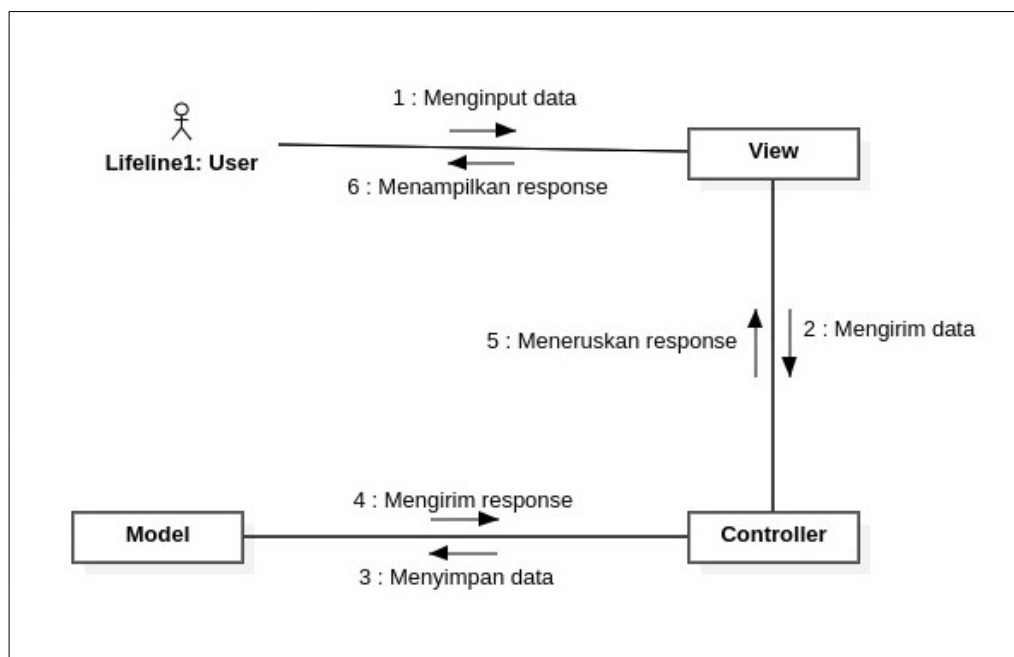
Pada *Collaboration Diagram* Menampilkan Tiket menggambarkan proses menampilkan tiket sebuah *Event*. Dimulai dari *User* mengakses *event* dan mengklik Tiket dan akan dikirim *request* ke *Controller*. *Controller* meminta data tiket *user* ke Model yang dikirim kembali ke *Controller*. *Controller* merender sebuah *view* dan tiket ditampilkan ke *User*.



Gambar 14: Collaboration Diagram Menampilkan Tiket

f) Collaboration Diagram Mengelola Event

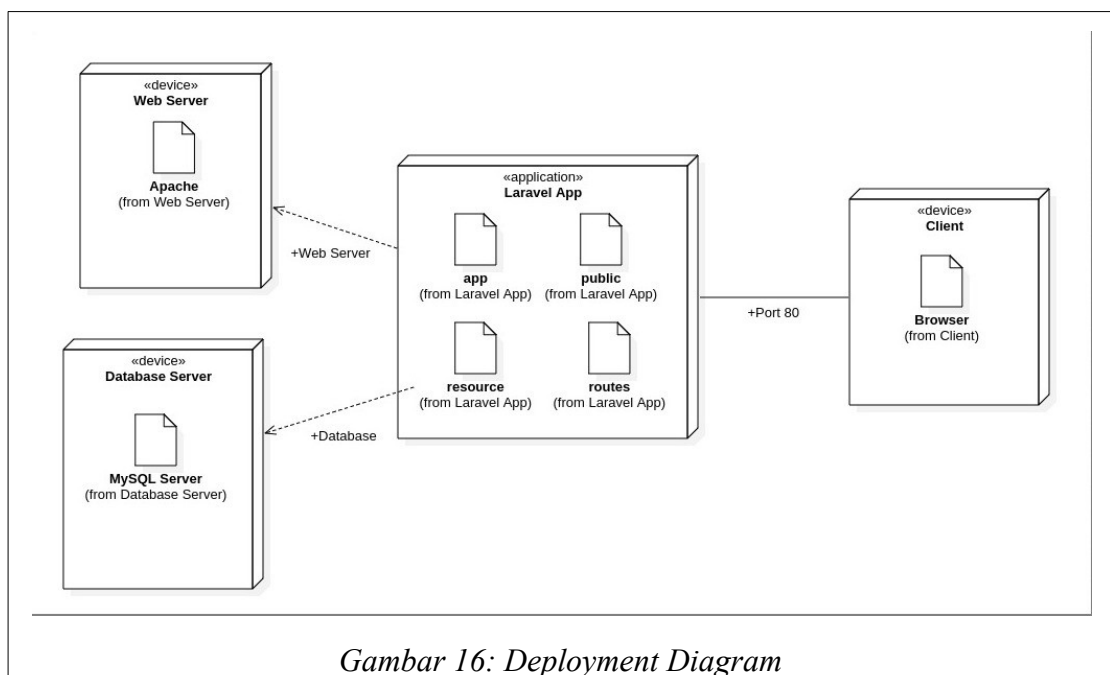
Pada *Collaboration* Diagram mengelola *Event* menggambarkan proses pembuatan *event*. Dimulai dari *User* mengakses menu *Create Event* dan mengisi data di *form* kemudian disubmit dan dikirimkan ke *Controller*. *Controller* meneruskan data ke *Model* yang nantinya akan disimpan ke dalam *database*. *Model* mengirimkan *response* ke *Controller* dan *Controller* menampilkan *response* ke *user*.



Gambar 15: Collaboration Diagram Mengelola Event

4.2.2 Deployment Diagram

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

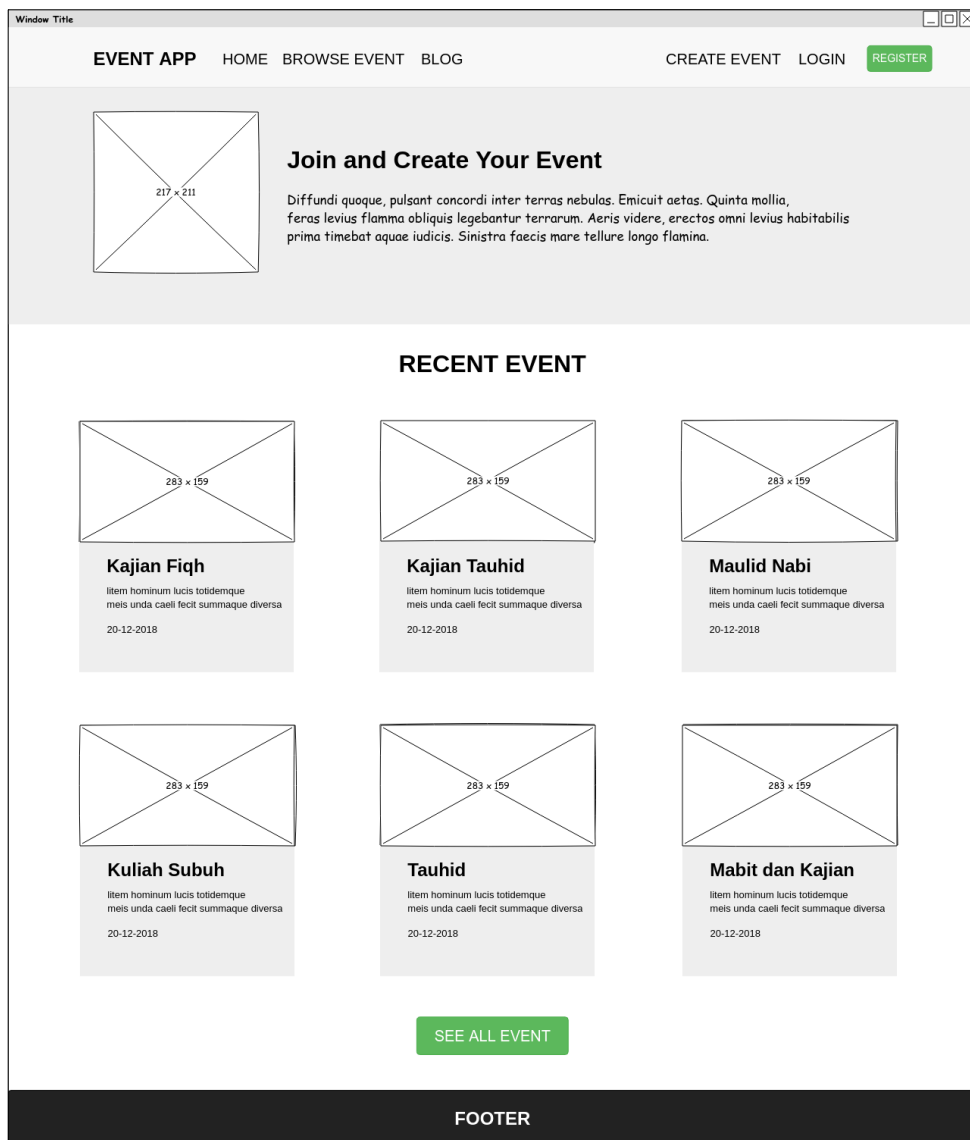


Gambar 16: Deployment Diagram

4.2.3 Antarmuka Sistem

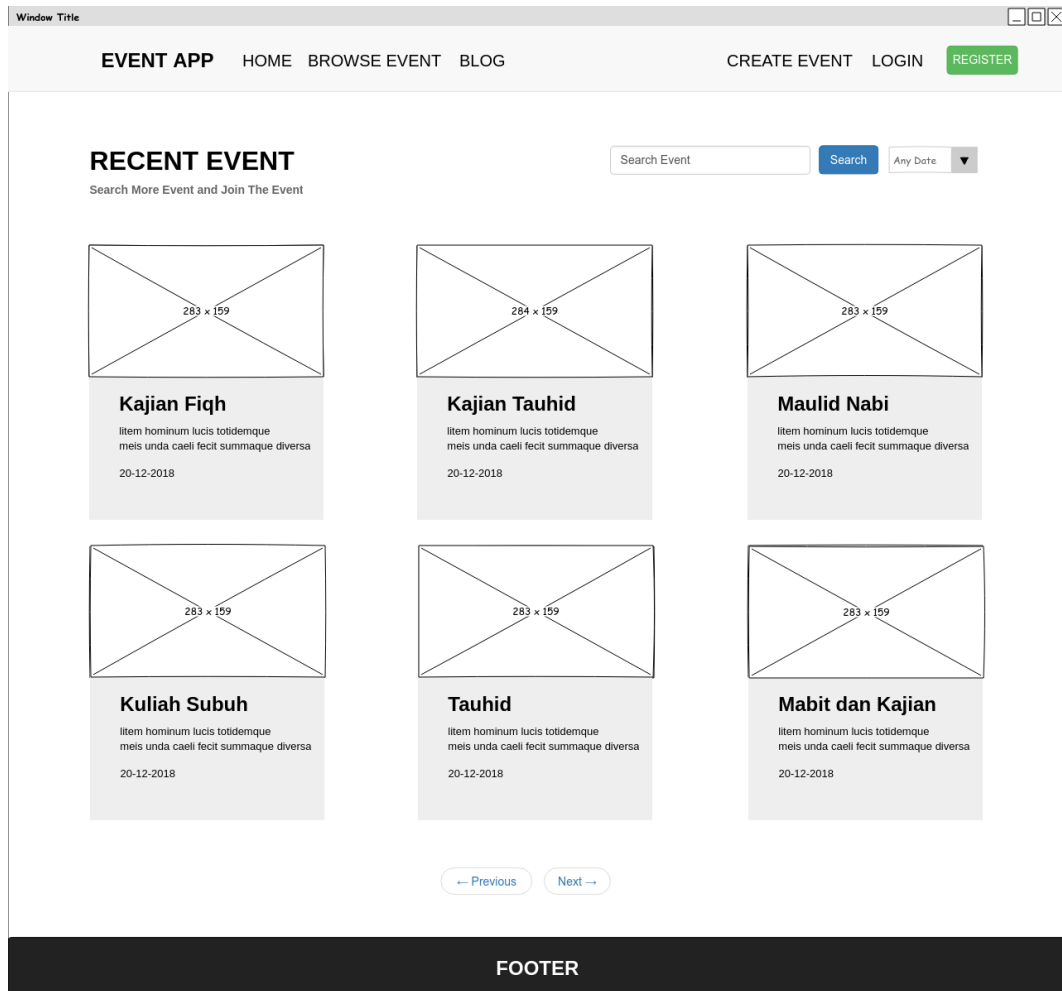
Setelah melakukan perancangan atau desain sistem, selanjutnya adalah membuat antarmuka sistem yang menggambarkan tampilan dari aplikasi yang akan dirancang. Berikut adalah antarmuka sistem utama yang akan dibangun:

1. Halaman *Home*



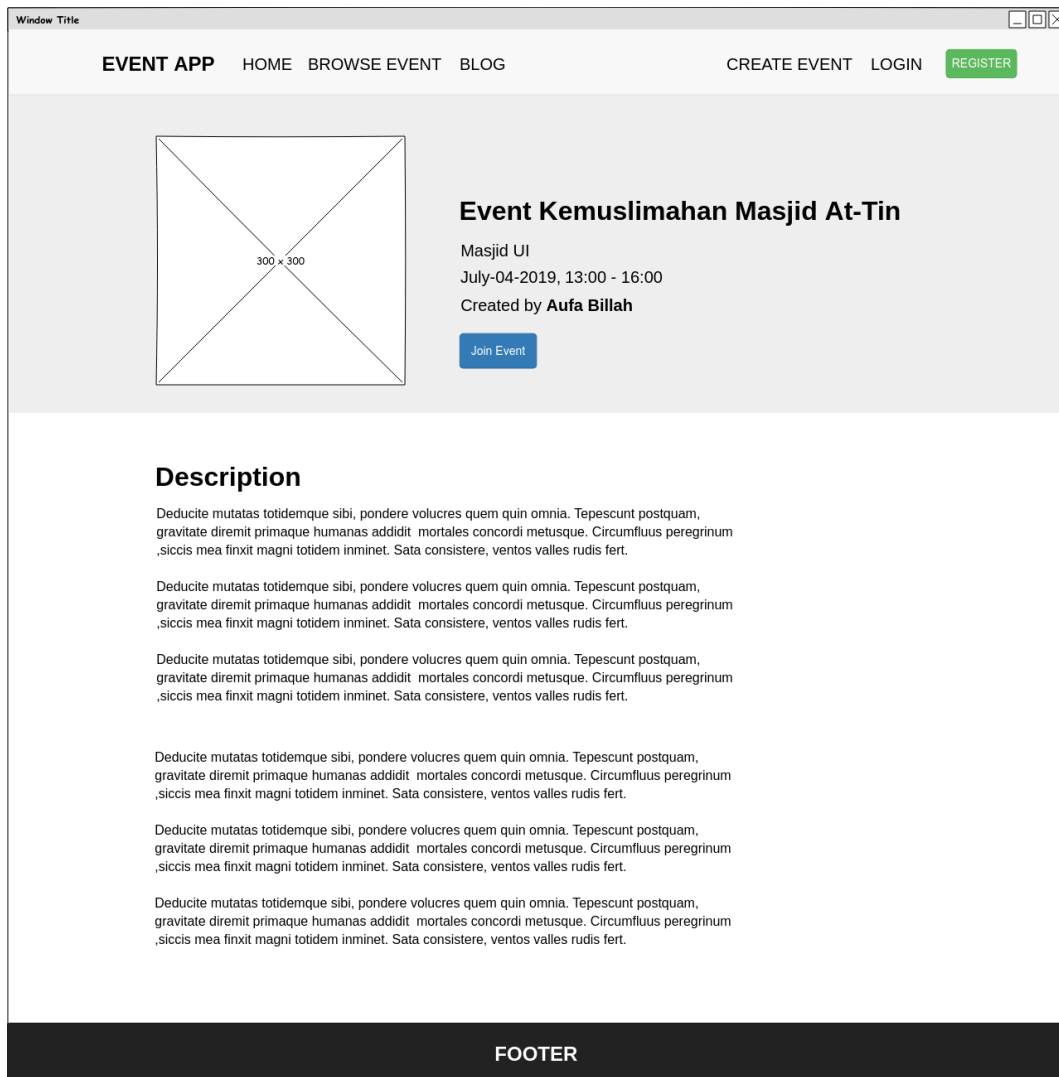
Gambar 17: Halaman Home

2. Halaman *Explore*



Gambar 18: *Explore*

3. Halaman Detail *Event*



Gambar 19: Detail Event

4. Halaman *Register*

The screenshot shows a web browser window titled "Window Title". The page has a light gray header with the text "EVENT APP" on the left and navigation links "HOME", "BROWSE EVENT", "BLOG", "CREATE EVENT", "LOGIN", and a green "REGISTER" button on the right. The main content area is white and features the heading "Welcome" followed by "Sign up". Below this are four input fields: "Username", "Email", "Password", and "Confirm Password". A blue "Register" button is positioned below the fields. Underneath the button are two links: "Forgot Password" and "Have an account? Sign in Here". The page concludes with a dark gray footer containing the word "FOOTER" in white capital letters.

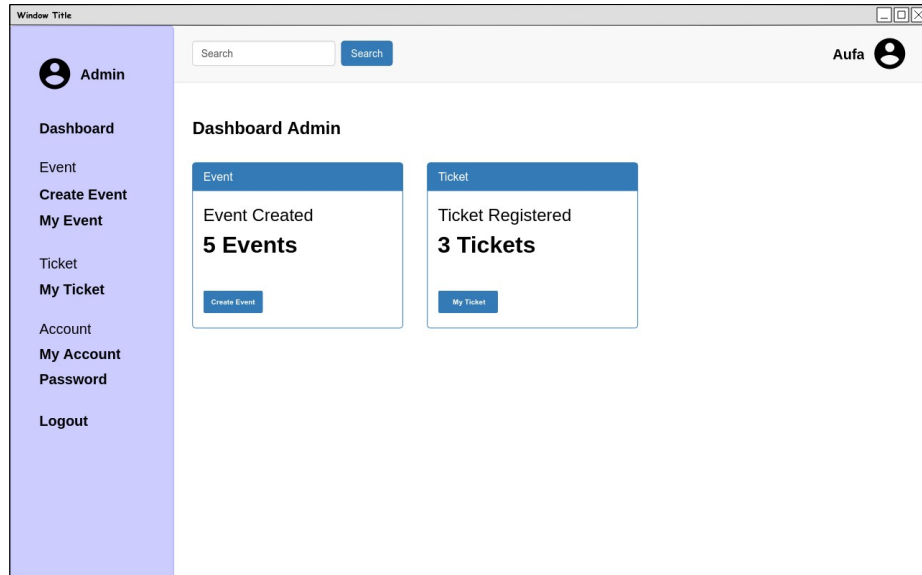
Gambar 20: Halaman Register

5. Halaman *Login*

The screenshot shows a web browser window titled "Window Title". The page has a light gray header with the text "EVENT APP" on the left and navigation links "HOME", "BROWSE EVENT", "BLOG", "CREATE EVENT", "LOGIN", and a green "REGISTER" button on the right. The main content area is white and features the heading "Welcome" followed by "Sign in". Below this are two input fields: "Username" and "Password". A blue "Login" button is positioned below the fields. Underneath the button are two links: "Forgot Password" and "Not Member? Sign Up Here". The page concludes with a dark gray footer containing the word "FOOTER" in white capital letters.

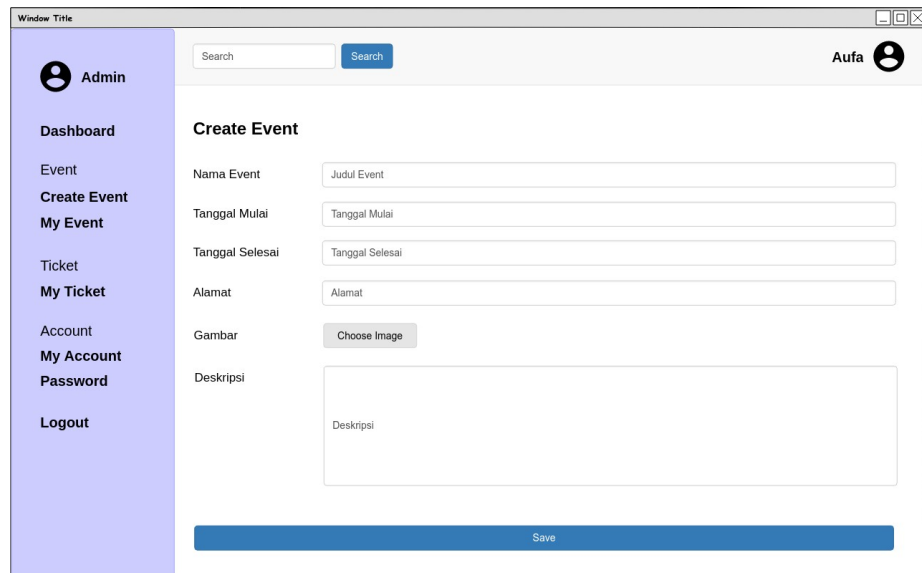
Gambar 21: Halaman Login

6. Halaman *Dashboard*



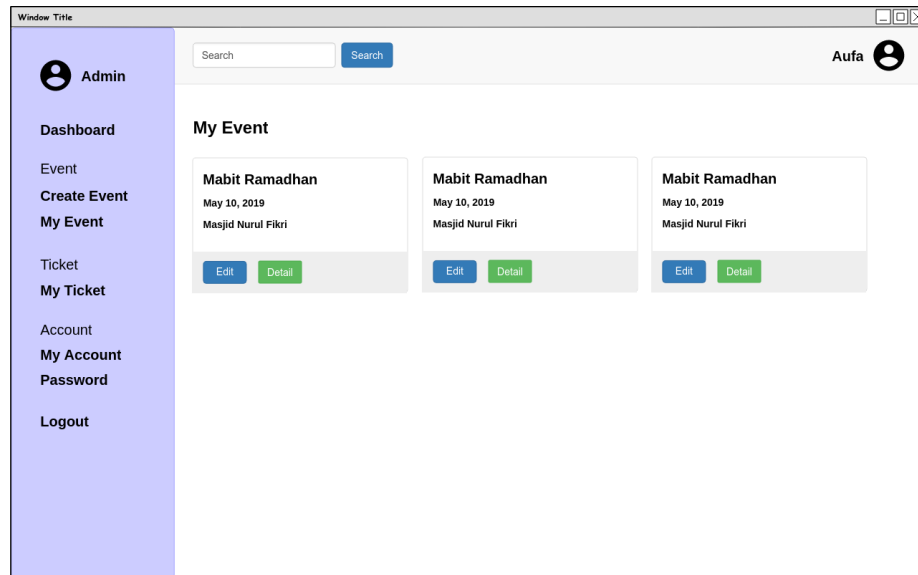
Gambar 22: Halaman Dashboard

7. Halaman *Create Event*



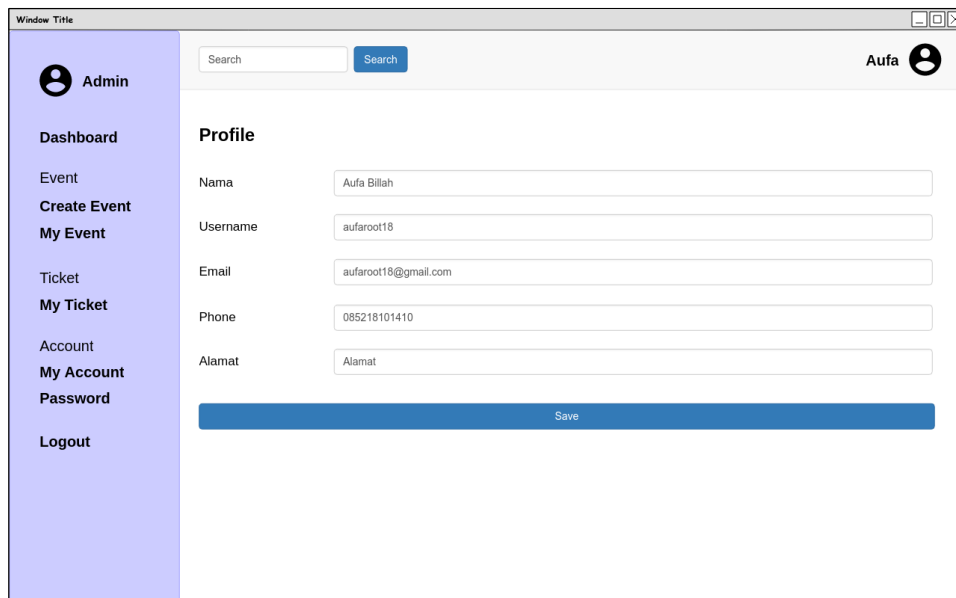
Gambar 23: Create Event

8. Halaman *My Event*



Gambar 24: Halaman My Event

9. Halaman *My Account*



Gambar 25: Halaman My Account

10. Halaman *Password*

The screenshot shows a web application interface for password management. On the left is a vertical sidebar with a light blue background, containing a list of navigation items: 'Admin' (with a user icon), 'Dashboard', 'Event', 'Create Event', 'My Event', 'Ticket', 'My Ticket', 'Account', 'My Account', 'Password', and 'Logout'. The main content area has a white background. At the top of this area is a search bar with the placeholder text 'Search' and a blue 'Search' button. To the right of the search bar is the user's name 'Aufa' next to a user icon. Below the search bar, the title 'Password' is displayed. Underneath the title are three input fields: 'Current Password', 'New Password', and 'Confirm Password'. Each field has its label to the left and the placeholder text 'Current Password', 'New Password', and 'Confirm Password' respectively. At the bottom of the form is a wide blue button labeled 'Save'.

Gambar 26: Halaman Password

4.3 Rencana Pengujian

4.3.1 Pengujian Fungsionalitas: *Black Box Testing*

Black Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifik fungsional program.

Berikut merupakan rencana pengujian *Black Box testing* yang akan dilakukan pada aplikasi.

Table 17: *Black Box Testing*

No	Skenario Pengujian	Butir Uji	Hasil yang Diharapkan
1	Mengakses Halaman <i>Home</i>	Menguji fungsional halaman <i>home</i> dalam menampilkan <i>event</i>	Menampilkan <i>event</i> terdekat (1 minggu ke depan)
2	Mengakses Halaman <i>Explore</i>	Menguji fungsional halaman <i>explore</i> untuk menampilkan semua <i>event</i>	Menampilkan semua <i>event</i> terdekat (1 minggu ke depan)
3	Mencari <i>Event</i>	Menguji pencarian <i>event</i> tanpa memasukkan kata yang dicari	Memunculkan <i>error</i> untuk mengisi kata yang ingin dicari
		Menguji pencari <i>event</i> dengan memasukkan kata yang dicari	Menampilkan <i>event</i> yang dicari berdasarkan nama <i>event</i>
4	Mengakses Detail <i>Event</i>	Menguji detail <i>event</i> dalam menampilkan informasi detail <i>event</i>	Menampilkan informasi detail dari <i>event</i>
		Menguji tombol <i>join</i> dalam keadaan <i>user</i> belum <i>login</i>	<i>User</i> akan diarahkan ke halaman <i>login</i> sebelum <i>join event</i>
		Menguji tombol <i>join</i> dalam keadaan <i>user</i> telah <i>login</i>	<i>User</i> berhasil <i>join</i> dan menampilkan tombol <i>download</i> tiket
5	Mengurutkan <i>Event</i>	Mengurutkan <i>event</i> terdekat (minggu ini)	Mengurutkan dan Menampilkan <i>event</i> terdekat (minggu ini)
		Mengurutkan <i>event</i> bulan ini	Mengurutkan dan Menampilkan <i>event</i> bulan ini
		Mengurutkan semua <i>event</i>	Mengurutkan dan Menampilkan semua <i>event</i>

6	Register Akun	Mendaftar akun dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>
		Mendaftar akun dengan memasukkan email yang sama	Menampilkan <i>error</i> bahwa email sudah terdaftar
		Mendaftar akun dengan mengisi seluruh <i>field</i>	User berhasil terdaftar
7	Login Akun	Login event dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>
		Login event dengan <i>username</i> atau <i>password</i> yang salah	Menampilkan <i>error</i> bahwa akun tidak terdaftar
		Login event dengan <i>username</i> dan <i>password</i> yang benar	Berhasil login dan diarahkan ke halaman <i>Dashboard</i>
8	Mengakses Halaman <i>Dashboard</i>	Menguji <i>redirect</i> halaman <i>dashboard</i> setelah login	Menampilkan halaman <i>dashboard</i> setelah login
		Menguji fungsional halaman <i>dashboard</i>	Menampilkan informasi seputar <i>event</i> di halaman <i>Dashboard</i>
9	Membuat <i>Event</i>	Membuat <i>event</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>
		Membuat <i>event</i> dengan mengisi semua <i>field</i>	<i>Event</i> berhasil dibuat dan memunculkan pesan berhasil
10	Mengakses <i>My Event</i>	Menguji fungsional <i>My Event</i>	Menampilkan <i>event</i> yang sudah dibuat
		Mengedit <i>event</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>
		Mengedit <i>event</i> dengan mengisi semua <i>field</i>	<i>Event</i> berhasil diedit dan memunculkan pesan berhasil diedit
		Mengklik <i>event</i> dan diarahkan ke detail event	Diarahkan ke detail <i>event</i>
11	Mengakses <i>My Ticket</i>	Menguji fungsional <i>My Ticket</i>	Menampilkan <i>event</i> yang sudah dijoin.
		Menguji <i>download ticket event</i>	Menampilkan <i>ticket event</i> dan bisa didownload
		Mengklik <i>event</i> dan diarahkan ke detail event	Diarahkan ke detail <i>event</i>
12	Mengakses <i>My Account</i>	Menguji fungsional <i>My Account</i>	Menampilkan informasi tentang <i>user</i>
		Mengedit salah satu <i>field</i>	Berhasil diedit dan memunculkan informasi baru

13	Mengakses <i>Password</i>	Menguji fungsional <i>Password</i>	Menampilkan halaman <i>password</i>
		Mengubah <i>password</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan pesan <i>error</i> untuk mengisi semua <i>field</i>
		Mengubah <i>password</i> namun <i>password</i> lama salah	Menampilkan pesan <i>error</i> bahwa <i>password</i> yang lama salah
		Mengubah <i>password</i> namun <i>field new password</i> dan <i>confirmation</i> tidak cocok	Menampilkan pesan <i>error</i> bahwa <i>new password</i> dan <i>confirmation</i> tidak cocok
		Mengubah <i>password</i> dan semua <i>field</i> sesuai	Mengubah <i>password</i> lama ke <i>password</i> baru
14	<i>Logout</i>	Menguji tombol <i>logout</i>	<i>User</i> berhasil <i>logout</i> dan diarahkan ke halaman <i>login</i>

4.3.2 Pengujian User: *User Acceptance Test (UAT)*

User Acceptance Test (UAT) merupakan suatu rangkaian proses pengujian di tingkat akhir fase pembangunan suatu aplikasi, di mana pengujian ini dilakukan oleh *end user* untuk mengetahui apakah aplikasi yang dibangun sudah sesuai dengan *User requirement*.

Berikut merupakan contoh rencana pengujian yang akan dilakukan oleh *user* dengan menjawab pertanyaan yang tersedia setelah melakukan uji coba aplikasi.

Table 18: *UAT*

No	Fitur	Deskripsi
1	<i>Home</i>	<i>User</i> mengakses halaman <i>home</i> untuk melihat <i>event</i>
2	<i>Search Event</i>	<i>User</i> mencari <i>event</i> berdasarkan nama <i>event</i>
3	<i>Sort Event</i>	<i>User</i> mengurutkan <i>event</i> berdasarkan kategori yang tersedia
4	<i>Detail Event</i>	<i>User</i> bisa melihat informasi detail suatu <i>event</i>
5	<i>Register</i>	<i>User</i> bisa membuat akun
6	<i>Login</i>	<i>User</i> bisa <i>login</i> menggunakan akun yang sudah dibuat
7	<i>Join Event</i>	<i>User</i> bisa <i>join event</i> dengan syarat harus <i>login</i> terlebih dahulu
8	<i>Dashboard</i>	<i>User</i> bisa mengakses halaman <i>dashboard</i> setelah <i>login</i>
9	<i>Create Event</i>	<i>User</i> bisa membuat <i>event</i>
10	<i>Edit Event</i>	<i>User</i> bisa mengedit <i>event</i> yang telah dibuat
11	<i>Tiket</i>	<i>User</i> mendownload tiket <i>event</i>
12	<i>Account</i>	<i>User</i> bisa mengubah informasi akun
13	<i>Password</i>	<i>User</i> bisa mengubah <i>password</i>

4.3.3 Kuesioner

Kuesioner berikut ditujukan kepada umum untuk menilai hasil aplikasi. Kuesioner berisi beberapa pernyataan dan dijawab menggunakan 5 pilihan skala yang mempunyai gradasi dari Sangat Setuju (SS) hingga Sangat Tidak Setuju (STS). Berikut tabel skala *likert* beserta nilainya.

Table 19: Skala Likert

Variabel	Keterangan	Nilai
Sangat Setuju	SS	5
Setuju	S	4
Ragu-Ragu	RG	3
Tidak Setuju	TS	2
Sangat Tidak Setuju	STS	1

Adapun Kuesioner sebagai berikut:

Table 20: Kuisisioner

No	Pertanyaan	SS	S	RG	TS	STS
1	User dengan mudah menemukan dan mencari Event					
2	User dengan mudah membuat Event					
3	User dengan mudah melakukan register Event					
4	User menemukan menu dengan mudah					
5	Fungsionalitas web sudah cukup baik					
6	Tampilan web sudah cukup baik					

BAB V

IMPLEMENTASI DAN EVALUASI

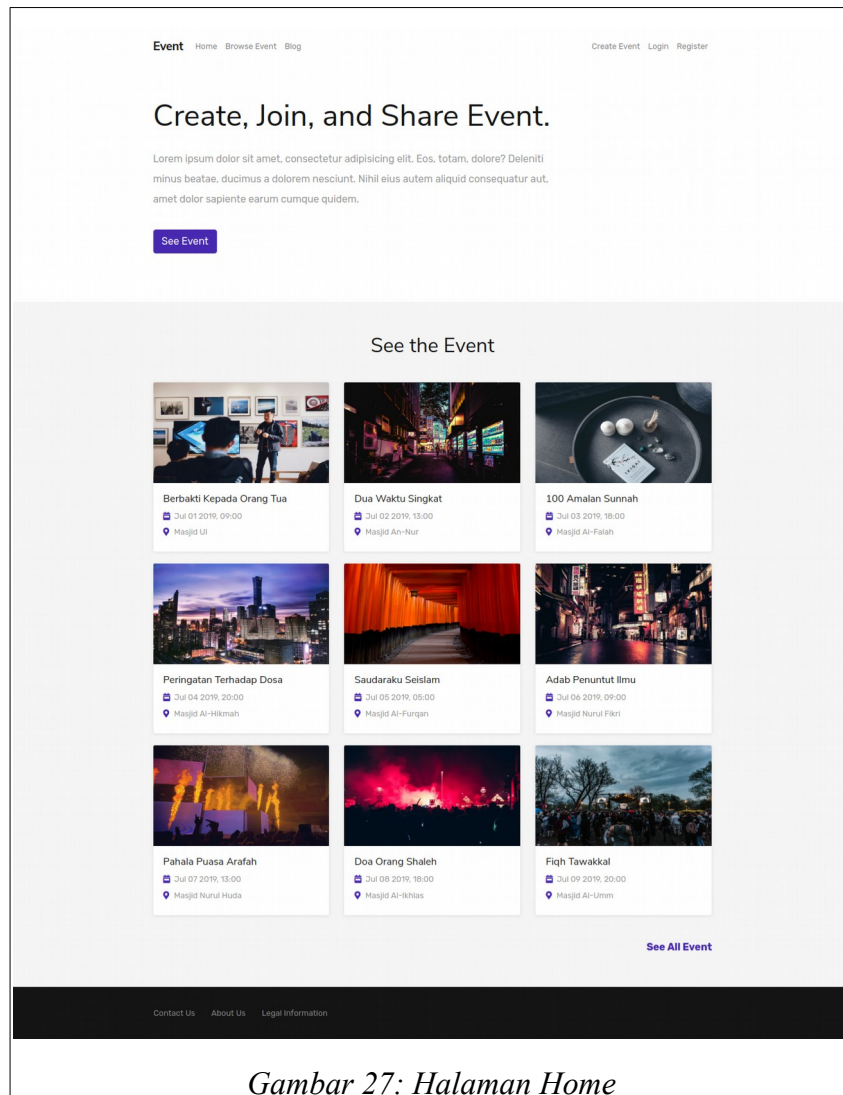
5.1 Implementasi

Implementasi adalah proses pengembangan aplikasi yang telah dirancang pada tahapan sebelumnya, sehingga diperoleh hasil berupa aplikasi yang sesuai dengan *requirement* yang telah didefinisikan pada tahapan Analisis dan Perancangan Sistem.

5.1.1 Implementasi antarmuka

Pada bagian implementasi antarmuka akan ditampilkan hasil dari antarmuka aplikasi yang sudah dibuat serta akan dijelaskan hasil dari implementasi tersebut.

1. Halaman *Home*

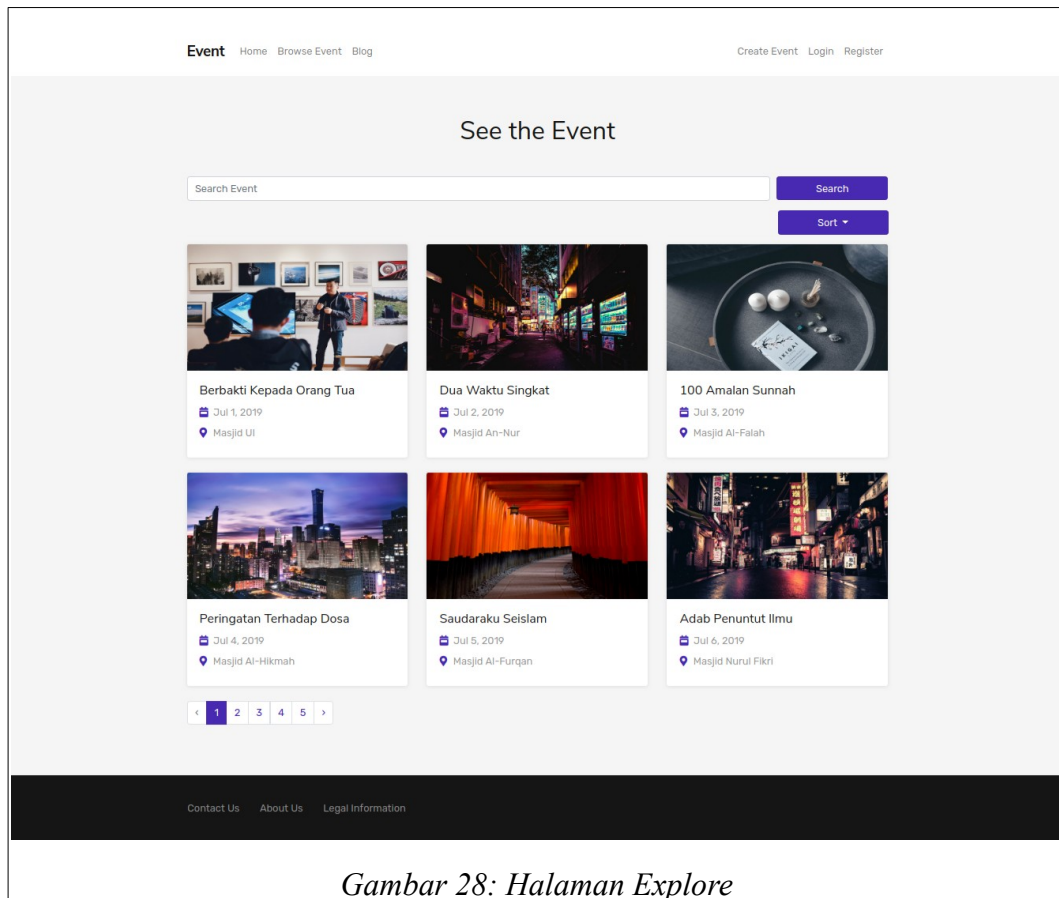


Gambar 27: Halaman Home

Gambar 27 merupakan implementasi antarmuka Halaman *Home*. Halaman *home* terdiri dari beberapa bagian yaitu menu navigasi di bagian atas, bagian *event-event* yang ditampilkan di bagian tengah, dan terdapat menu *See All Event* untuk melihat seluruh *event*.

Event yang ditampilkan di Halaman *Home* adalah *event* 1 minggu ke depan yang terhitung dari hari sekarang (*today*).

2. Halaman *Explore*



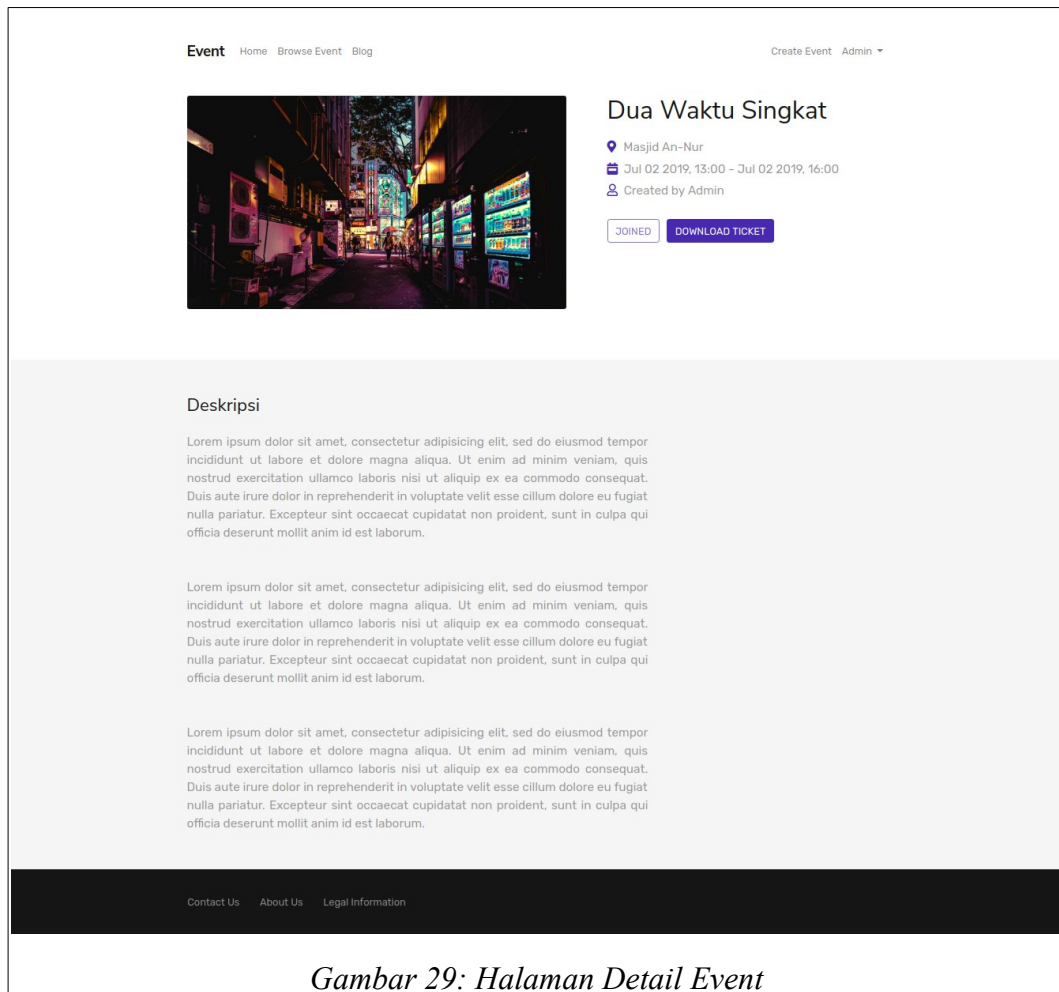
Gambar 28: Halaman *Explore*

Gambar 28 merupakan implementasi antarmuka halaman *explore*. *User* bisa mengakses halaman *explore* pada menu navigasi ***Browse Event***.

Pada halaman *explore*, *User* dapat melihat, mencari, serta mengurutkan *event* berdasarkan kategori tertentu yaitu *this week*, *this month*, dan *all event*.

Event default yang ditampilkan pada halaman *explore* adalah *event* 1 minggu yang akan datang terhitung dari hari sekarang (*today*).

3. Halaman Detail *Event*



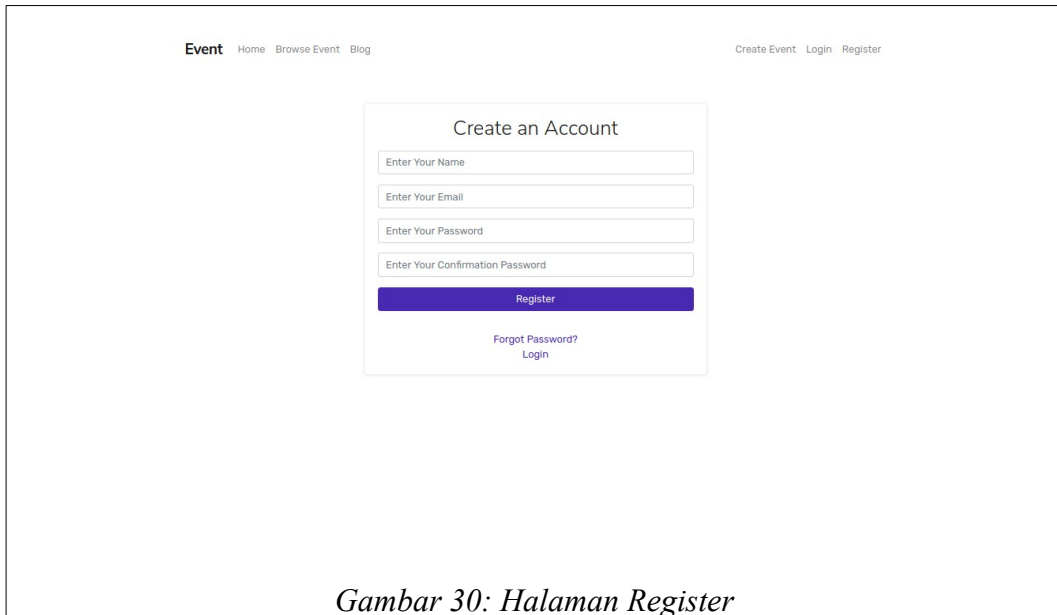
Gambar 29: Halaman Detail Event

Gambar 29 merupakan implementasi antarmuka halaman detail *event*. *User* akan diarahkan ke halaman detail *event* ketika mengklik sebuah *event* yang akan menampilkan detail dari sebuah *event*.

Halaman detail *event* menampilkan informasi detail mengenai sebuah *event* seperti gambar, judul, lokasi, tanggal mulai, tanggal selesai, pembuat *event*, tombol *join event*, serta deskripsi.

User yang ingin bergabung atau *join event* harus *login* terlebih dahulu.

4. Halaman *Register*



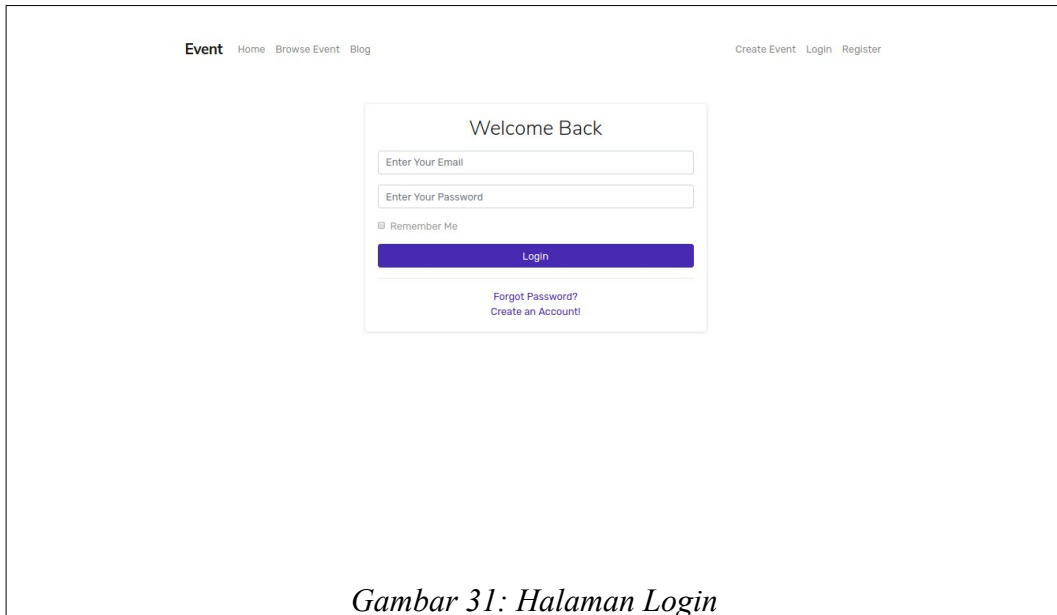
The screenshot displays a web page for creating an account. At the top left, the word "Event" is followed by links for "Home", "Browse Event", and "Blog". At the top right, there are links for "Create Event", "Login", and "Register". The main content area features a "Create an Account" form with four input fields: "Enter Your Name", "Enter Your Email", "Enter Your Password", and "Enter Your Confirmation Password". Below these fields is a prominent blue "Register" button. Underneath the button, there are links for "Forgot Password?" and "Login".

Gambar 30: Halaman Register

Gambar 30 merupakan implementasi antarmuka halaman *Register*. *User* dapat membuat akun pada halaman *register* dengan mengisi informasi pada *form* yang disediakan.

User diharuskan mendaftar terlebih dahulu untuk menggunakan fitur-fitur pada aplikasi seperti membuat *event*, *join event*, *download* tiket, halaman *dashboard*, dll.

5. Halaman *Login*



The screenshot displays a web application's login interface. At the top, a navigation bar includes the site name 'Event' and links for 'Home', 'Browse Event', and 'Blog' on the left, and 'Create Event', 'Login', and 'Register' on the right. The main content area features a central white box with a light gray border. Inside this box, the heading 'Welcome Back' is centered. Below the heading are two input fields: 'Enter Your Email' and 'Enter Your Password'. A checkbox labeled 'Remember Me' is positioned below the password field. A prominent blue 'Login' button is centered below the inputs. At the bottom of the box, two links are provided: 'Forgot Password?' and 'Create an Account!'.

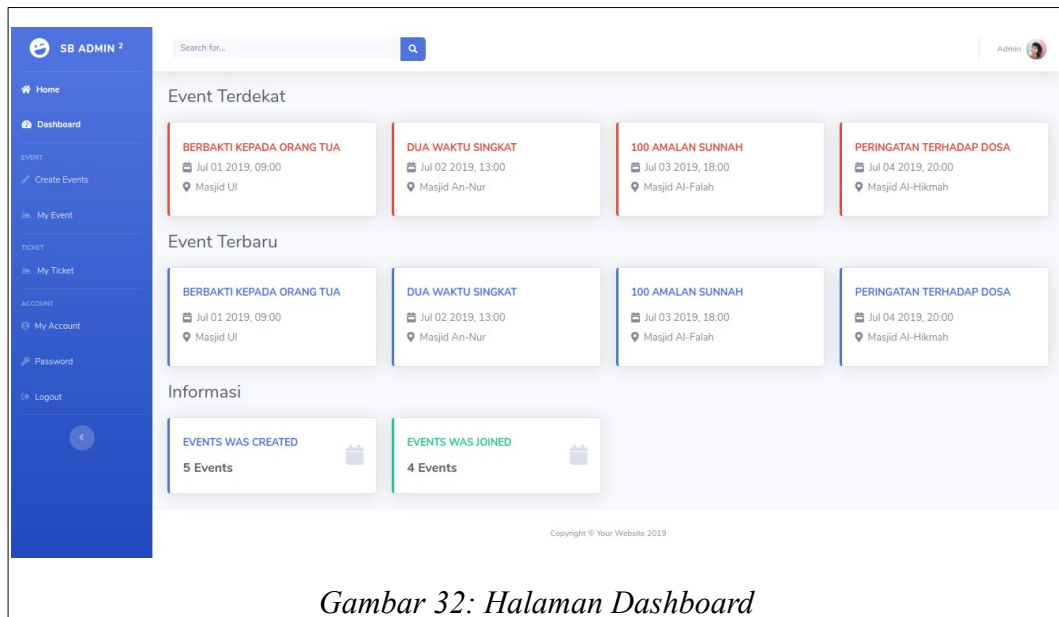
Gambar 31: Halaman Login

Pada gambar 31 merupakan implementasi antarmuka halaman *Login*. *User* dapat melakukan *login* dengan mengisi informasi pada *form* yang disediakan.

User yang sudah membuat akun dapat mengakses halaman *login* untuk masuk ke aplikasi dan menggunakan fitur-fitur yang tersedia.

User yang berhasil *login* akan diarahkan ke halaman *dashboard*.

6. Halaman *Dashboard*



Gambar 32: Halaman *Dashboard*

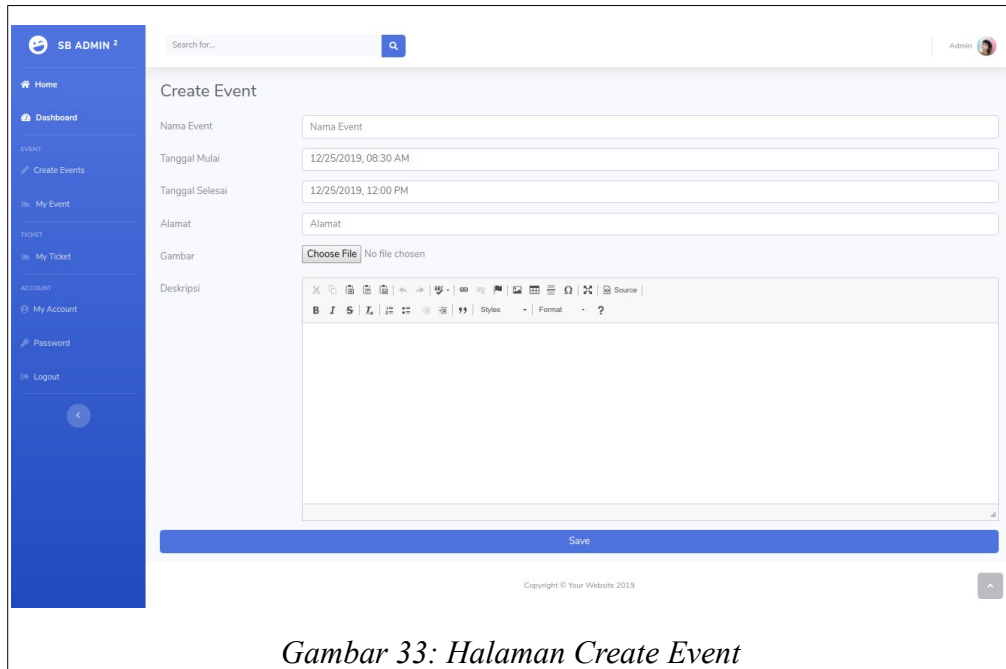
Gambar 32 merupakan implementasi antarmuka Halaman *Dashboard*. User yang berhasil *login* akan diarahkan ke Halaman *Dashboard*.

Halaman *Dashboard* berisi beberapa bagian yang terdiri dari menu navigasi yang berada di sebelah kiri dan halaman utama *dashboard* yang berisi informasi seputar *event*.

Menu navigasi yang berada di sebelah kiri merupakan fitur-fitur yang dapat digunakan oleh User seperti *Create Event*, *My Event*, *My Ticket*, *My Account*, *Password*, dan *Logout*.

Pada halaman utama *Dashboard* terdapat informasi seputar *event* terdekat, *event* terbaru, *event* yang telah dibuat, dan *event* yang telah diikuti (*joined*).

7. Halaman *Create Event*



Gambar 33: Halaman *Create Event*

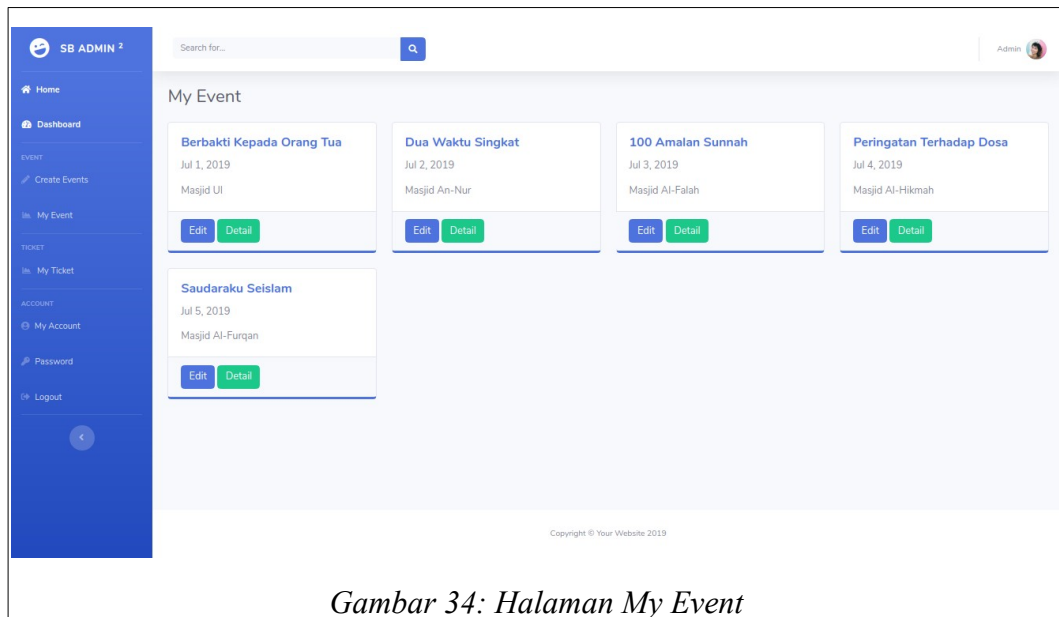
Gambar 33 merupakan implementasi antarmuka *Create Event*. *User* dapat membuat *event* dengan mengakses *Create Event* di sebelah kiri menu navigasi.

User dapat membuat *event* dengan mengisi informasi tentang *event* pada *form* yang disediakan. Informasi yang harus diisi berupa:

1. Nama *Event*.
2. Tanggal Mulai *Event*.
3. Tanggal Selesai *Event*.
4. Alamat *Event*.
5. Gambar *Event*.
6. Deskripsi *Event*.

Setelah selesai mengisi informasi yang dibutuhkan, *User* dapat mengklik tombol *save* untuk menyimpan dan mempublikasi *Event*.

8. Halaman *My Event*

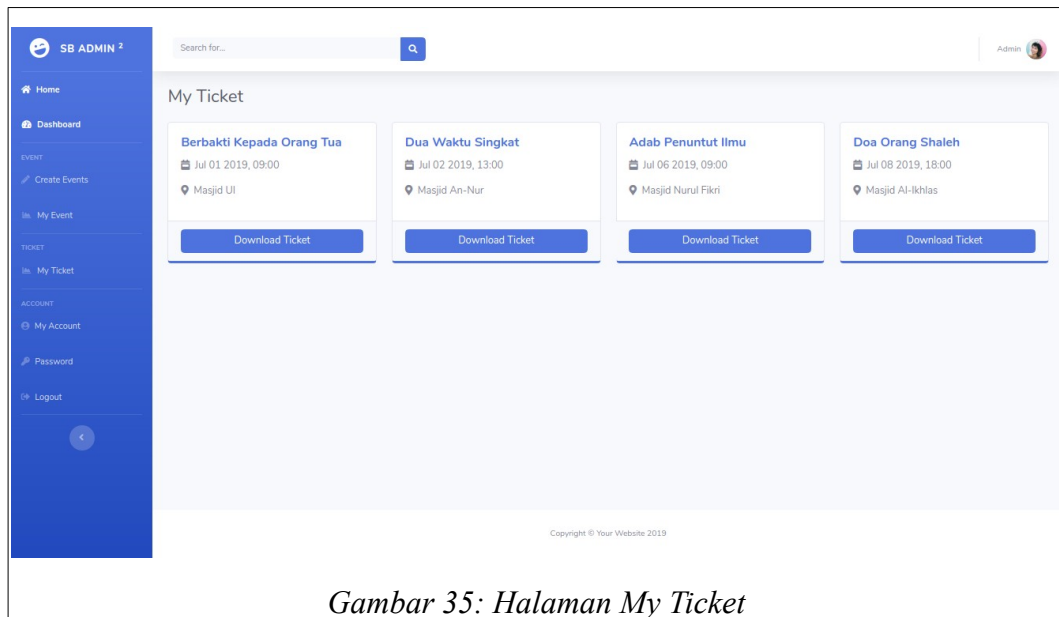


Gambar 34: Halaman *My Event*

Gambar 34 merupakan implementasi antarmuka *My Event*. *User* dapat mengakses *My Event* melalui menu navigasi di sebelah kiri.

My Event merupakan halaman yang berisi *event-event* yang telah dibuat oleh *user*. *User* dapat melihat setiap *event* yang telah dibuat, dapat mengedit setiap *event* yang telah dibuat, dan melihat informasi detail dari setiap *event*.

9. Halaman *My Ticket*

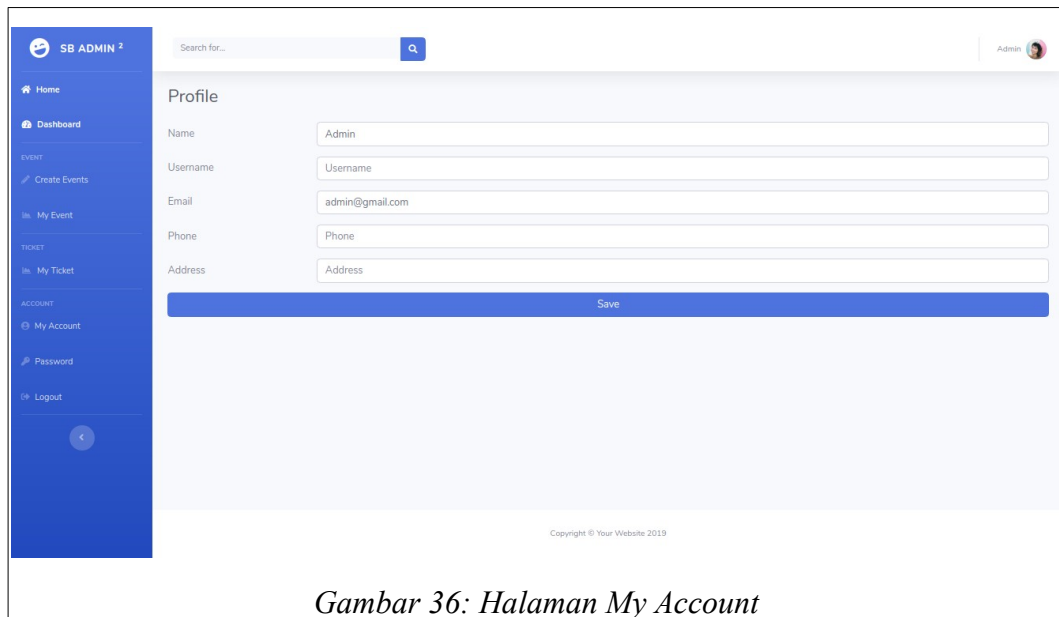


Gambar 35: Halaman *My Ticket*

Gambar 35 merupakan implementasi antarmuka *My Ticket*. *User* dapat mengakses *My Ticket* melalui menu navigasi di sebelah kiri.

Halaman *My Ticket* merupakan halaman yang berisi *event-event* yang sudah diikuti atau yang sudah di-join. *User* dapat mendownload ticket pada setiap *event* yang sudah diikuti atau di-join.

10. Halaman *My Account*



Gambar 36: Halaman My Account

Gambar 36 merupakan implementasi antarmuka *My Account*. *User* dapat mengakses *My Account* melalui menu navigasi di sebelah kiri.

Halaman *My Account* merupakan halaman informasi *user*. *User* dapat mengubah informasi seputar akun seperti *Name*, *Username*, *Email*, *Phone*, dan *Address*.

11. Halaman *Password*

SB ADMIN 2

Search for...

Admin

Home

Dashboard

EVENT

Create Events

My Event

TICKET

My Ticket

ACCOUNT

My Account

Password

Logout

Profile

Current Password

New Password

Confirm New Password

Save

Copyright © Your Website 2019

Gambar 37: Halaman Password

Gambar 37 merupakan implementasi antarmuka *Password*. *User* dapat mengakses *Password* melalui menu navigasi di sebelah kiri.

Halaman *Password* merupakan halaman yang digunakan oleh *User* untuk mengubah *password*. *User* dapat mengubah *password* dengan mengisi informasi yang disediakan pada form berupa:

1. *Current Password*: Password lama.
2. *New Password*: Password baru.
3. *Confirm New Password*: Konfirmasi Password baru.

5.2 Evaluasi Sistem

Evaluasi sistem adalah proses untuk mengevaluasi serta menguji sistem yang sudah dikembangkan untuk mengetahui aplikasi sudah berjalan sesuai dengan *requirement*.

Untuk mendapatkan hasil dari aplikasi yang sudah dikembangkan, maka akan dilakukan evaluasi sistem melalui beberapa tahapan yaitu:

1. Pengujian *Black Box*: Pengujian fungsional yang dilakukan oleh pengembang aplikasi.
2. UAT (*User Acceptance Test*): Pengujian yang dilakukan oleh *user*.
3. Kuesioner: Mendapatkan *feedback* dari aplikasi yang sudah berjalan.

5.2.1 Hasil *Black Box Testing*

Table 21: Hasil *Black Box Testing*

No	Skenario Pengujian	Butir Uji	Hasil yang Diharapkan	Hasil Pengujian
1	Mengakses Halaman <i>Home</i>	Menguji fungsional halaman <i>home</i> dalam menampilkan event	Menampilkan <i>event</i> terdekat (1 minggu ke depan)	Sesuai
2	Mengakses Halaman <i>Explore</i>	Menguji fungsional halaman <i>explore</i> untuk menampilkan semua <i>event</i>	Menampilkan semua <i>event</i> terdekat (1 minggu ke depan)	Sesuai
3	Mencari <i>Event</i>	Menguji pencarian <i>event</i> tanpa memasukkan kata yang dicari	Memunculkan <i>error</i> untuk mengisi kata yang ingin dicari	Sesuai
		Menguji pencarian <i>event</i> dengan memasukkan kata yang dicari	Menampilkan <i>event</i> yang dicari berdasarkan nama event	Sesuai
4	Mengakses Detail <i>Event</i>	Menguji detail <i>event</i> dalam menampilkan informasi detail <i>event</i>	Menampilkan informasi detail dari <i>event</i>	Sesuai
		Menguji tombol <i>join</i> dalam keadaan <i>user</i> belum <i>login</i>	<i>User</i> akan diarahkan ke halaman <i>login</i> sebelum <i>join event</i>	Sesuai
		Menguji tombol <i>join</i> dalam	<i>User</i> berhasil <i>join</i> dan menampilkan	Sesuai

		keadaan user telah <i>login</i>	tombol <i>download ticket</i>	
5	Mengurutkan Event	Mengurutkan <i>event</i> terdekat (minggu ini)	Mengurutkan dan Menampilkan <i>event</i> terdekat (minggu ini)	Sesuai
		Mengurutkan <i>event</i> bulan ini	Mengurutkan dan Menampilkan <i>event</i> bulan ini	Sesuai
		Mengurutkan semua <i>event</i>	Mengurutkan dan Menampilkan semua <i>event</i>	Sesuai
6	Register Akun	Mendaftar akun dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>	Sesuai
		Mendaftar akun dengan memasukkan email yang sama	Menampilkan <i>error</i> bahwa email sudah terdaftar	Sesuai
		Mendaftar akun dengan mengisi seluruh <i>field</i>	User berhasil terdaftar	Sesuai
7	Login Akun	<i>Login event</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>	Sesuai
		<i>Login event</i> dengan <i>username</i> atau <i>password</i> yang salah	Menampilkan <i>error</i> bahwa akun tidak terdaftar	Sesuai
		<i>Login event</i> dengan <i>username</i> dan <i>password</i> yang benar	Berhasil <i>login</i> dan diarahkan ke halaman <i>Dashboard</i>	Sesuai
8	Mengakses Halaman <i>Dashboard</i>	Menguji <i>redirect</i> halaman <i>dashboard</i> setelah <i>login</i>	Menampilkan halaman <i>dashboard</i> setelah <i>login</i>	Sesuai
		Menguji fungsional halaman <i>dashboard</i>	Menampilkan informasi seputar <i>event</i> di halaman <i>Dashboard</i>	Sesuai
9	Membuat <i>Event</i>	Membuat <i>event</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>	Sesuai
		Membuat <i>event</i> dengan mengisi semua <i>field</i>	<i>Event</i> berhasil dibuat dan memunculkan pesan berhasil	Sesuai
10	Mengakses <i>My Event</i>	Menguji fungsional <i>My Event</i>	Menampilkan <i>event</i> yang sudah dibuat	Sesuai
		Mengedit <i>event</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan <i>error</i> untuk mengisi semua <i>field</i>	Sesuai
		Mengedit <i>event</i> dengan mengisi semua <i>field</i>	<i>Event</i> berhasil diedit dan memunculkan pesan berhasil diedit	Sesuai
		Mengklik <i>event</i> dan diarahkan ke detail <i>event</i>	Diarahkan ke detail <i>event</i>	Sesuai
11	Mengakses <i>My Ticket</i>	Menguji fungsional <i>My Ticket</i>	Menampilkan <i>event</i> yang sudah dijoin.	Sesuai

		Menguji <i>download</i> tiket <i>event</i>	Menampilkan tiket <i>event</i> dan bisa didownload	Sesuai
		Mengklik <i>event</i> dan diarahkan ke detail <i>event</i>	Diarahkan ke detail <i>event</i>	Sesuai
12	Mengakses <i>My Account</i>	Menguji fungsional <i>My Account</i>	Menampilkan informasi tentang user	Sesuai
		Mengedit salah satu <i>field</i>	Berhasil diedit dan memunculkan informasi baru	Sesuai
13	Mengakses <i>Password</i>	Menguji fungsional <i>Password</i>	Menampilkan halaman <i>password</i>	Sesuai
		Mengubah <i>password</i> dengan tidak mengisi salah satu <i>field</i>	Menampilkan pesan <i>error</i> untuk mengisi semua <i>field</i>	Sesuai
		Mengubah <i>password</i> namun <i>password</i> lama salah	Menampilkan pesan <i>error</i> bahwa <i>password</i> yang lama salah	Sesuai
		Mengubah <i>password</i> namun <i>field new password</i> dan <i>confirmation</i> tidak cocok	Menampilkan pesan <i>error</i> bahwa <i>new password</i> dan <i>confirmation</i> tidak cocok	Sesuai
		Mengubah <i>password</i> dan semua <i>field</i> sesuai	Mengubah <i>password</i> lama ke <i>password</i> baru	Sesuai
14	<i>Logout</i>	Menguji tombol <i>logout</i>	<i>User</i> berhasil <i>logout</i> dan diarahkan ke halaman <i>login</i>	Sesuai

Pengujian *Black Box* Testing memiliki 14 skenario pengujian. Setiap skenario pengujian dapat memiliki lebih dari 1 butir uji.

Dari 14 skenario pengujian *Black Box* Testing yang dilakukan oleh pengembang aplikasi, semua fitur telah berjalan dengan baik sesuai dengan *requirement*. **Hasil dari skenario pengujian dapat disimpulkan bahwa 100% fitur telah sesuai dengan *requirement*.**

5.2.2 UAT (*User Acceptance Test*)

UAT (*User Acceptance Test*) adalah proses pengujian aplikasi yang dilakukan oleh *user*. Pengujian ini bertujuan untuk mengetahui apakah aplikasi sudah sesuai dengan *requirement*.

Table 22: Hasil UAT

No	Fitur	Deskripsi	Hasil				
			User 1	User 2	User 3	User 4	User 5
1	<i>Home</i>	<i>User</i> mengakses halaman <i>home</i> untuk melihat <i>event</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
2	<i>Search Event</i>	<i>User</i> mencari <i>event</i> berdasarkan nama <i>event</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
3	<i>Sort Event</i>	<i>User</i> mengurutkan <i>event</i> berdasarkan kategori yang tersedia	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
4	<i>Detail Event</i>	<i>User</i> bisa melihat informasi detail suatu <i>event</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
5	<i>Register</i>	<i>User</i> bisa membuat akun	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
6	<i>Login</i>	<i>User</i> bisa <i>login</i> menggunakan akun yang sudah dibuat	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
7	<i>Join Event</i>	<i>User</i> bisa <i>join event</i> dengan syarat harus <i>login</i> terlebih dahulu	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
8	<i>Dashboard</i>	<i>User</i> bisa mengakses halaman <i>dashboard</i> setelah <i>login</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
9	<i>Create Event</i>	<i>User</i> bisa membuat <i>event</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
10	<i>Edit Event</i>	<i>User</i> bisa mengedit <i>event</i> yang telah dibuat	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
11	<i>Ticket</i>	<i>User</i> mendownload tiket <i>event</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
12	<i>Account</i>	<i>User</i> bisa mengubah informasi akun	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
13	<i>Password</i>	<i>User</i> bisa mengubah <i>password</i>	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai

UAT memiliki 13 fitur yang diuji oleh *user*, di mana setiap fitur akan memiliki hasil sesuai atau tidak sesuai berdasarkan dari hasil pengujian aplikasi.

Dari 5 *user* yang melakukan UAT (*User Acceptance Test*), semua fitur yang diuji oleh *user* memiliki hasil sesuai. **Hasil dari UAT (*User Acceptance Test*) dapat disimpulkan bahwa 100% fitur telah sesuai dengan *requirement*.**

5.2.3 Hasil kuesioner

kuesioner berikut adalah hasil yang didapatkan dari 10 responden yang mengisi kuesioner untuk memberikan *feedback* pada aplikasi yang dibuat. Penilaian kuesioner menggunakan skala *likert* untuk mendapatkan persentase dari hasil kuesioner. Kuesioner dapat dijadikan bukti bahwa aplikasi yang sudah dibuat sudah sesuai dengan *requirement*. Kuesioner terdiri dari 7 pertanyaan umum yang sesuai dengan *requirement user*. Berikut adalah 7 pertanyaan dari kuesioner.

Table 23: Kuisisioner

No	Pertanyaan
Pertanyaan untuk Guest (Tamu)	
1	User dapat melihat dan mencari <i>Event</i> pada Aplikasi web
2	User dapat melakukan Register dan <i>Login</i> pada Aplikasi Web
Pertanyaan untuk User yang sudah login	
3	User dapat membuat <i>Event</i> melalui fitur <i>Create Event</i>
4	User dapat bergabung dengan <i>Event</i> melalui Fitur <i>Join Event</i>
5	User dapat mendownload tiket <i>Event</i> melalui fitur <i>My Ticket</i>
Pertanyaan Umum	
6	Fungsional web sudah berjalan dengan baik
7	Tampilan web sudah terlihat dengan baik

Berikut adalah skala *likert* yang digunakan sebagai acuan untuk mendapatkan persentase hasil kuesioner.

Table 24: Skala Likert

Variabel	Keterangan	Nilai
Sangat Setuju	SS	5
Setuju	S	4
Ragu-Ragu	RG	3
Tidak Setuju	TS	2
Sangat Tidak Setuju	STS	1

Kuesioner diberikan kepada 10 responden untuk mendapatkan jawaban terkait aplikasi yang sudah dibuat.

Berikut adalah hasil dari kuesioner yang diisi oleh 10 responden.

Table 25: Hasil Kuisisioner

No	Pertanyaan	Jawaban Responden					Total Responden
		SS	S	RG	TS	STS	
1	User dapat melihat dan mencari <i>Event</i> pada Aplikasi web	9	1				10
2	User dapat melakukan <i>Register</i> dan <i>Login</i> pada Aplikasi Web	9	1				10
3	User dapat membuat <i>Event</i> melalui fitur <i>Create Event</i>	7	3				10
4	User dapat Bergabung dengan <i>Event</i> melalui Fitur <i>Join Event</i>	9	1				10
5	User dapat mendownload <i>Ticket Event</i> melalui fitur <i>My Ticket</i>	8	2				10
6	Fungsional web sudah berjalan dengan baik	8	2				10
7	Tampilan web sudah terlihat dengan baik	4	5	1			10

Setelah mendapatkan jawaban dari 10 responden, langkah selanjutnya adalah hasil jawaban responden dikali dengan nilai skala *likert*.

Berikut adalah hasil perhitungan kuesioner yang dikali dengan nilai skala likert.

Table 26: Hasil Perhitungan Skala Likert

No	Pertanyaan	SS*5	S*4	RG*3	TS*2	STS*1	Jumlah Total
1	User dapat melihat dan mencari <i>Event</i> pada Aplikasi web	45	4	0	0	0	49
2	User dapat melakukan <i>Register</i> dan <i>Login</i> pada Aplikasi Web	45	4	0	0	0	49
3	User dapat membuat <i>Event</i> melalui fitur <i>Create Event</i>	35	12	0	0	0	47
4	User dapat Bergabung dengan <i>Event</i> melalui Fitur <i>Join Event</i>	45	4	0	0	0	49
5	User dapat mendownload <i>Ticket Event</i> melalui fitur <i>My Ticket</i>	40	8	0	0	0	48
6	Fungsional web sudah berjalan dengan baik	40	8	0	0	0	48
7	Tampilan web sudah terlihat dengan baik	20	20	3	0	0	43

Setelah mendapatkan jumlah total yang telah dikali dengan skala *likert*, maka selanjutnya adalah mencari rata-rata keseluruhan dari hasil kuesioner.

Untuk mendapatkan hasil interpretasi maka rumus yang digunakan adalah:

$$\text{Jumlah Total} / (\text{Nilai Skala Likert Tertinggi} * \text{Jumlah Responden}) * 100.$$

Maka didapatkan hasil rata-rata keseluruhan kuesioner menggunakan skala *likert*.

Table 27: Hasil Rata-Rata Keseluruhan

No	Pertanyaan	Jumlah Total	Hasil Interpretasi (JT / 50 * 100)
1	User dapat melihat dan mencari <i>Event</i> pada Aplikasi web	49	98
2	User dapat melakukan <i>Register</i> dan <i>Login</i> pada Aplikasi Web	49	98
3	User dapat membuat <i>Event</i> melalui fitur <i>Create Event</i>	47	94
4	User dapat Bergabung dengan <i>Event</i> melalui Fitur <i>Join Event</i>	49	98
5	User dapat mendownload <i>Ticket Event</i> melalui fitur <i>My Ticket</i>	48	96
6	Fungsional web sudah berjalan dengan baik	48	96
7	Tampilan web sudah terlihat dengan baik	43	86
Jumlah Persentase			666
Rata-rata Keseluruhan			95%

Berdasarkan hasil dari kuesioner yang telah dilakukan maka didapatkan rata-rata keseluruhan dari Aplikasi yang sudah dibuat. **Hasil rata-rata keseluruhan yang didapatkan dari kuesioner sebesar 95%. Maka dari hasil tersebut dapat disimpulkan bahwa aplikasi sudah sesuai dengan *requirement*.**

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini berisi mengenai penarikan kesimpulan dan saran dari penulis terhadap keseluruhan proses yang telah dilakukan guna mendapatkan pengembangan aplikasi yang lebih baik.

6.1 Kesimpulan

Berdasarkan hasil pengujian **Rancang Bangun Sistem Informasi Event Keagamaan Berbasis Web Menggunakan *Framework Laravel*** pada penelitian ini maka didapatkan kesimpulan sebagai berikut:

1. Aplikasi sistem informasi event keagamaan yang dikembangkan berbasis web menggunakan *Framework Laravel* mampu menyajikan *event* keagamaan.
2. Aplikasi sistem informasi keagamaan memiliki fitur: melihat event, join event, mengelola event, ticket, mengelola akun, login dan register , dashboard user.
3. Aplikasi sistem informasi event keagamaan sudah sesuai dengan *requirement* berdasarkan hasil rata-rata keseluruhan yang didapatkan dari *Black Box Testing*, UAT (*User Acceptance Test*), dan kuesioner.

6.2 Saran

Penelitian ini masih terdapat kekurangan sehingga masih memerlukan pengembangan lebih lanjut, terdapat beberapa saran yang penulis ajukan untuk pengembangan penelitian yang akan datang sebagai berikut:

1. Aplikasi *event* keagamaan dapat dibuatkan untuk versi *mobile*.
2. Pembuatan *web service* agar dapat digunakan oleh *mobile client*.
3. Dilakukan pengujian *Responsive Web Design* terutama pada tampilan *mobile version* berdasarkan hasil dari nilai terendah kuesioner yaitu tampilan website.
4. Aplikasi dapat dikembangkan lebih lanjut terutama dari segi tampilan serta penambahan fitur yang belum terdapat di aplikasi saat ini.

DAFTAR PUSTAKA

- 1: Sareh Aghaei, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani, Evolution of the World Wide Web from Web 1.0 to Web 4.0, 2012
- 2: Internet Live Stats, Total number of Websites, 2018, <http://www.internetlivestats.com/total-number-of-websites/>
- 3: Kenneth C. Laudon, Jane P. Laudon, Management Information Systems: Managing The Digital Firm, 2012
- 4: James A. O'Brien, George Marakas, Management Information Systems, 2011
- 5: MARTIN BJÖREMO, PREDRAG TRNINIĆ, Evaluation of web application frameworks, 2010
- 6: Chanchai Supaartagorn, PHP Web Authoring for Database Management based on MVC Pattern, 2011
- 7: Pastima Simanjuntak, Arwin Kasnady, ANALISIS MODEL VIEW CONTROLLER (MVC) PADA BAHASA PHP, 2016
- 8: Tutorials Point, Laravel Tutorial, 2016
- 9: MySQL, What is MySQL, 2018, <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- 10: Sri Dharwiyanti, Pengantar Unified Modeling Language (UML), 2003
- 11: Mingtao Shi, Software Functional Testing from the Perspective of Business Practice, 2010
- 12: Mohd. Ehmer Khan, Different Approaches To Black Box Testing Technique For Finding Errors, 2011
- 13: Diedit, Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya, 2018
- 14: Obeth Dewa Wardhana, Pembuatan Sistem Informasi Registrasi Event Dan Ticketing “LAGIRAME”, 2017
- 15: Nanang Wisnu Pambudi, Pengembangan Sistem Informasi Event Mahasiswa Berbasis Web Di Universitas Negeri Yogyakarta, 2017
- 16: Khoirunnisa Septiani, Sistem Informasi Kajian Dan Event Kerohanian Islam Berbasis Web, 2016

LAMPIRAN

Lampiran 1: Hasil UAT

No	Fitur	Deskripsi	Hasil				
			User 1	User 2	User 3	User 4	User 5
1	Home	User mengakses halaman home untuk melihat event	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
2	Search Event	User mencari event berdasarkan nama event	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
3	Sort Event	User mengurutkan event berdasarkan kategori yang tersedia	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
4	Detail EEvent	User bisa melihat informasi detail suatu event	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
5	Register	User bisa membuat akun	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
6	Login	User bisa login menggunakan akun yang sudah dibuat	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
7	Join Event	User bisa join event dengan syarat harus login terlebih dahulu	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
8	Dashboard	User bisa mengakses halaman dashboard setelah login	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
9	Create Event	User bisa membuat event	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
10	Edit Event	User bisa mengedit event yang telah dibuat	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
11	Ticket	User mendownload ticket event	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
12	Account	User bisa mengubah informasi akun	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai
13	Password	User bisa mengubah password	Sesuai	Sesuai	Sesuai	Sesuai	Sesuai

Lampiran 2: Hasil Kuesioner

Nama	Jurusan	User dapat melihat dan mencari Event pada Aplikasi web	User dapat melakukan Register dan Login pada Aplikasi Web	User dapat membuat Event melalui fitur Create Event	User dapat Bergabung dengan Event melalui Fitur Join Event	User dapat mendownload Ticket Event melalui fitur My Ticket	Fungsional web sudah berjalan dengan baik	Tampilan web sudah terlihat dengan baik
Wahyu	Sistem Informasi	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
Alfian Rahmatullah	Teknik Informatika	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Ragu-Ragu
Dzaki Mahfuzh Hamadah	Teknik Informatika	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Setuju
Choniyu Azwan	Teknik Informatika	Sangat Setuju	Sangat Setuju	Setuju	Sangat Setuju	Setuju	Sangat Setuju	Sangat Setuju
Muhammad Luqni Baehaqi	Sistem Informasi	Setuju	Setuju	Setuju	Setuju	Sangat Setuju	Sangat Setuju	Setuju
Rizka Amalia	Sistem Informasi	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Setuju
Muh Isfhani Ghiath	Teknik Informatika	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Setuju	Sangat Setuju
Yasir	Teknik Informatika	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Setuju
Ihsanul Fikri Abiyyu	Teknik Informatika	Sangat Setuju	Sangat Setuju	Setuju	Sangat Setuju	Setuju	Setuju	Setuju
Rizky Hidayat Panjaitan	Teknik Informatika	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju