**TUGAS AKHIR**

**PERANCANGAN SISTEM INFORMASI LAYANAN JASA *EVENT ORGANIZER* BERBASIS WEB**

Ditulis untuk Memenuhi Sebagian Persyaratan Akademik

Untuk Memperoleh Gelar Strata Satu Sarjana Komputer (S.Kom.)

**Oleh:**

|  |  |  |
| --- | --- | --- |
| **NAMA** | **:** | **Irene Andriani** |
| **NIM** | **:** | **00000013440** |



**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PELITA HARAPAN**

**TANGERANG**

**2019**

|  |  |
| --- | --- |
|  | PERNYATAAN KEASLIAN KARYA TUGAS AKHIR |

Saya mahasiswa Fakultas Ilmu Komputer, Universitas Pelita Harapan dengan informasi di bawah ini:

**Nama**  : **<Nama Lengkap>**

**Nomor Induk Mahasiswa** : **<NIM>**

**Program Studi** : Sistem Informasi

Dengan ini menyatakan bahwa tugas akhir dengan judul:

**<JUDUL>**

1. Dibuat dan diselesaikan sendiri, dengan mengacu pada beberapa referensi (jurnal ilmiah, buku, konferensi, karya ilmiah, dan sumber referensi terpercaya lainnya) dan didukung dengan tinjauan lapangan.
2. Bukan merupakan duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai untuk mendapatkan gelar sarjana di universitas lain, kecuali pada bagian-bagian sumber informasi dicantumkan dengan cara referensi yang semestinya.
3. Bukan merupakan karya terjemahan dari kumpulan buku atau jurnal acuan yang tertera di dalam referensi pada karya tugas akhir saya.

Kalau terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka tugas akhir ini batal.

Karawaci, <dd mmmm yyyy«Tgl\_buat\_SC»>

Yang membuat pernyataan,

Materai

6000

<Nama Lengkap>

|  |  |
| --- | --- |
|  | **UNIVERSITAS PELITA HARAPAN**  **FAKULTAS ILMU KOMPUTER** |

# PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR

**<JUDUL>**

Oleh :

**Nama : <Nama Lengkap>**

**NIM : <NIM>**

**Program Studi : Sistem Informasi**

**Peminatan : *<*Peminatan*>***

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam Sidang Tugas Akhir guna memperoleh gelar Strata Satu Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten.

**Tangerang, <dd mmmm yyyy>**

**(isi dengan tanggal pada waktu disetujui untuk maju sidang tugas akhir)**

**Menyetujui :**

**Pembimbing Pertama Pembimbing Kedua**

**( ) ( )**

**Ketua Program Studi Pembantu Dekan**

**Sistem Informasi Fakultas Ilmu Komputer**

**( ) ( )**

|  |  |
| --- | --- |
|  | **UNIVERSITAS PELITA HARAPAN**  **FAKULTAS ILMU KOMPUTER** |

# PERSETUJUAN TIM PENGUJI TUGAS AKHIR

Pada hari <hari>, <dd mmmm yyyy> telah diselenggarakan Sidang Tugas Akhir mahasiswa/i dengan keterangan:

**Nama : <Nama Lengkap>**

**NIM : <NIM>**

**Judul : <JUDUL>**

Sebagai persyaratan akademik untuk mencapai gelar Strata Satu Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten, yang diuji oleh tim penguji yang terdiri dari:

|  |  |  |
| --- | --- | --- |
| **Nama** | **Status** | **Tanda tangan** |
|  |  |  |
| <Nama Dosen> | sebagai ketua sidang | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| <Nama Dosen> | sebagai anggota | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| <Nama Dosen> | sebagai anggota | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# ABSTRAK

<Nama Lengkap> (<NIM>)

**<JUDUL>**

(<jumlah halaman sebelum pendahuluan dalam romawi> + <jumlah halaman> halaman: <jumlah gambar> gambar, <jumlah tabel> tabel, <jumlah lampiran> lampiran)

<Times New Roman, spasi 1pt, maksimum 250 kata>

**Kata kunci:** <maksimum 7 kata kunci dipisahkan dengan koma>

**Referensi:** <Jumlah referensi> (<tahun referensi terlama – tahun referensi terbaru>)

# ABSTRACT

<Full Name> (<Student ID>)

**<TITLE>**

(<total of pages before chapter 1> + <total of pages> pages: <total of figures> figures, <total of tables> tables, <total of appendix/appendices> appendices)

<Times New Roman, single spacing, max 250 words>

*Keywords*: <max 7 words>

References: <total of references> (<years of references>)

# KATA PENGANTAR

<Times New Roman, spasi 1.5pt>

Karawaci, <dd mmmm yyyy>

Penulis

# DAFTAR ISI

[PERNYATAAN KEASLIAN KARYA TUGAS AKHIR ii](#_Toc505241733)

[PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR iii](#_Toc505241734)

[PERSETUJUAN TIM PENGUJI TUGAS AKHIR iii](#_Toc505241735)

[ABSTRAK iii](#_Toc505241736)

[ABSTRACT iii](#_Toc505241737)

[KATA PENGANTAR iii](#_Toc505241738)

[DAFTAR ISI iii](#_Toc505241739)

[DAFTAR GAMBAR iii](#_Toc505241740)

[DAFTAR TABEL iii](#_Toc505241741)

[DAFTAR LAMPIRAN iii](#_Toc505241742)

[BAB I PENDAHULUAN 3](#_Toc505241743)

[1.1 Latar Belakang 3](#_Toc505241744)

[1.2 Perumusan Masalah 3](#_Toc505241745)

[1.3 Pembatasan Masalah 3](#_Toc505241746)

[1.4 Tujuan dan Manfaat Penelitian 3](#_Toc505241747)

[1.4.1 Tujuan Penelitian 3](#_Toc505241748)

[1.4.2 Manfaat Penelitian 3](#_Toc505241749)

[1.5 Metodologi Penelitian 3](#_Toc505241750)

[1.5.1 Metode Pengumpulan Data 3](#_Toc505241751)

[1.5.2 Metodologi Pengembangan Sistem 3](#_Toc505241752)

[1.6 Sistematika Penulisan 3](#_Toc505241753)

[BAB II LANDASAN TEORI 3](#_Toc505241754)

[2.1 Theory 1 3](#_Toc505241755)

[2.1.1 Theory 1.1 3](#_Toc505241756)

[BAB III 3](#_Toc505241757)

[SISTEM SAAT INI 3](#_Toc505241758)

[3.1 Profil Perusahaan 3](#_Toc505241759)

[3.1.1 Visi Perusahaan 3](#_Toc505241760)

[3.1.2 Misi Perusahaan 3](#_Toc505241761)

[3.2 Struktur Organisasi 3](#_Toc505241762)

[3.3 Analisis Sistem Saat ini 3](#_Toc505241763)

[3.3.1 Kendala Sistem Saat ini 3](#_Toc505241764)

[BAB IV ANALISIS DAN PERANCANGAN SISTEM USULAN 3](#_Toc505241765)

[4.1 Analisis Kelayakan Sistem 3](#_Toc505241766)

[4.2 Tahap Analisis 3](#_Toc505241767)

[4.2.1 Sistem Usulan (Target Sistem) 3](#_Toc505241768)

[BAB V KESIMPULAN DAN SARAN 3](#_Toc505241769)

[5.1 Kesimpulan 3](#_Toc505241770)

[5.2 Saran 3](#_Toc505241771)

[DAFTAR PUSTAKA 3](#_Toc505241772)

[LAMPIRAN A HASIL WAWANCARA A-3](#_Toc505241773)

# DAFTAR GAMBAR

[Gambar 2.1 < Judul gambar Times New Roman 10, diletakkan di bawah gambar, diawali dengan nomor bab, align: center > 3](#_Toc496534092)

Penulisan daftar gambar / tabel / lampiran menggunakan font Times New Roman 10pt, spasi 1. Gunakan *tabel of content* jangan diketik manual.

# DAFTAR TABEL

[[Tabel 2.1 <Judul tabel Times New Roman 10, diletakkan di atas tabel, diawali dengan nomor bab, align: center> 3](#_Toc496526847)](#_Toc496534120)

# DAFTAR LAMPIRAN

[LAMPIRAN A HASIL WAWANCARA A-3](#_Toc496526954)

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Seiring dengan berkembangnya teknologi dan komunikasi pada era globalisasi ini, internet memegang peran yang penting dalam kehidupan sehari-hari. Dengan adanya internet sebagai sumber informasi yang luas, segala sesuatu pun dapat diperoleh dengan praktis dan cepat. Selain memperoleh informasi, internet juga banyak dimanfaatkan sebagai sarana untuk berbisnis. Banyak orang yang menjual berbagai macam produk ataupun menyediakan berbagai macam jasa melalui internet; salah satu contohnya adalah jasa *event organizer*.

*Event Organizer* (EO) dapat didefinisikan sebagai sebuah profesi yang dijalankan oleh sekelompok orang dengan tugasnya masing-masing yang telah ditetapkan dalam rangka mencapai satu tujuan tertentu. Tujuan tersebut yaitu menghasilkan acara yang sesuai dengan permintaan klien. EO sendiri terdiri dari berbagai macam jenis; ada yang bergerak di bidang olahraga, seni, pameran, maupun acara yang bersifat pribadi seperti pesta pernikahan, ulang tahun, dan acara lainnya [1]. Melihat perkembangan EO yang semakin pesat, peluang untuk melakukan bisnis ini pun cukup terbuka. Jasa-jasa EO pun semakin dibutuhkan, terutama bagi orang-orang di zaman sekarang yang ingin semuanya serba praktis dan tidak ingin mengurusi hal-hal yang rumit.

Dalam penyelenggaraan sebuah acara, banyak hal yang perlu dipertimbangkan seperti lokasi, anggaran, dekorasi, dan hal lainnya yang berkaitan dengan acara. Penggunaan jasa EO yang sesuai sangatlah penting agar acara dapat berjalan dengan lancar dan sukses. Berdasarkan hasil survei yang telah dilakukan terhadap 111 responden dari berbagai provinsi di Indonesia, 36,9% di antaranya pernah menggunakan jasa EO namun 21,6% di antaranya masih menghadapi kesulitan dalam mencari jasa EO. Kesulitan tersebut dikarenakan penyediaan informasi yang masih tergolong minim sehingga sulit mencari EO yang terpercaya dan sesuai dengan acara, tema, dan *budget* (Lampiran 1). Bagi para EO yang baru memulai bisnisnya, tentu juga mengalami kesulitan dalam mendapatkan kepercayaan klien untuk menggunakan jasanya.

Melihat permasalahan yang ada, riset terhadap media pun dilakukan dan didapatkan beberapa *website* yang secara spesifik hanya menampilkan informasi tentang satu EO. Hal tersebut menyebabkan proses pencarian jasa EO menjadi kurang efektif dimana proses tersebut dilakukan satu per satu melalui internet maupun di berbagai media sosial. Ditemukan juga sebuah *website* bernama sejasa.com yang menyediakan berbagai macam jasa termasuk EO, namun fitur yang disediakan kurang membantu dimana klien harus menjawab pertanyaan yang telah disediakan terlebih dahulu dan kemudian menunggu para EO untuk menghubunginya. Selain itu, masih belum ditemukan sebuah sistem yang secara komprehensif menampilkan informasi mengenai berbagai jasa EO yang ada di Indonesia.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, maka penelitian ini memiliki rumusan masalah sebagai berikut:

1. Para klien yang mengalami kesulitan dalam mencari jasa EO dikarenakan sistem pencariannya yang masih dilakukan satu per satu melalui internet maupun di berbagai media sosial.
2. Penyediaan informasi mengenai EO yang ada di Indonesia masih kurang lengkap dan jelas seperti informasi harga dari paket yang disediakan, *portfolio* dari acara yang pernah diselenggarakan, dan juga testimoni dari klien yang pernah menggunakan jasanya.
3. Banyaknya EO yang dapat ditemukan di internet dengan berbagai macam kategori dan jasa yang disediakan sehingga menimbulkan keraguan bagi klien dalam memilih jasa EO yang tepat.
4. Para EO yang mengalami kesulitan pada tahap awal untuk mempublikasikan jasa yang disediakannya.

## 1.3 Pembatasan Masalah

Penelitian ini memiliki batasan-batasan masalah sebagai berikut:

1. Aplikasi ini berbasis *web*.
2. Aplikasi yang dirancang akan menampilkan informasi yang lengkap mengenai jasa-jasa yang disediakan oleh berbagai EO yang ada di Indonesia (*profile* perusahaan, harga dari paket yang disediakan, *portfolio* dari acara yang pernah diselenggarakan, testimoni dari klien yang pernah menggunakan jasanya, lokasi beserta informasi kontak).
3. Aplikasi ini dapat menjadi perantara yang mempermudah interaksi antar para EO dan klien.
4. Aplikasi ini dapat membantu pengambilan keputusan dalam pemilihan EO yang sesuai dengan preferensi klien.
5. Aplikasi tidak mencakup proses pembayaran.

## 1.4 Tujuan dan Manfaat Penelitian

Berdasarkan latar belakang yang telah dijabarkan, maka penelitian ini memiliki tujuan dan manfaat sebagai berikut:

### 1.4.1 Tujuan Penelitian

Berikut tujuan dari penelitian ini:

1. Menghasilkan aplikasi berbasis *web* yang dapat menyediakan informasi mengenai jasa-jasa dari berbagai EO yang ada di Indonesia sehingga dapat mempermudah klien dalam mencari jasa EO.
2. Menghasilkan aplikasi yang dapat membantu klien dalam memilih EO yang tepat dengan memberikan rekomendasi EO yang paling sesuai dengan preferensi klien.
3. Menghasilkan aplikasi yang dapat menjadi sarana bagi para EO untuk mempublikasikan jasa yang disediakannya.

### 1.4.2 Manfaat Penelitian

Berikut manfaat dari penelitian ini:

1. Aplikasi dapat mempermudah klien dalam melihat dan memperoleh informasi yang lengkap dan jelas mengenai jasa yang disediakan oleh berbagai EO.
2. Aplikasi dapat membantu mempertemukan klien dengan EO yang tepat sehingga peluang kesuksesan dari acara yang ingin diselenggarakan dapat semakin meningkat.
3. Aplikasi dapat menjadi sarana untuk mempublikasikan jasa EO yang ada sehingga dapat dikenal secara lebih luas.
4. Dengan adanya aplikasi ini dapat meningkatkan perkembangan bisnis EO di Indonesia.

## 1.5 Metodologi Penelitian

Berikut adalah beberapa metode penelitian yang akan digunakan dalam melakukan penelitian ini:

### 1.5.1 Metode Pengumpulan Data

Berikut metode pengumpulan data yang akan digunakan dalam penelitian ini:

1. Observasi

Observasi dilakukan untuk mengamati secara langsung bagaimana proses pencarian dan penggunaan layanan jasa EO pada saat ini.

1. Studi Literatur

Studi literatur dilakukan untuk mencari referensi mengenai *website* atau hasil penelitian sebelumnya dari berbagai macam literatur, dan juga mempelajari teori-teori yang berkaitan dengan *tools* yang dapat digunakan dalam menunjang perancangan sistem.

1. Kuesioner

Kuesioner akan dibuat dan kemudian disebarkan kepada beberapa orang untuk mengetahui respon mereka terhadap adanya sistem yang dapat membantu pencarian jasa EO.

### 1.5.2 Metodologi Pengembangan Sistem

Pengembangan sistem dalam penelitian ini akan menggunakan metode *Rapid Application Development* (RAD) atau *Rapid Prototyping*. RAD merupakan sebuah model yang melibatkan pengembangan dan pembangunan *prototype iterative* dimana proses perancangan dan perbaikan sistem dapat dilakukan secara berulang karena adanya respon dari *user.* Metode ini dipilih karena adanya keterbatasan waktu untuk melaksanakan perancangan sistem, selain itu keterlibatan *user* juga dibutuhkan untuk pengembangan sistem yang lebih baik.

## 1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini akan memuat uraian secara garis besar dari isi penelitan yang terdiri dari lima bab dan dilengkapi dengan beberapa lampiran. Berikut adalah penjabaran dari masing-masing bab:

**BAB I PENDAHULUAN**

Dalam bab ini akan menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metode penelitian yang akan digunakan, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Dalam bab ini akan membahas tentang berbagai teori dan definisi-definisi yang menjadi dasar dalam proses penyusunan penelitian ini.

**BAB III SISTEM SAAT INI**

Dalam bab ini akan dijelaskan gambaran dari analisis sistem saat ini beserta kendala-kendala yang dihadapi.

**BAB IV SISTEM USULAN**

Dalam bab ini akan menjelaskan tentang analisis kelayakan sistem yang diusulkan, pemodelan dari sistem usulan, tahapan implementasi dan pengujian sistem.

**BAB V KESIMPULAN DAN SARAN**

Dalam bab ini akan dipaparkan kesimpulan dari penelitian yang dilakukan dan juga saran-saran yang dapat menjadi bahan pertimbangan untuk pengembangan sistem yang lebih baik.

# BAB II LANDASAN TEORI

## 2.1 *System Development Life Cycle* (SDLC)

SDLC merupakan sebuah metodologi yang digunakan untuk mengembangkan sistem informasi yang dapat mendukung kebutuhan bisnis, melakukan desain terhadap suatu sistem, membangun sistem tersebut, dan menyampaikan sistem yang telah dirancang kepada *user* [2]. SDLC terdiri dari beberapa fase yang dimulai dari fase perencanaan (*planning*), fase analisis (*analysis*), fase desain (*design*), dan fase implementasi (*implementation*).

### 2.1.1 Planning

Fase perencanaan atau *planning* merupakan proses untuk memahami mengapa sistem informasi tersebut harus dibangun dan menentukan bagaimana cara pelaksanaannya yang tepat [2]. Pada fase ini, dilakukan proses identifikasi terhadap nilai bisnis (*business value*), dan kebutuhan-kebutuhan bisnis (*business requirements*) yang dapat mendukung pembangunan sistem sehingga sistem yang akan dibangun dapat berguna bagi organisasi. Proses analisis kelayakan (*feasibility analysis*) juga dilakukan pada fase ini dimana proses tersebut mencakup masalah teknis, ekonomi, dan organisasi yang dapat menentukan kelangsungan proyek.

### 2.1.2 Analysis

Fase analisis atau *analysis* merupakan proses untuk mengetahui siapa yang akan menggunakan sistem, apa saja yang dapat dilakukan oleh sistem, dimana dan kapan sistem tersebut dapat digunakan [2]. Pada fase ini, dilakukan analisa terhadap kondisi sistem yang sedang berjalan pada saat ini (*as-is system*), mengidentifikasi masalah-masalah yang terdapat di dalamnya, dan kemudian mengembangkan sebuah konsep yang dapat digunakan untuk membangun sistem yang baru (*to-be system*).

### 2.1.3 Design

Fase desain atau *design* merupakan proses untuk menentukan bagaimana sistem akan beroperasi dengan melihat spesifikasi dalam sistem seperti perangkat keras (*hardware)*, perangkat lunak (*software*), jaringan infrastruktur (*network infrastructure*), tampilan antarmuka (*user interface*), program, *database* dan data-data lain yang diperlukan [2]. Pada fase ini, akan ditentukan siapakah yang membangun sistem dan kemudian dilanjutkan dengan pengembangan desain arsitektur sistem yang mencakup perangkat keras, perangkat lunak, dan jaringan infrastruktur yang digunakan. Selain itu, desain antarmuka dilakukan untuk menentukan bagaimana *user* akan menggunakan sistem yang dibangun. Penentuan akan dimana dan apa saja data yang akan disimpan juga dilakukan pada fase ini.

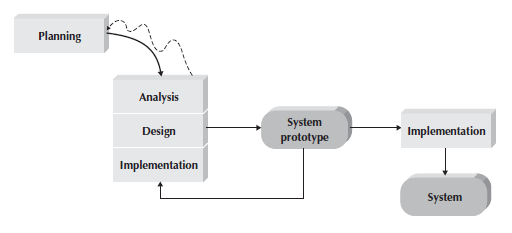
### 2.1.4 Implementation

Fase implementasi atau *implementation* adalah fase terakhir pada SDLC dan merupakan proses dimana sistem telah dibangun [2]. Pada fase ini, sistem yang dibangun akan diuji terlebih dahulu untuk memastikan bahwa sistem dapat bekerja sesuai dengan yang direncanakan. Setelah pengujian sistem, maka akan dilakukan proses instalasi dimana sistem yang lama akan dimatikan dan diganti dengan sistem yang baru dengan menggunakan metode konversi yang ditentukan oleh organisasi.

## 2.2 *Rapid Application Development* (RAD)

Metodologi yang akan digunakan untuk mengembangkan sistem pada penelitian ini adalah RAD. RAD memfokuskan pada proses pengembangan sistem yang dilakukan secara lebih cepat dan kemudian disampaikan kepada *user* [2]. Dengan hal ini, dapat meningkatkan pengertian *user* akan sistem yang dirancang. RAD terbagi lagi menjadi tiga metode yaitu *phased development*, *prototyping* dan *throwaway prototyping*.

Proses perancangan sistem dalam penelitian ini akan menggunakan metode *prototyping*. Dalam metode ini, akan dilakukan proses analisis, desain, dan implementasi secara bersamaan. Ketiga proses tersebut akan dilakukan secara berulang dalam suatu siklus hingga sistem dapat terselesaikan secara menyeluruh [2]. Metode ini memiliki kelebihan yaitu penyediaan sistem yang dapat berinteraksi dengan *user* dilakukan dalam waktu yang singkat, meskipun sistem masih belum dapat digunakan secara luas. Dengan metode ini, *user* juga dapat memastikan pengembangan sistem telah dilakukan dengan baik sehingga tidak ada lagi penundaan dalam pelaksanaan proyek. Gambar 2.1 berikut ini akan menunjukkan bagaimana penerapan dengan menggunakan metode *prototyping*.

[](#_DAFTAR_GAMBAR)

Gambar 2.1 Metode *Prototyping*

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis & Design: An Object-Oriented Approach with UML, 5th Edition, USA: Wiley, 2015

## 2.3 *Unified Modelling Language* (UML)

UML adalah sebuah bahasa pemodelan secara visual yang digunakan dalam pengembangan sistem berorientasi objek (*object-oriented*). UML menyediakan berbagai aturan umum mengenai teknik perancangan sistem yang akan digambarkan melalui diagram. Dengan adanya teknik penggambaran diagram, dapat mempermudah proses pemodelan pengembangan sistem dari tahap analisis hingga tahap implementasi [2].

UML mendefinisikan lima belas teknik penggambaran diagram yang digunakan untuk memodelkan suatu sistem, yang kemudian dibagi menjadi dua kategori, yaitu:

1. *Structure diagram;*

Diagram ini digunakan untuk menggambarkan struktur dari suatu sistem, yang terdiri dari *class*, *object*, *package*, *deployment*, *component*, *composite structure*, dan *profile diagram*.

1. *Behavior diagram;*

Diagram ini digunakan untuk menggambarkan perilaku dari suatu sistem, yang terdiri dari *activity*, *sequence*, *communication*, *interaction overview*, *timing*, *behavior state machine*, *protocol state machine*, dan *use-case diagram*.

Teknik pemodelan yang akan digunakan untuk proses pengembangan sistem dalam penelitian ini adalah *use-case diagram*, *activity diagram* dan *class diagram*.

### 2.3.1 Pemodelan Fungsional

Pemodelan fungsional ini menggambarkan proses dari suatu bisnis dan interaksi antar sistem informasi dengan lingkungannya. Dalam pengembangan sistem berorientasi objek, terdapat dua jenis model yang digunakan untuk menggambarkan fungsionalitas dari suatu sistem informasi, yaitu *use-case* dan *activity diagram* [2].

1. *Use-case diagram;*

*Use-case diagram* merupakan diagram yang menggambarkan secara sederhana fungsi-fungsi utama yang ada dalam sistem dan bagaimana *user* atau pengguna lainnya yang terkait dapat berinteraksi dengan sistem tersebut. Penggunaan *use-case diagram* ini memudahkan pengidentifikasian fitur-fitur yang diinginkan oleh *user* sehingga tercipta *requirements* yang dapat menunjang perancangan sistem. Elemen-elemen dalam *use-case diagram* dapat dilihat pada tabel 2.1 berikut ini.

Tabel 2.1 Elemen-elemen *Use-case diagram*

|  |  |  |
| --- | --- | --- |
| Nama Elemen | Fungsi | Notasi |
| *Actor* | Menggambarkan tokoh atau sistem yang memperoleh manfaat dan berada di luar subyek. *Actor* dapat diasosiasikan dengan *actor* lain menggunakan *specialization* atau *superclass association*. |  |
| *Use case* | Merepresentasikan fungsi utama dari sistem. |  |
| *Subject boundary* | Merepresentasikan ruang lingkup dari subyek. |  |
| *Association relationship* | Menghubungkan *actor* dengan satu atau banyak *use case*. |  |
| *Include relationship* | Merepresentasikan penyertaan fungsi dari satu *use case* dengan *use case* lain. |  |
| *Extend relationship* | Merepresentasikan perluasan dari satu use case dengan perilaku yang bersifat *optional*. |  |
| *Generalization relationship* | Merepresentasikan *specialized use case* dengan *use case* yang lebih umum. |  |

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis & Design: An Object-Oriented Approach with UML, 5th Edition, USA: Wiley, 2015

1. *Activity diagram;*

*Activity diagram* merupakan diagram yang memodelkan perilaku dalam suatu proses bisnis yang terlepas dari obyek. Diagram ini akan menggambarkan kegiatan utama dan hubungan antar kegiatan-kegiatan tersebut dalam suatu proses. *Activity diagram* digunakan untuk mengetahui kegiatan apa saja yang harus dilakukan oleh *actor* atau *user* dalam suatu sistem. Elemen-elemen dalam *activity diagram* dapat dilihat pada tabel 2.2 berikut ini.

Tabel 2.2 Elemen-elemen *Activity diagram*

|  |  |  |
| --- | --- | --- |
| Nama Elemen | Fungsi | Notasi |
| *Action* | Merepresentasikan suatu perilaku yang sederhana. |  |
| *Activity* | Merepresentasikan suatu kegiatan. |  |
| *Object node* | Merepresentasikan objek yang terhubungan dengan serangkaian *object flow*. |  |
| *Control flow* | Menggambarkan urutan dari eksekusi kegiatan. |  |
| *Object flow* | Menggambarkan alur suatu objek dari satu kegiatan ke kegiatan lain. |  |
| *Initial node* | Menggambarkan awal dimulainya sebuah kegiatan. |  |
| *Final-activity node* | Menghentikan semua *control flow* atau *object flow* dalam suatu kegiatan. |  |
| *Final-flow node* | Menghentikan *specific control flow* atau *object flow*. |  |
| *Decision node* | Merepresentasikan suatu kondisi untuk memastikan *control flow* atau *object flow* menuju pada satu jalur. |  |
| *Merge node* | Mengumpulkan kembali *decision path* yang berbeda dengan menggunakan *decision node*. |  |
| *Fork node* | Membagi perilaku menjadi serangkaian alur kegiatan yang berjalan secara paralel atau bersamaan. |  |

Tabel 2.3 Elemen-elemen *Activity diagram* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| *Join node* | Mengumpulkan kembali serangkaian alur kegiatan yang berjalan secara paralel atau bersamaan. |  |
| *Swimlane* | Memecah sebuah activity diagram ke dalam baris dan kolom untuk menetapkan suatu kegiatan individual kepada individu atau objek yang bertanggung jawab. |  |

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis & Design: An Object-Oriented Approach with UML, 5th Edition, USA: Wiley, 2015

### 2.3.2 Pemodelan Struktural

Pemodelan struktural ini menggambarkan objek yang akan digunakan dan diciptakan oleh suatu sistem bisnis dengan mengilustrasikan orang, tempat, atau informasi-informasi yang disimpan dan bagaimana hal-hal tersebut terkait satu sama lain. Tujuan pemodelan struktural ini adalah untuk menciptakan istilah-istilah yang dapat digunakan oleh *system analyst* dan *user* agar dapat berkomunikasi secara lebih efektif [2]. Salah satu pemodelan struktur yang akan digunakan dalam penelitian ini adalah *class diagram*.

1. *Class diagram;*

*Class diagram* merupakan sebuah model statis yang memperlihatkan *class* dan hubungan antar *class* yang ditetapkan dalam sistem. Diagram ini akan menggambarkan *class*, beserta perilaku, status dan hubungan yang terjalin antar *class*. Elemen-elemen pada *class diagram* dapat dilihat pada tabel 2.3 berikut ini.

Tabel 2.4 Elemen-elemen *Class* *diagram*

|  |  |  |
| --- | --- | --- |
| Nama Elemen | Fungsi | Notasi |
| *Class* | Merepresentasikan suatu jenis orang, tempat, atau benda yang informasinya akan disimpan oleh sistem. |  |
| *Attribute* | Merepresentasikan *properties* yang mendeskripsikan keadaan suatu objek. |  |

Tabel 2.5 Elemen-elemen *Class* *diagram* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| *Operation* | Merepresentasikan tindakan atau fungsi yang dapat dilakukan oleh *class*. |  |
| *Association* | Merepresentasikan hubungan antara beberapa *class* atau sebuah *class* dengan dirinya sendiri. |  |
| *Generalization* | Merepresentasikan hubungan yang sejenis antara beberapa *class*. |  |
| *Aggregation* | Merepresentasikan bagian hubungan yang logis antara beberapa *class* atau sebuah *class* dengan dirinya sendiri. |  |
| *Composition* | Merepresentasikan bagian hubungan yang *physical* antara beberapa *class* atau sebuah *class* dengan dirinya sendiri. |  |

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis & Design: An Object-Oriented Approach with UML, 5th Edition, USA: Wiley, 2015

## 2.4 *Eight Golden Rules of Interface Design*

*Eight golden rules* adalah sebuah prinsip desain yang dapat digunakan sebagai panduan dalam proses perancangan *user interface* dalam suatu *website*. Berikut adalah delapan aturan tersebut [3]:

1. *Strive for consistency*

Dalam perancangan *user interface* suatu *website*, konsistensi sangat dibutuhkan dimana tindakan yang serupa dapat diakses dengan cara yang sama, hal-hal yang berkaitan dikelompokkan bersama, dan *messaging* menggunakan struktur yang seragam.

1. *Enable frequent user-to-use shortcuts*

Seiring dengan meningkatnya frekuensi penggunaan, menyebabkan munculnya keinginan *user* untuk mengurangi tindakan yang perlu dilakukan dalam sistem namun laju interaksi pada sistem tetap meningkat. Pemberian respon dalam waktu yang singkat dapat meningkatkan ketertarikan *user* untuk mengakses *website*.

1. *Offer informative feedback*

Memberikan *feedback* yang informatif setiap *user* melakukan interaksi, dengan tujuan agar *user* dapat mengetahui bahwa perintahnya telah direspon oleh sistem.

1. *Design dialogue to yield closure*

Bagian dari pemberian feedback yang informatif dimana ketika *user* telah menyelesaikan suatu tindakan maka sistem akan memberitahu *user* bahwa tindakan tersebut telah terselesaikan. Hal tersebut dapat mengurangi kekhawatiran *user* dalam menggunakan *website*.

1. *Prevent errors*

Melakukan desain antarmuka yang baik untuk mencegah terjadinya kesalahan yang dilakukan oleh *user* dalam menjalankan sistem. Ketika *user* membuat kesalahan, sistem dapat mendeteksi kesalahan tersebut dan menyediakan instruksi yang sederhana, berguna dan spesifik untuk mengatasinya.

1. *Permit easy reversal of actions*

Mempermudah *user* untuk kembali ke halaman sebelumnya dan membatalkan tindakan yang dapat menyebabkan terjadinya kesalahan.

1. *Support internal locus of control*

*User* dapat mengatur tampilan antarmuka dalam sistem sesuai dengan preferensinya sendiri dan sistem dapat merespon setiap *input* dari *user*. Dengan hal ini, dapat tercipta *user experience* yang baik terhadap *website* tersebut.

1. *Reduce short-term memory load*

Tampilan didesain secara sederhana agar dapat mengurangi beban ingatan jangka pendek yang dapat dialami *user* dalam mengingat data apa saja yang harus diinput dalam sistem.

## 2.5 *Relational Database Management System* (RDBMS)

*Database* merupakan sebuah kumpulan data yang terkait secara logis dan deskripsi dari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Di dalam sebuah *database* terdapat *entity*, *attribute*, dan *relationship*. *Entity* merupakan sebuah objek atau benda yang dapat dibedakan dari objek lainnya. *Attribute* merupakan properti yang mendeskripsikan beberapa aspek dari *entity*. *Relationship* adalah hubungan antar *entity* [4].

*Database Management System* (DBMS) adalah sebuah sistem *software* yang memungkinkan *user* untuk mendefinisikan, menciptakan, mempertahankan, dan memiliki kontrol akses terhadap *database*. Fasilitas yang disediakan oleh DBMS adalah sebagai berikut [4]:

1. *Data Definition Languag*e membebaskan *user* untuk menentukan tipe data, struktur, dan kendala pada data yang akan disimpan di *database*.
2. *Data Manipulation Language* membebaskan *user* untuk melakukan *insert*, *update*, *delete*, dan memperoleh data pada *database*. DML menyediakan fasilitas berupa bahasa *query* yang dapat digunakan untuk melakukan permintaan terhadap *database* dan sistem informasi. Bahasa *query* yang paling umum adalah *Structured Query Language* (SQL).
3. Penyediaan akses kontrol pada *database*.

*Relational Database Management System* (RDBMS) merupakan sebuah sistem yang didasarkan oleh model relasional untuk mengatur sekumpulan data yang kemudian dipresentasikan melalui tabel. Data relasional memiliki struktur-struktur sebagai berikut [4]:

1. *Relation* adalah sebuah tabel yang berisi kolom dan baris.
2. *Attribute* adalah kolom dari sebuah relasi.
3. *Domain* adalah kumpulan nilai yang dapat digunakan untuk satu atribut atau lebih.
4. *Tuple* adalah suatu baris dari sebuah relasi.
5. *Degree* adalah jumlah atribut yang ada dalam sebuah relasi.
6. *Cardinality* adalah jumlah *tuple* yang ada dalam sebuah relasi.
7. *Relational database* adalah kumpulan relasi yang telah ternormalisasi dengan nama relasi yang berbeda-beda.

Dalam pemodelan relasional, tidak diperbolehkan adanya duplikasi *tuple* dalam sebuah relasi pada *database*. Setiap *tuple* dalam sebuah relasi harus diidentifikasi menggunakan *relational keys*. *Relational keys* terdiri dari atribut-atribut sebagai berikut [4]:

1. *Superkey* adalah sebuah atau sekumpulan atribut yang secara unik mengidentifikasi *tuple* dalam sebuah relasi.
2. *Candidate key* adalah sebuah *superkey* yang memiliki nilai unik dan dapat dipilih sebagai *primary key* dalam suatu tabel. Dalam sebuah relasi, akan ada kemungkinan munculnya beberapa *candidate key*. Ketika sebuah *key* terdiri dari dua atau lebih atribut, maka *key* tersebut dinamakan *composite key*.
3. *Primary key* adalah sebuah *candidate key* yang dipilih untuk mengidentifikasi *tuple* dalam sebuah relasi secara unik. *Candidate key* yang tidak terpilih menjadi *primary key* dinamakan *alternate key*.
4. *Foreign key* adalah sebuah atau sekumpulan atribut dalam sebuah relasi yang memiliki kesamaan dengan *candidate key* dari beberapa relasi lainnya. *Foreign key* ini merupakan kunci penghubung.

## 2.6 PHP: *Hypertext Preprocessor*

PHP merupakan bahasa pemrograman berbentuk *script* yang dapat menciptakan kedinamisan dalam sebuah *website* [5]. PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Pada awalnya, PHP merupakan singkatan dari *Personal Home Page*. Namun seiring dengan perkembangannya yang cepat dan baik, singkatan PHP berubah menjadi *Hypertext Preprocessor*.

PHP lebih berfokus pada *server-side scripting* yang merupakan sebuah teknik pemrograman *web* dimana bahasa dan perintah dalam program dapat dijalankan melalui *web server* [6]. PHP dapat digunakan di berbagai *major operating systems*, termasuk Linux, Unix, Microsoft Windows, MacOS dan sistem operasi lainnya. Berbagai *web server* seperti Apache, IIS (*Internet Information Server*), dan PWS (*Personal Web Server*) juga telah mendukung penggunaan PHP. Dengan menggunakan PHP, *user* mendapatkan kebebasan untuk memilih sebuah sistem operasi dan *web server*. Selain itu, *user* juga dapat menentukan jenis pemrograman apa yang ingin digunakan yaitu antara pemrograman prosedural atau pemrograman berorientasi objek, ataupun campuran keduanya. Fitur signifikan lain yang dimiliki oleh PHP adalah dukungan yang luas pada berbagai basis data seperti MySQL, PostgreSQL, Oracle, dan basis data lainnya [7].

## 2.7 JavaScript

JavaScript merupakan bahasa pemrograman berorientasi objek yang memungkinkan proses penerjemahan dan kompilasi kode pada elemen-elemen yang ada dalam sebuah dokumen HTML (*Hypertext Markup Language*) [5]. JavaScript digunakan untuk memberikan fitur yang interaktif dalam sebuah *website* seperti animasi, tombol yang dapat diklik, menu *pop-up*, dan banyak lagi. Dengan menggunakan JavaScript, dapat menghasilkan sebuah *website* yang interaktif dan dinamis sehingga mempermudah *user* untuk menggunakan *website* tersebut.

Penggunaan bahasa pemrograman JavaScript memiliki berbagai fitur umum yang memungkinkan terjadinya hal-hal berikut ini [8]:

1. Dapat menyimpan nilai-nilai (*values*) di dalam variabel. Ketika *user* telah memberikan *input* atas perintah yang diberikan oleh sistem, maka sistem akan menyimpan *input* tersebut ke dalam variabel yang ada.
2. Dapat beroperasi pada potongan teks yang lebih dikenal sebagai *strings* dalam *programming*.
3. Dapat merespon berbagai kejadian yang terjadi pada sistem dengan menjalankan kode. Adanya penyediaan *button* yang dapat mempermudah sistem untuk mendeteksi adanya interaksi yang dilakukan oleh *user* sehingga sistem dapat langsung melanjutkan ke tahap berikutnya.

## 2.8 *Testing*

*Testing* merupakan teknik yang digunakan untuk memverifikasi dan memvalidasi kualitas dari suatu sistem. *Testing* dilakukan bukan untuk menunjukkan bahwa sistem terbebas dari *error*, melainkan *testing* dilakukan dengan tujuan untuk menemukan kesalahan yang terdapat pada sistem [2].

*Black box testing* merupakan salah satu teknik pengujian yang bertujuan untuk menguji suatu sistem dari segi fungsionalitasnya. Pada *black box testing*, penguji tidak memiliki akses pada bagian *internal* sistem. Penguji hanya perlu mengetahui bagaimana *output* dan *input* dapat dikeluarkan oleh sistem tanpa memahami bagaimana proses yang terjadi di dalamnya. *Black box testing* dilakukan berdasarkan sudut pandang *user* dimana penentuan *requirements* dilakukan oleh *user*. Keuntungan penggunaan teknik pengujian ini adalah membantu untuk mengidentifikasi ambiguitas atau inkonsistensi *requirements* yang terdapat pada sistem [9].

## 2.9 Sistem Informasi

Sistem informasi merupakan kumpulan dari komponen yang saling berkaitan dan memiliki tujuan yang sama yaitu untuk memperoleh, mengolah, menyimpan, dan menyebarkan informasi sehingga dapat membantu pengambilan keputusan dalam organisasi. Komponen-komponen tersebut adalah berbagai jenis teknologi informasi seperti komputer, *software*, *database*, *communication system*, dan internet [10]. Sistem informasi tidak berfokus pada bagian teknis dalam teknologi informasi, melainkan sistem informasi berfokus pada penggunaan teknologi informasi dalam menyediakan informasi yang dapat digunakan untuk memenuhi kebutuhan *user*. Dengan adanya sistem informasi yang dapat memberikan informasi yang akurat dan memiliki *value*, tentu akan mempermudah *user* maupun organisasi dalam melaksanakan tugas tertentu sehingga tujuan yang diinginkan dapat lebih mudah tercapai.

## 2.10 *Event Organizer* (EO)

*Event* merupakan sebuah acara yang diselenggarakan dengan tujuan untuk memperingati hal-hal penting atau bermakna yang terjadi sepanjang hidup manusia baik secara individu maupun kelompok yang dapat berikatan dengan adat, budaya, tradisi, dan agama [11].

Seiring perkembangannya zaman, muncul banyak sekali ide dalam menyelenggarakan sebuah *event*. *Event* tidak hanya digunakan untuk memperingati peristiwa-peristiwa yang bersifat personal, melainkan dapat digunakan sebagai media untuk mempromosikan kegiatan-kegiatan bisnis dalam sebuah organisasi. Namun dalam rangka mencapai tujuan tersebut, diperlukan sebuah perencanaan yang baik agar *event* dapat berjalan dengan lancar.

*Event organizer* (EO) adalah sebuah profesi yang memerlukan adanya pertemuan antar sekelompok orang dengan tujuan perayaan, pendidikan, pemasaran dan reuni. Sekelompok orang tersebut akan melakukan serangkaian kegiatan untuk mewujudkan sebuah acara [12]. Tugas seorang EO adalah untuk membantu mewujudkan permintaan klien dengan menghasilkan sebuah acara yang sesuai dengan yang diharapkan klien. Penggunaan EO dapat memberikan banyak kelebihan bagi klien yaitu menghemat waktu, membantu pemilihan tema dan dekorasi yang tepat, dan membantu anggaran menjadi lebih terkendali. EO terdiri dari berbagai macam jenis; ada yang bergerak di bidang olahraga, seni, pameran, maupun acara yang bersifat pribadi seperti pesta pernikahan, ulang tahun, dan acara lainnya [1].

## 2.11 *Decision Support System* (DSS)

*Decision Support System* (DSS) atau Sistem Pendukung Keputusan didefinisikan sebagai sebuah sistem informasi berbasis komputer yang dapat mendukung kegiatan sebuah bisnis atau organisasi dalam mengambil keputusan (sumber). DSS dirancang sebagai sistem yang bersifat interaktif dengan tujuan untuk membantu para pengambil keputusan mengumpulkan informasi yang berguna dari berbagai data mentah, dokumen, pengetahuan pribadi, atau model bisnis yang telah disediakan untuk mengidentifikasi, memecahkan masalah dan mengambil keputusan (sumber).

# 

# BAB III

# SISTEM SAAT INI

## 3.1 Profil Perusahaan

PT. XYZ merupakan sebuah perusahaan jasa yang berlokasi di Jakarta Barat dan yang bergerak di bidang *Event Organizer*. Perusahaan ini berdiri pada tahun 2007 ketika para *founder*-nya masih menjalani pendidikan sarjana dengan tujuan menciptakan sebuah acara yang baik dengan harga yang terjangkau. Perusahaan ini telah melayani lebih dari 1300 acara lokal, nasional maupun internasional sejak tahun 2007.

PT. XYZ menyediakan berbagai layanan jasa berupa penyelenggaraan acara seperti pesta ulang tahun, pernikahan, seminar, reuni, acara perpisahan sekolah, dan banyak acara lainnya. Selain itu, perusahaan ini juga menawarkan layanan berupa dekorasi seperti *stage decoration*, *booth production*, *multimedia and animation*, *area decoration*, dan *table decoration*. Perusahaan ini dapat membantu dalam memilih vendor terbaik dan berkualitas dari hal-hal yang dibutuhkan oleh klien agar acara dapat berjalan sesuai yang diharapkan. Perusahaan ini telah mempublikasikan paket dan harga jualnya secara *online* untuk memastikan setiap klien mendapatkan harga dan perlakuan yang sama namun juga terbaik.

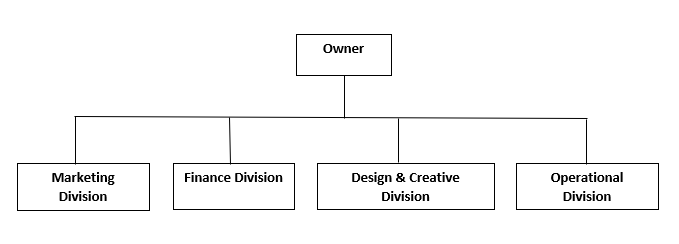
### 3.1.1 Visi Perusahaan

PT. XYZ memiliki visi untuk menjadi EO yang profesional, terpercaya, terbaik, dan dapat menjadi pilihan pertama bagi para masyarakat, perusahaan maupun organisasi setiap membuat acara.

### 3.1.2 Misi Perusahaan

Misi dari PT. XYZ adalah dapat memberikan nilai tambah dan keunikan di setiap acara yang telah diselenggarakan sehingga dapat memberikan dampak positif bagi klien yang menggunakan jasanya. Kepuasan klien merupakan prioritas utama dari perusahaan ini.

## 3.2 Struktur Organisasi

Struktur organisasi adalah susunan lembaga yang menggambarkan hubungan kerja dalam sebuah organisasi. Dengan adanya struktur organisasi, maka dapat diketahui tanggung jawab dan tugas yang dipegang dalam setiap bagian organisasi. Pada struktur organisasi PT. XYZ terdapat berbagai peran yaitu *owner*, *marketing division*, *finance division*, *design & creative division*, dan *operational division* yang dapat dilihat pada Gambar 3.1. 

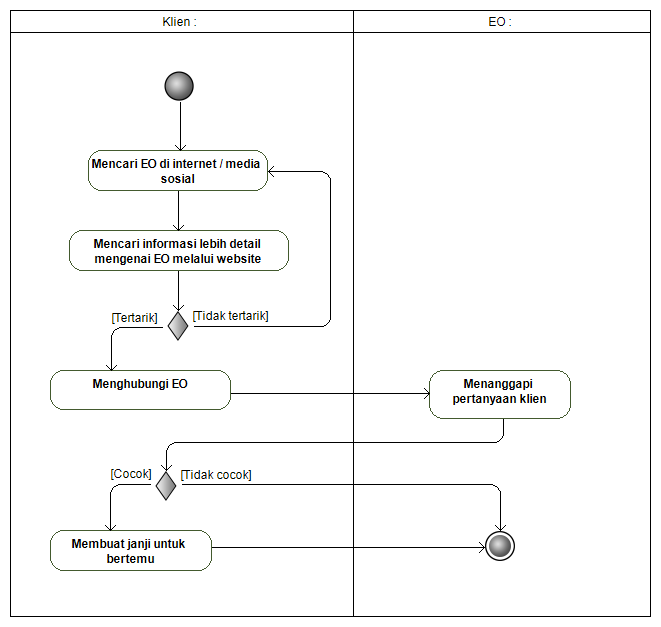
Gambar 3.1 Struktur Organisasi

Berikut adalah deskripsi tugas dari masing-masing peran yang ada pada struktur organisasi di atas:

1. *Owner*, merupakan pemilik dari perusahaan. *Owner* bertanggung jawab menangani seluruh manajemen dalam perusahaan dan mengawasi performa kerja dari setiap divisi.
2. *Marketing division*, bertanggung jawab untuk mencari klien, melakukan prospek pada calon klien, melakukan kesepakatan dengan klien.
3. *Finance division*, bertanggung jawab untuk menangani seluruh keuangan perusahaan.
4. *Design & creative division*, bertanggung jawab untuk membuat desain, konsep acara, konsep ruangan dan dekorasi lainnya.
5. *Operational division*, bertanggung jawab untuk memimpin proyek dan mengkoordinasi seluruh kepentingan dalam penyelenggaraan acara serta memastikan acara dapat berjalan dengan lancar dan sesuai dengan konsep yang telah dibuat.

## 3.3 Analisis Sistem Saat ini

Berikut adalah sistem pencarian jasa EO pada saat ini:



Gambar 3.2 *Activity Diagram* Pencarian Jasa EO

Klien mencari jasa EO yang ada melalui internet atau media sosial kemudian mencari informasi lebih detail mengenai EO tersebut melalui *website* secara satu per satu. Jika klien merasa tertarik untuk menggunakan EO tersebut, maka klien akan menghubungi EO tersebut. Apabila klien telah memperoleh informasi yang dibutuhkan dan merasa cocok dengan EO tersebut, maka klien akan membuat janji untuk bertemu secara langsung dengan EO tersebut dan membahas lebih lanjut mengenai kebutuhan klien.

### Kendala Sistem Saat ini

Kendala yang terdapat dalam sistem pencarian jasa EO pada saat ini:

1. Dari pihak klien, kendala yang dihadapi adalah pencarian jasa EO yang dilakukan secara satu per satu dari banyaknya EO yang dapat ditemukan di internet maupun di berbagai media sosial sehingga membutuhkan waktu yang lama dalam menentukan jasa EO yang ingin digunakan. Selain itu, penyediaan informasi mengenai harga paket yang disediakan, *portfolio* dari acara yang pernah diselenggarakan, dan testimoni dari klien yang pernah menggunakan jasanya seringkali tidak tertera secara lengkap dan jelas di dalam *website* maupun didalam akun media sosial sehingga menimbulkan kesulitan dalam mencari jasa EO yang terpercaya dan sesuai dengan acara, tema, dan *budget*.
2. Dari pihak EO, kendala yang dihadapi adalah sulitnya mempromosikan jasa-jasa yang disediakan dan mendapatkan kepercayaan klien pada tahap awalnya.

# BAB IV ANALISIS DAN PERANCANGAN SISTEM USULAN

## 4.1 Analisis Kelayakan Sistem

Analisis kelayakan terhadap sistem usulan yang dilakukan pada tugas akhir ini terdiri dari tiga teknik analisis, yaitu kelayakan teknis (*technical feasibility*), kelayakan ekonomi (*economic feasibility*) dan kelayakan organisasi (*organizational feasibility*).

1. Kelayakan teknis (*technical feasibility*)

Secara teknis, terdapat beberapa risiko yang mungkin dihadapi dalam penggunaan aplikasi yang akan dirancang ini, yaitu:

1. Tingkat risiko mengenai *familiarity* terhadap aplikasi adalah sedang.

Pengguna aplikasi layanan jasa *event organizer* ini terdiri dari berbagai EO yang ingin mempublikasikan jasanya dan klien yang ingin mencari jasa EO. Pengguna tentu telah terbiasa menggunakan aplikasi berbasis *web* yang sering dimanfaatkan untuk melakukan kegiatan berbisnis maupun untuk mencari informasi. Selain itu, aplikasi usulan yang akan dirancang menyediakan berbagai fitur yang sering muncul di berbagai *website* pada umumnya sehingga tidak menjadi hambatan yang besar bagi pengguna untuk memahami dan menggunakan aplikasi ini.

1. Tingkat risiko mengenai *familiarity* terhadap teknologi adalah sedang.

Pengguna telah terbiasa menggunakan teknologi berbasis *web* yang cukup umum digunakan dikarenakan penggunaannya yang dapat memberikan banyak kemudahan dalam pelaksanaan kegiatan sehari-hari, salah satunya adalah dalam proses pencarian informasi untuk memperluas pengetahuan.

1. Tingkat risiko dari segi pembangunan aplikasi adalah tinggi.

Perancangan aplikasi usulan ini memiliki *requirements* yang cukup banyak dengan melibatkan beberapa pengguna yang harus dipenuhi. Selain itu, proses pembuatan aplikasi ini mempunyai waktu sekitar kurang lebih empat bulan.

1. Kelayakan ekonomi (*economic feasibility*)

Aplikasi usulan ini dapat memberikan keuntungan bagi PT. XYZ, terutama keuntungan yang bersifat *intangible*. Dengan adanya aplikasi berbasis *web* untuk layanan jasa *event organizer* ini, kegiatan yang berhubungan dengan pencarian EO dapat berjalan lebih cepat sehingga dapat memberikan manfaat bagi pengguna sekaligus meningkatkan kinerja dari perusahaan.

Berikut daftar *hardware* dan *software* yang dibutuhkan untuk mendukung aplikasi berbasis *web* ini terdapat pada Tabel 4.1.

Tabel 4.1 Spesifikasi Minimum yang Dibutuhkan untuk Sistem Usulan

|  |  |
| --- | --- |
| ***Hardware*** | |
| *Harddisk* | 500 GB |
| RAM | 4 GB |
| *Processor* | Intel® Core™ i3 |
| ***Software*** | |
| Sistem Operasi | Windows 7 ke atas |
| *Web Browser* | Google Chrome, Mozilla Firefox |
| *Web Server* | Apache |
| *Database* | MySQL |

Sedangkan *hardware* dan *software* yang dimiliki oleh PT. XYZ terdapat pada Tabel 4.2.

Tabel 4.2 Spesifikasi yang Dimiliki oleh PT. XYZ

|  |  |
| --- | --- |
| ***Hardware*** | |
| *Harddisk* | 500 GB |
| RAM | 4 GB |
| *Processor* | Intel® Core™ i3 |
| ***Software*** | |
| Sistem Operasi | Windows 7 Professional |

Dapat dilihat dari Tabel 4.1 dan Tabel 4.2 bahwa PT. XYZ telah memenuhi spesifikasi *hardware* dan *software* yang akan diperlukan untuk aplikasi berbasis *web* ini sehingga tidak perlu menambah biaya untuk menambah perangkat baru.

1. Kelayakan organisasi (*organizational feasibility*)

Analisis kelayakan organisasi merupakan analisis yang bertujuan untuk menilai bagaimana sistem usulan dapat diterima oleh pengguna. Dari perspektif organisasi, aplikasi berbasis *web* ini memiliki risiko yang rendah. Perancangan aplikasi ini dapat memberikan keuntungan bagi pengguna dikarenakan tujuannya yang dapat mempermudah proses pencarian EO serta membantu para EO untuk mempublikasikan jasanya. Selain itu, pengembangan aplikasi ini juga didukung oleh perusahaan.

## 4.2 Tahap Analisis

Pada tahap ini akan dilakukan analisis terhadap *business requirements* dalam sistem usulan dan juga pemodelan dari sistem usulan secara fungsional maupun struktural. Hasil dari analisis tersebut akan digunakan sebagai dasar dalam pembuatan sistem yang diusulkan.

### 4.2.1 User Requirements

*Requirements* merupakan suatu penjelasan mengenai hal apa saja yang harus dilakukan oleh sistem dan karakteristik yang dimiliki oleh sistem yang dapat berguna bagi *user*. *Requirements* diperoleh melalui hasil analisis terhadap kebutuhan *user* dan studi literatur. Secara umum, *user requirements* terdiri dari *functional requirements* dan *non-functional requirements*.

#### *4.2.1.1 Functional Requirements*

*Functional requirements* didefinisikan sebagai kebutuhan yang harus dapat dilakukan oleh sistem. *Functional requirements* pada aplikasi layanan jasa *event organizer* berbasis *web* ini dibagi menjadi dua, yaitu:

1. Aplikasi berbasis *web* untuk *front-end*

Berikut adalah fungsi-fungsi yang harus dimiliki pada aplikasi layanan jasa *event organizer* berbasis *web* untuk *front-end*:

1. Melakukan pendaftaran akun
2. Fungsi ini digunakan oleh *user* untuk membuat akun;
3. User mengisi formulir yang disediakan berupa *email*, *password*, nama dan nomor *handphone*.
4. *User* akan menerima bukti konfirmasi melalui *email* yang akan dikirimkan oleh sistem.
5. Melakukan pendaftaran sebagai EO
6. Fungsi ini digunakan oleh EO untuk membuat akun;
7. EO mengisi informasi mengenai data usaha yang dibutuhkan pada formulir pendaftaran sebagai EO;
   1. Semua informasi dalam formulir wajib diisi dengan lengkap agar dapat diproses lebih lanjut oleh sistem;
   2. EO perlu mengunggah dokumen seperti foto KTP, foto diri dengan KTP, foto tagihan listrik/telepon/air yang mencantumkan alamat, foto SIUP/TDP untuk memvalidasi bisnis.
8. Melihat jasa EO
9. Fungsi ini digunakan oleh *user* untuk melihat jasa EO yang ada;
10. Sistem melakukan kategorisasi terhadap EO yang telah terdaftar berdasarkan jenis pelayanan dan lokasi serta menampilkan informasinya yang lengkap.
11. Mengajukan *request* layanan

Klien dapat mengajukan *request* layanan dengan mengisi informasi-informasi yang dibutuhkan pada formulir yang telah disediakan dengan lengkap dan jelas.

1. Mengelola *request* layanan

EO dapat melihat dan menghapus *request* layanan yang telah diajukan oleh klien.

1. Memberi *review*

Klien memberi *review* terhadap EO berupa *rating* dan testimoni mengenai pelayanan yang telah disediakan EO tersebut.

1. Melihat rekomendasi EO
2. Klien menentukan bobot kepentingan dari kriteria EO yang diinginkan.
3. Sistem akan menampilkan rekomendasi EO yang sesuai dengan preferensi klien.
4. Mengelola *portfolio*

Fungsi ini digunakan oleh EO untuk menambah, mengubah, serta mengurangi data mengenai *portfolio*-nya.

1. Mengelola paket

Fungsi ini digunakan oleh EO untuk menambah, mengubah, serta mengurangi data mengenai paket yang disediakan.

1. Mengelola *appointment reminder*
2. Fungsi ini digunakan oleh EO untuk menambah dan mengurangi data serta mengubah status *appointment reminder*;
3. EO dapat menentukan tanggal, jam dan keterangan mengenai hal yang harus dilakukan dan klien akan menerima notifikasi berupa *email*.
4. Mengelola pesan
5. Fungsi ini digunakan oleh klien untuk mengirim pesan kepada EO;
6. EO menerima pesan yang dikirimkan melalui fitur *inbox* dan dapat membalas pesan tersebut.
7. Aplikasi berbasis *web* untuk *back-end*

Berikut adalah fungsi-fungsi yang harus dimiliki pada aplikasi layanan jasa *event organizer* berbasis *web* untuk *back-end*:

1. Mengelola akun

Fungsi ini digunakan oleh *admin* untuk melakukan perubahan serta pengurangan data mengenai akun klien dan EO.

1. Mengelola kriteria

Fungsi ini digunakan oleh *admin* untuk melakukan perubahan kriteria berupa nama, tipe dan bobotnya.

1. Mengelola data kriteria EO

Fungsi ini digunakan oleh *admin* untuk melakukan penambahan, perubahan serta pengurangan data kriteria berupa nama alternatif (EO), kriteria, dan nilai yang dimiliki.

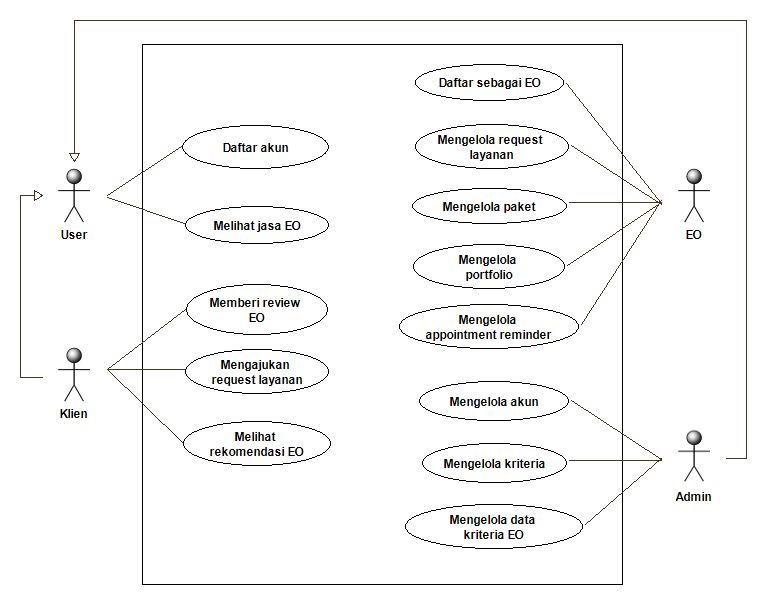
#### *4.2.1.2 Non-functional Requirements*

*Non-functional requirements* menjelaskan apa saja batasan-batasan yang terdapat pada sistem dan bagaimana sistem dapat berjalan. *Non-functional requirements* pada aplikasi layanan jasa *event organizer* berbasis *web*, antara lain:

1. Operasional
2. Aplikasi ini dapat dijalankan pada *web browser* Google Chrome (versi 53 hingga sekarang) dan Mozilla Firefox (versi 48 hingga sekarang).
3. Aplikasi ini dapat dijalankan dengan baik pada sistem operasi Windows 7 dan setelahnya.
4. Performa
5. Aplikasi ini dapat diakses selama 24 jam.
6. Informasi yang ada dalam *database* dapat diperbaharui secara *real-time*.
7. Keamanan
8. Aplikasi ini memiliki beberapa level akses pengguna yang terdiri dari klien, EO, dan *admin* yang masing-masing memiliki akses terhadap fungsi-fungsi tertentu.
9. Untuk menjalankan berbagai fungsi dalam aplikasi, pengguna harus masuk ke akun dengan memasukkan *email* dan *password* yang benar.
10. Lingkungan organisasi
11. Tampilan aplikasi dirancang dengan baik dan memiliki struktur yang teratur agar mudah dimengerti dan digunakan oleh pengguna.
12. Penggunaan bahasa dalam aplikasi ini adalah Bahasa Inggris.

### 4.2.2 Use Case Diagram

Berikut *use case diagram* dari aplikasi layanan jasa *event organizer* berbasis *web* yang diusulkan dapat dilihat pada Gambar 4.1.



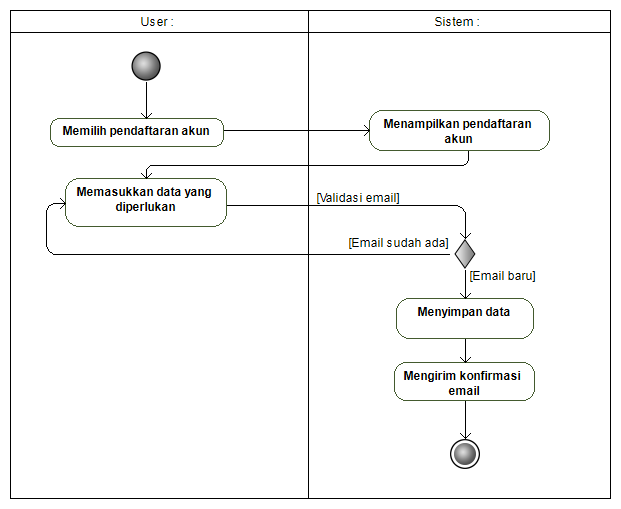
Gambar 4.1 *Use Case Diagram* Sistem Usulan

### 4.2.3 Activity Diagram

Berikut adalah proses-proses utama dari aplikasi layanan jasa *event organizer* berbasis *web* ini yang digambarkan melalui *activity diagram*:

1. Daftar akun

Proses pendaftaran akun dapat dilihat pada Gambar 4.2.



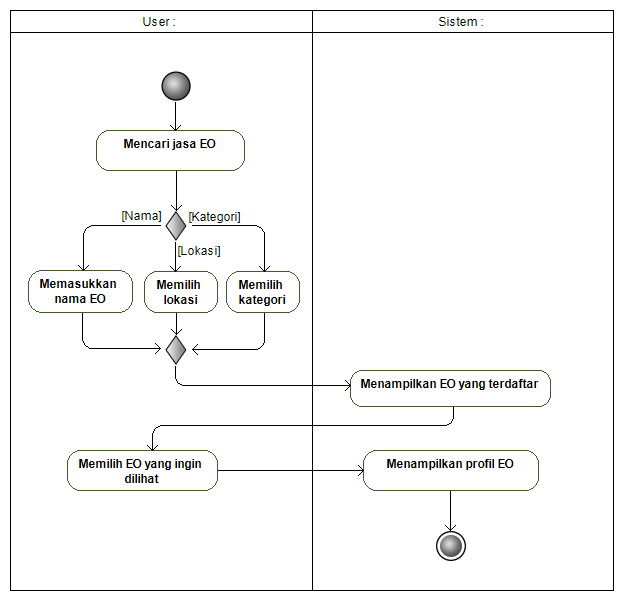
Gambar 4.2 *Activity Diagram:* Daftar akun

Proses pendaftaran akun terdiri dari langkah-langkah sebagai berikut:

* + 1. *User* melakukan pendaftaran akun secara *online*;
    2. Sistem menampilkan halaman pendaftaran akun;
    3. *User* memasukkan nama lengkap, nomor *handphone*, *e-mail*, dan *password*;
    4. Sistem melakukan verifikasi terhadap data yang telah dimasukkan oleh *user*;
    5. Jika data yang dimasukkan tidak lengkap dan benar, maka sistem menampilkan notifikasi *error* dan *user* kembali ke langkah nomor 3;
    6. Jika data yang dimasukkan telah lengkap dan benar, maka sistem akan menyimpan data dan mengirimkan *e-mail* sebagai bukti konfirmasi *user* telah melakukan pendaftaran akun.

1. Melihat jasa EO

Proses melihat jasa EO dapat dilihat pada Gambar 4.3.

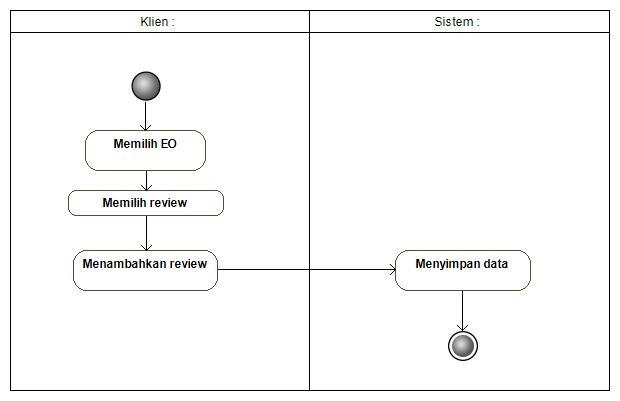


Gambar 4.3 *Activity Diagram*: Melihat jasa EO

Proses melihat jasa EO terdiri dari langkah-langkah sebagai berikut:

1. *User* mengunjungi halaman utama *website*;
2. *User* dapat mencari jasa EO dengan memasukkan nama, lokasi, maupun kategori dari EO yang diinginkan;
3. Sistem menampilkan EO yang telah terdaftar;
4. *User* memilih EO yang ingin dilihat;
5. Sistem menampilkan profil dari EO yan telah dipilih oleh *user*.
6. Memberi *review*

Proses pemberian *review* dapat dilihat pada Gambar 4.4.

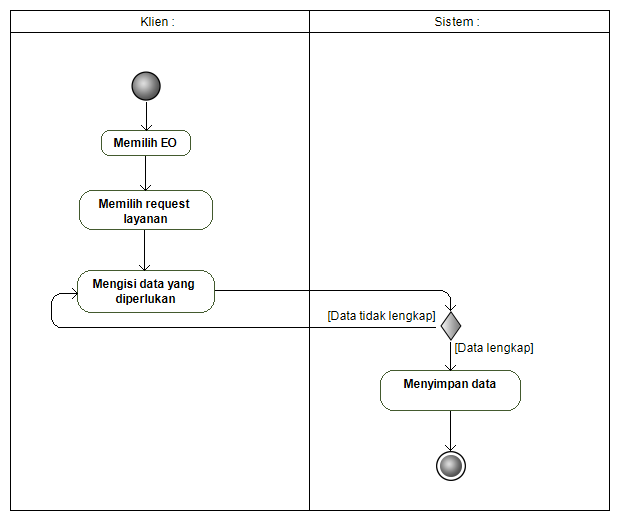


Gambar 4.4 *Activity Diagram*: Memberi *review*

Proses pemberian *review* terdiri dari langkah-langkah sebagai berikut:

1. Klien yang telah menyelesaikan sebuah *appointment* dengan EO akan menerima *e-mail* berupa *link* untuk melakukan pemberian *review*;
2. Sistem menampilkan halaman pemberian *review*;
3. Klien menambahkan *review* terhadap EO tersebut;
4. Sistem akan menyimpan data dan menampilkan *review* tersebut pada profil EO.
5. Mengajukan *request* layanan

Proses pengajuan *request* layanan dapat dilihat pada Gambar 4.5.

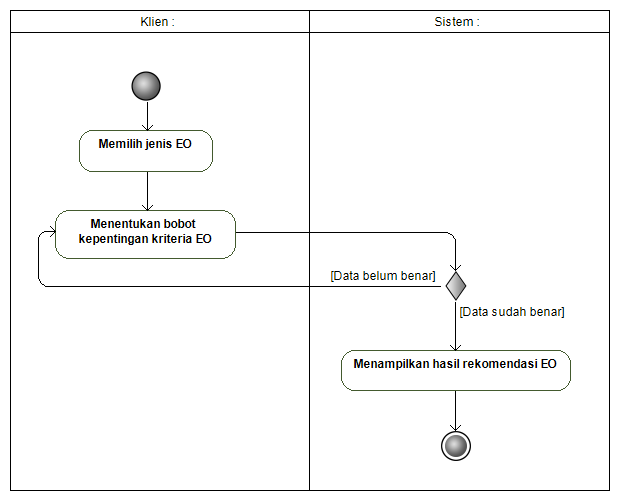


Gambar 4.5 *Activity Diagram*: Mengajukan *request* layanan

Proses pengajuan *request* layanan terdiri dari langkah-langkah sebagai berikut:

1. Klien memilih jasa EO yang diinginkan dan memilih untuk mengajukan *request* layanan;
2. Sistem menampilkan formulir untuk pengajuan *request* layanan;
3. Klien mengisi data yang diperlukan berupa detail acara yang ingin diselenggarakan dan jenis paket yang diinginkan;
4. Jika data yang dimasukkan telah lengkap dan benar, maka sistem akan menyimpan data tersebut dan mengajukannya kepada EO.
5. Melihat rekomendasi EO

Proses untuk melihat rekomendasi EO dapat dilihat pada Gambar 4.6.

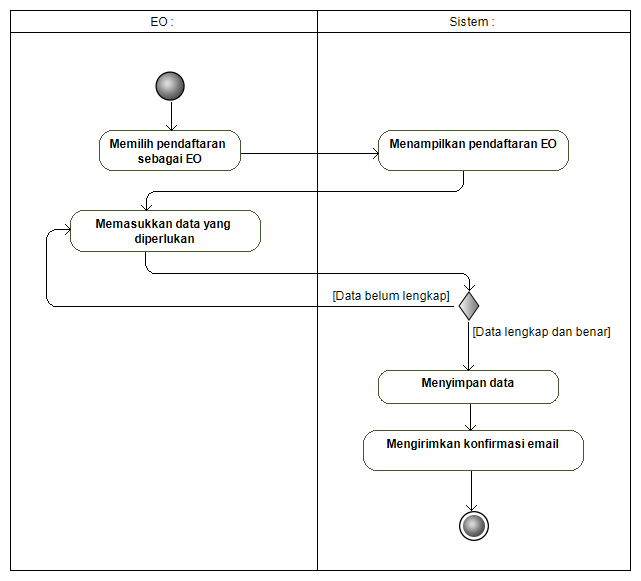


Gambar 4.6 *Activity Diagram*: Melihat rekomendasi EO

Proses untuk melihat rekomendasi EO terdiri dari langkah-langkah sebagai berikut:

1. Klien memilih untuk melihat rekomendasi EO;
2. Sistem menampilkan halaman berupa formulir pemberian bobot kriteria EO yang harus diisi beserta instruksi yang telah disediakan;
3. Klien memasukkan data yang diperlukan berupa kategori EO yang diinginkan, pemberian bobot kepentingan dari kriteria EO yang disediakan dengan lengkap dan benar;
4. Jika data telah benar, maka sistem akan memproses data tersebut dan menampilkan hasil rekomendasi EO yang telah diurutkan dari yang paling direkomendasikan.
5. Daftar sebagai EO

Proses pendaftaran sebagai EO dapat dilihat pada Gambar 4.7.

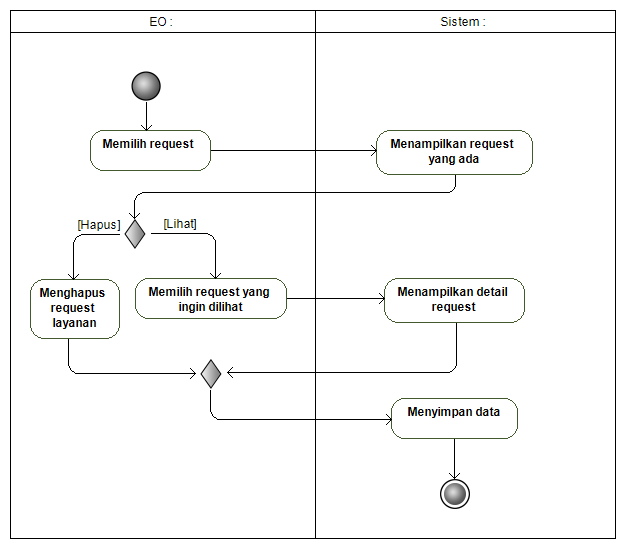


Gambar 4.7 *Activity Diagram*: Daftar sebagai EO

Proses pendaftaran sebagai EO terdiri dari langkah-langkah sebagai berikut:

1. EO memilih untuk melakukan pendaftaran sebagai EO;
2. Sistem menampilkan halaman pendaftaran sebagai EO;
3. EO memasukkan data yang diperlukan berupa *e-mail*, *password*, nomor *handphone*, profil perusahaan dan data lain yang diperlukan untuk validasi bisnis;
4. Jika data yang dimasukkan tidak lengkap dan benar, maka sistem menampilkan notifikasi *error* dan EO kembali ke langkah nomor 3;
5. Jika data telah dimasukkan dengan lengkap dan benar, maka sistem akan menyimpan data dan data tersebut akan dikonfirmasi oleh *admin* terlebih dahulu;
6. Sistem mengirimkan *e-mail* berupa bukti konfirmasi EO telah melakukan pendaftaran akun.
7. Mengelola *request* layanan

Proses pengelolaan *request* layanan dapat dilihat pada Gambar 4.8.

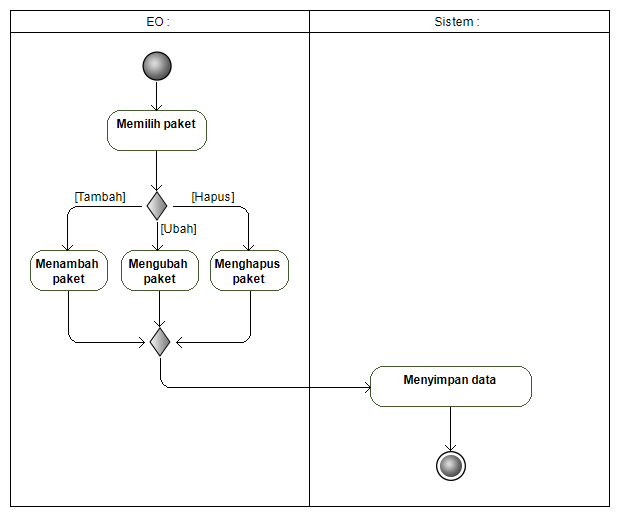


Gambar 4.8 *Activity Diagram*: Mengelola *request* layanan

Proses pengelolaan *request* layanan terdiri dari langkah-langkah sebagai berikut:

1. EO memilih *request* pada halaman *dashboard*;
2. Sistem menampilkan halaman *request* berupa *request* layanan yang telah diajukan oleh berbagai klien;
3. EO memilih untuk melihat *request* layanan;
4. EO memilih *request* layanan yang ingin dilihat;
5. Sistem menampilkan detail dari *request* layanan yang telah dipilih.
6. EO memilih untuk menghapus *request* layanan;
7. EO memilih *request* layanan yang ingin dihapus;
8. Sistem menampilkan notifikasi berupa konfirmasi penghapusan *request* layanan;
9. Jika EO mengkonfirmasi, maka *request* layanan tersebut akan dihapus dari halaman *request*.
10. Mengelola paket

Proses pengelolaan paket dapat dilihat pada Gambar 4.9.

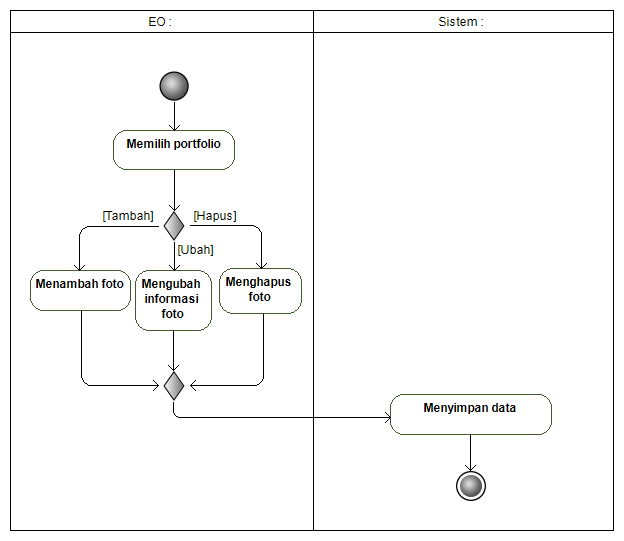


Gambar 4.9 *Activity Diagram*: Mengelola paket

Proses pengelolaan paket terdiri dari langkah-langkah sebagai berikut:

1. EO memilih paket pada halaman *dashboard*;
2. Sistem menampilkan halaman paket berupa paket yang disediakan oleh EO tersebut;
3. EO memilih untuk menambahkan paket;
4. Sistem menampilkan formulir untuk melakukan penambahan paket;
5. EO mengisi data yang diperlukan berupa nama paket, jenis, harga dan keterangannya dengan benar;
6. Jika data telah dimasukkan dengan benar, maka sistem akan menyimpan data tersebut.
7. EO memilih untuk mengubah paket;
8. EO memilih paket yang ingin diubah;
9. Sistem menampilkan halaman untuk melakukan perubahan paket;
10. EO mengubah data mengenai paket yang diinginkan;
11. Sistem menyimpan perubahan data.
12. EO memilih untuk menghapus paket;
13. EO memilih paket yang ingin dihapus;
14. Sistem menampilkan notifikasi berupa konfirmasi penghapusan paket;
15. Jika EO mengkonfirmasi, maka paket tersebut akan dihapus dari halaman paket.
16. Mengelola *portfolio*

Proses pengelolaan *portfolio* dapat dilihat pada Gambar 4.10.

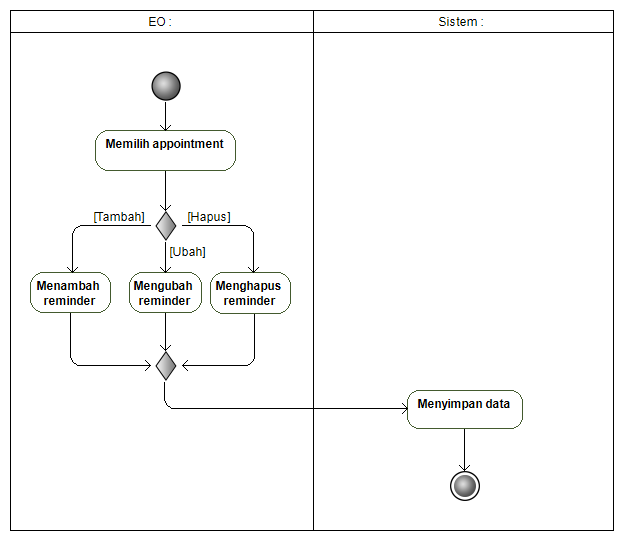


Gambar 4.10 *Activity Diagram*: Mengelola *portfolio*

Proses pengelolaan *portfolio* terdiri dari langkah-langkah sebagai berikut:

1. EO memilih *portfolio* pada halaman *dashboard*;
2. Sistem menampilkan halaman *portfolio* berupa *portfolio* yang dimiliki oleh EO tersebut;
3. EO memilih untuk menambahkan foto;
   * + 1. Sistem menampilkan formulir untuk melakukan penambahan foto;
       2. EO mengisi data yang diperlukan berupa foto dan keterangannya dengan benar;
       3. Jika data telah dimasukkan dengan benar, maka sistem akan menyimpan data tersebut.
4. EO memilih untuk mengubah informasi foto;
5. EO memilih foto yang ingin diubah;
6. Sistem menampilkan halaman untuk melakukan perubahan foto;
7. EO mengubah informasi foto yang diinginkan;
8. Sistem menyimpan perubahan data.
9. EO memilih untuk menghapus foto;
10. EO memilih foto yang ingin dihapus;
11. Sistem menampilkan notifikasi berupa konfirmasi penghapusan foto;
12. Jika EO mengkonfirmasi, maka foto tersebut akan dihapus dari halaman *portfolio*.
13. Mengelola *appointment reminder*

Proses pengelolaan *appointment reminder* dapat dilihat pada Gambar 4.11.



Gambar 4.11 *Activity Diagram*: Mengelola *appointment reminder*

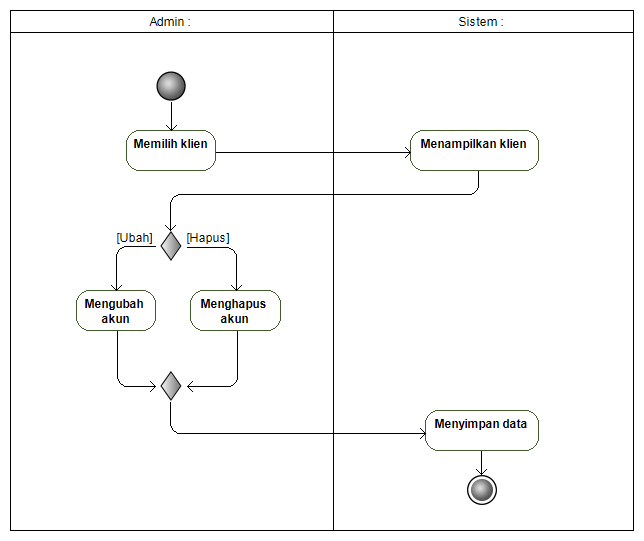
Proses pengelolaan *appointment reminder* terdiri dari langkah-langkah sebagai berikut:

1. EO memilih *appointment* pada halaman *dashboard*;
2. Sistem menampilkan halaman *appointment* berupa *appointment reminder* yang dimiliki oleh EO tersebut dengan berbagai klien;
3. EO memilih untuk menambahkan *appointment reminder*;
4. Sistem menampilkan formulir untuk melakukan penambahan *appointment reminder*;
5. EO mengisi data yang diperlukan berupa tanggal, waktu dan keterangan *appointment reminder*;
6. Jika data telah dimasukkan dengan benar, maka sistem akan menyimpan data tersebut.
7. EO memilih untuk mengubah *appointment reminder*;
8. EO memilih *appointment reminder* yang ingin diubah;
9. Sistem menampilkan halaman untuk melakukan perubahan *appointment reminder*;
10. EO mengubah status dari appointment reminder jika *appointment* telah terselesaikan;
11. Sistem menyimpan perubahan data.
12. EO memilih untuk menghapus *appointment reminder*;
13. EO memilih *appointment reminder* yang ingin dihapus;
14. Sistem menampilkan notifikasi berupa konfirmasi penghapusan *appointment reminder*;
15. Jika EO mengkonfirmasi, maka *appointment reminder* tersebut akan dihapus dari halaman *appointment*.
16. Mengelola akun

Proses pengelolaan akun yang dilakukan oleh *admin* dapat berupa akun klien dan akun EO.

1. Klien

Proses pengelolaan akun klien dapat dilihat pada Gambar 4.12.



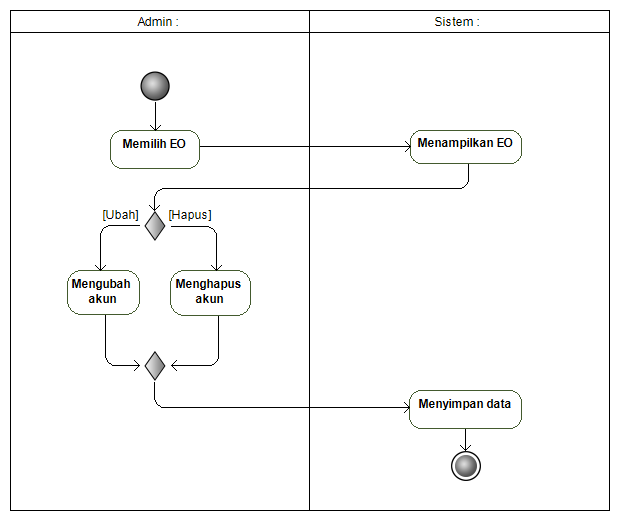
Gambar 4.12 *Activity Diagram*: Mengelola akun klien

Proses pengelolaan akun klien terdiri dari langkah-langkah sebagai berikut:

* + - 1. *Admin* memilih akun klien pada halaman *dashboard*;
      2. Sistem menampilkan halaman berupa akun-akun klien yang telah terdaftar;
      3. *Admin* memilih untuk mengubah akun klien;

1. *Admin* memilih akun klien yang ingin diubah;
2. Sistem menampilkan halaman untuk melakukan perubahan akun klien;
3. *Admin* dapat mengubah status dari akun klien tersebut;
4. Sistem menyimpan perubahan data.
   * + 1. *Admin* memilih untuk menghapus akun klien;
5. *Admin* memilih akun klien yang ingin dihapus;
6. Sistem menampilkan notifikasi berupa konfirmasi penghapusan akun klien;
7. Jika *admin* mengkonfirmasi, maka akun klien tersebut akan dihapus.
8. EO

Proses pengelolaan akun EO dapat dilihat pada Gambar 4.13.



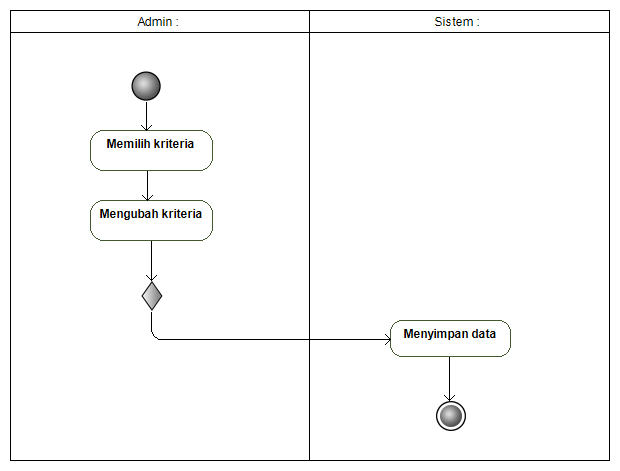
Gambar 4.13 *Activity Diagram*: Mengelola akun EO

Proses pengelolaan akun EO terdiri dari langkah-langkah sebagai berikut:

* + - 1. *Admin* memilih akun EO pada halaman *dashboard*;
      2. Sistem menampilkan halaman berupa akun-akun EO yang telah terdaftar;
      3. *Admin* memilih untuk mengubah akun EO;

1. *Admin* memilih akun EO yang ingin diubah;
2. Sistem menampilkan halaman untuk melakukan perubahan akun EO;
3. *Admin* dapat mengubah status dari akun EO tersebut;
4. Sistem menyimpan perubahan data.
   * + 1. *Admin* memilih untuk menghapus akun EO;
5. *Admin* memilih akun EO yang ingin dihapus;
6. Sistem menampilkan notifikasi berupa konfirmasi penghapusan akun EO;
7. Jika *admin* mengkonfirmasi, maka akun EO tersebut akan dihapus.
8. Mengelola kriteria

Proses pengelolaan kriteria dapat dilihat pada Gambar 4.14.

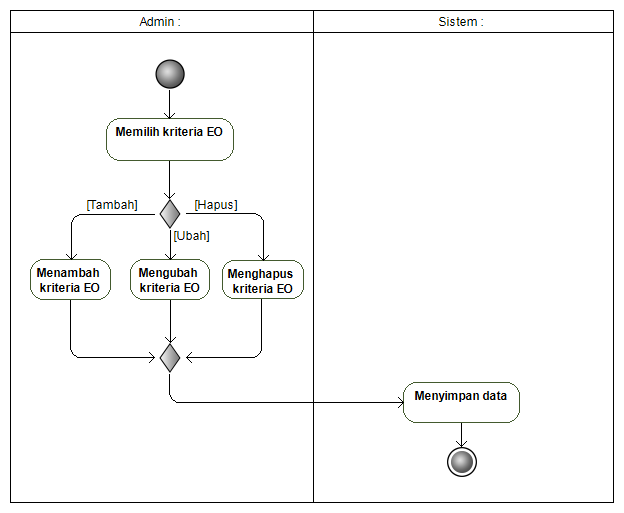


Gambar 4.14 *Activity Diagram*: Mengelola kriteria

Proses pengelolaan kriteria terdiri dari langkah-langkah sebagai berikut:

1. *Admin* memilih kriteria pada halaman *dashboard*;
2. Sistem menampilkan halaman berupa kriteria-kriteria yang terdaftar dalam sistem beserta detailnya;
3. *Admin* memilih untuk mengubah kriteria;
4. *Admin* memilih kriteria yang ingin diubah;
5. Sistem menampilkan halaman untuk melakukan perubahan kriteria;
6. *Admin* dapat mengubah bobot dari kriteria tersebut;
7. Sistem menyimpan perubahan data.
8. Mengelola data kriteria EO

Proses pengelolaan data kriteria EO dapat dilihat pada Gambar 4.15.



Gambar 4.15 *Activity Diagram*: Mengelola data kriteria EO

Proses pengelolaan data kriteria EO terdiri dari langkah-langkah sebagai berikut:

1. *Admin* memilih kriteria EO pada halaman *dashboard*;
2. Sistem menampilkan halaman berupa data kriteria dari berbagai EO yang telah terdaftar dalam sistem;
3. *Admin* memilih untuk menambah data kriteria EO;
4. Sistem menampilkan formulir untuk melakukan penambahan data kriteria EO;
5. *Admin* mengisi data yang diperlukan berupa EO beserta kriteria-kriteria yang dimiliki dengan benar;
6. Jika data telah dimasukkan dengan benar, maka sistem akan menyimpan data tersebut.
7. *Admin* memilih untuk mengubah data kriteria EO;
8. *Admin* memilih data kriteria EO yang ingin diubah;
9. Sistem menampilkan halaman untuk melakukan perubahan data kriteria EO;
10. *Admin* dapat mengubah kriteria yang dimiliki EO tersebut;
11. Sistem menyimpan perubahan data.
12. *Admin* memilih untuk menghapus data kriteria EO;
13. *Admin* memilih data kriteria EO yang ingin dihapus;
14. Sistem menampilkan notifikasi berupa konfirmasi penghapusan data kriteria EO;
15. Jika *admin* mengkonfirmasi, maka data kriteria EO tersebut akan dihapus.

### 4.2.4 Class Diagram

## 4.3 Tahap Perancangan

### 4.3.1 Perancangan Lapisan Manajemen Data

#### *4.3.1.1 Table Relationship Diagram*

#### *4.3.1.2 Deskripsi Tabel*

Berikut ini merupakan deskripsi dari setiap tabel yang terdapat dalam sistem usulan:

1. Tabel *app\_reminder*

Tabel *app\_reminder* digunakan untuk menyimpan data *appointment reminder* yang telah dibuat oleh EO. Deskripsi tabel *app\_reminder* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Deskripsi Tabel *app\_reminder*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_reminder | *Integer* | - | *Primary Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| id\_user | *Integer* | - | *Foreign Key* |
| tgl\_reminder | *Varchar* | 20 |  |
| wkt\_reminder | *Varchar* | 20 |  |
| ket\_reminder | *Text* |  |  |
| status | *Char* | 10 |  |

1. Tabel *bookmark*

Tabel *bookmark* digunakan untuk menyimpan data *bookmark* yang telah diberikan oleh klien terhadap EO. Deskripsi tabel *bookmark* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Deskripsi Tabel *bookmark*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_bookmark | *Integer* | - | *Primary Key* |
| id\_user | *Integer* | - | *Foreign Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| status | *Char* | 10 |  |

1. Tabel eo

Tabel eo digunakan untuk menyimpan data EO yang telah mendaftarkan diri melalui sistem. Deskripsi tabel eo dapat dilihat pada Tabel 4.5.

Tabel 4.5 Deskripsi Tabel eo

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_eo | *Integer* | - | *Primary Key* |
| email\_eo | *Varchar* | 50 | *Unique* |
| password\_eo | *Varchar* | 20 |  |
| foto\_eo | *Text* |  |  |
| nama\_eo | *Varchar* | 100 |  |
| ket\_eo | *Text* |  |  |
| id\_kota | *Integer* | - | *Foreign Key* |
| alamat\_eo | *Text* |  |  |
| nohp\_eo | *Char* | 15 |  |
| foto\_ktp | *Text* |  |  |
| fotodiri\_ktp | *Text* |  |  |
| foto\_alamat | *Text* |  |  |
| foto\_siup | *Text* |  |  |
| tahun\_diri | *Char* | 5 |  |
| link\_web | *Varchar* | 100 |  |
| status | *Char* | 10 |  |

1. Tabel indikator\_penilaian

Tabel indikator\_penilaian digunakan untuk menyimpan data penilaian dari logika *fuzzy*. Deskripsi tabel indikator\_penilaian dapat dilihat pada Tabel 4.6.

Tabel 4.6 Deskripsi Tabel indikator\_penilaian

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_indikator\_penilaian | *Integer* | - | *Primary Key* |
| indikator\_penilaian | *Varchar* | 20 |  |
| nilai | *Float* |  |  |

1. Tabel kategori

Tabel kategori digunakan untuk menyimpan data kategori yang ada dalam sistem. Deskripsi tabel kategori dapat dilihat pada Tabel 4.7.

Tabel 4.7 Deskripsi Tabel kategori

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kategori | *Integer* | - | *Primary Key* |
| nama\_kategori | *Varchar* | 100 |  |

1. Tabel kategori\_eo

Tabel kategori\_eo digunakan untuk menyimpan data EO beserta kategori yang dimilikinya. Deskripsi tabel kategori\_eo dapat dilihat pada Tabel 4.8.

Tabel 4.8 Deskripsi Tabel kategori\_eo

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_eo | *Integer* | - | *Foreign Key* |
| id\_kategori | *Integer* | - | *Foreign Key* |

1. Tabel kota

Tabel kota digunakan untuk menyimpan data berbagai kota yang ada di Indonesia. Deskripsi tabel kota dapat dilihat pada Tabel 4.9.

Tabel 4.9 Deskripsi Tabel kota

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kota | *Integer* | - | *Primary Key* |
| nama\_kota | *Varchar* | 50 |  |
| id\_provinsi | *Integer* | - | *Foreign Key* |

1. Tabel kriteria

Tabel kriteria digunakan untuk menyimpan data kriteria yang ada dalam sistem. Deskripsi tabel kriteria dapat dilihat pada Tabel 4.10.

Tabel 4.10 Deskripsi Tabel kriteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria | *Integer* | - | *Primary Key* |
| nama\_kriteria | *Varchar* | 20 |  |
| jenis\_kriteria | *Char* | 10 |  |

1. Tabel kriteria\_budget

Tabel kriteria\_budget digunakan untuk menyimpan data mengenai kriteria *budget* beserta detail kriteria dan nilai yang dimiliki. Deskripsi tabel kriteria\_budget dapat dilihat pada Tabel 4.11.

Tabel 4.11 Deskripsi Tabel kriteria\_budget

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria\_budget | *Integer* | - | *Primary Key* |
| id\_kriteria | *Integer* | - | *Foreign Key* |
| detail\_kriteria | *Varchar* | 30 |  |
| id\_indikator\_penilaian | *Integer* | - | *Foreign Key* |

1. Tabel kriteria\_eo

Tabel kriteria\_eo digunakan untuk menyimpan data EO beserta kriteria-kriteria yang dimilikinya. Deskripsi tabel kriteria\_eo dapat dilihat pada Tabel 4.12.

Tabel 4.12 Deskripsi Tabel kriteria\_eo

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria\_eo | *Integer* | - | *Primary Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| id\_kriteria\_budget | *Integer* | - | *Foreign Key* |
| id\_kriteria\_reputasi | *Integer* | - | *Foreign Key* |
| id\_kriteria\_fasilitas | *Integer* | - | *Foreign Key* |
| id\_kriteria\_konsep | *Integer* | - | *Foreign Key* |

1. Tabel kriteria\_fasilitas

Tabel kriteria\_fasilitas digunakan untuk menyimpan data mengenai kriteria fasilitas beserta detail kriteria dan nilai yang dimiliki. Deskripsi tabel kriteria\_fasilitas dapat dilihat pada Tabel 4.13.

Tabel 4.13 Deskripsi Tabel kriteria\_fasilitas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria\_fasilitas | *Integer* | - | *Primary Key* |
| id\_kriteria | *Integer* | - | *Foreign Key* |
| detail\_kriteria | *Varchar* | 30 |  |
| id\_indikator\_penilaian | *Integer* | - | *Foreign Key* |

1. Tabel kriteria\_konsep

Tabel kriteria\_konsep digunakan untuk menyimpan data mengenai kriteria konsep beserta detail kriteria dan nilai yang dimiliki. Deskripsi tabel kriteria\_konsep dapat dilihat pada Tabel 4.14.

Tabel 4.14 Deskripsi Tabel kriteria\_konsep

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria\_konsep | *Integer* | - | *Primary Key* |
| id\_kriteria | *Integer* | - | *Foreign Key* |
| detail\_kriteria | *Varchar* | 30 |  |
| id\_indikator\_penilaian | *Integer* | - | *Foreign Key* |

1. Tabel kriteria\_reputasi

Tabel kriteria\_reputasi digunakan untuk menyimpan data mengenai kriteria reputasi beserta detail kriteria dan nilai yang dimiliki. Deskripsi tabel kriteria\_reputasi dapat dilihat pada Tabel 4.15.

Tabel 4.15 Deskripsi Tabel kriteria\_reputasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_kriteria\_reputasi | *Integer* | - | *Primary Key* |
| id\_kriteria | *Integer* | - | *Foreign Key* |
| detail\_kriteria | *Varchar* | 30 |  |
| id\_indikator\_penilaian | *Integer* | - | *Foreign Key* |

1. Tabel paket

Tabel paket digunakan untuk menyimpan data paket yang telah disediakan oleh berbagai EO. Deskripsi tabel paket dapat dilihat pada Tabel 4.16.

Tabel 4.16 Deskripsi Tabel paket

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_paket | *Integer* | - | *Primary Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| nama\_paket | *Varchar* | 50 |  |
| jenis\_paket | *Varchar* | 50 |  |
| harga\_paket | *BigInteger* | 20 |  |
| ket\_paket | *Text* |  |  |

1. Tabel pesan

Tabel pesan digunakan untuk menyimpan data pesan yang telah dikirimkan oleh klien maupun EO. Deskripsi tabel pesan dapat dilihat pada Tabel 4.17.

Tabel 4.17 Deskripsi Tabel pesan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_pesan | *Integer* | - | *Primary Key* |
| id\_user | *Integer* | - | *Foreign Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| tgl\_pesan | *Varchar* | 50 |  |
| subjek | *Varchar* | 100 |  |
| pesan | *Text* |  |  |
| status | *Char* | 10 |  |
| sender | *Char* | 10 |  |

1. Tabel *portfolio*

Tabel *portfolio* digunakan untuk menyimpan data *portfolio* yang dimiliki oleh berbagai EO. Deskripsi tabel *portfolio* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Deskripsi Tabel *portfolio*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_portfolio | *Integer* | - | *Primary Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| foto | *Text* |  |  |
| ket\_foto | *Text* |  |  |

1. Tabel provinsi

Tabel provinsi digunakan untuk menyimpan data berbagai provinsi yang ada di Indonesia. Deskripsi tabel provinsi dapat dilihat pada Tabel 4.19.

Tabel 4.19 Deskripsi Tabel provinsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_provinsi | *Integer* | - | *Primary Key* |
| nama\_provinsi | *Varchar* | 50 |  |

1. Tabel request\_layanan

Tabel request\_layanan digunakan untuk menyimpan data *request* layanan yang telah diajukan oleh klien terhadap EO. Deskripsi tabel request\_layanan dapat dilihat pada Tabel 4.20.

Tabel 4.20 Deskripsi Tabel request\_layanan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_request | *Integer* | - | *Primary Key* |
| id\_user | *Integer* | - | *Foreign Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| tgl\_request | *Varchar* | 50 |  |
| tgl\_acara | *Varchar* | 50 |  |
| tipe\_acara | *Varchar* | 100 |  |
| lokasi\_acara | *Text* |  |  |
| jml\_peserta | *Integer* | 10 |  |
| durasi\_acara | *Varchar* | 50 |  |
| ket\_acara | *Text* |  |  |
| id\_paket | *Integer* | - | *Foreign Key* |
| status | *Char* | 10 |  |

1. Tabel *review*

Tabel *review* digunakan untuk menyimpan data *review* yang telah diberikan oleh klien terhadap EO. Deskripsi tabel *review* dapat dilihat pada Tabel 4.21.

Tabel 4.21 Deskripsi Tabel *review*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_user | *Integer* | - | *Foreign Key* |
| id\_eo | *Integer* | - | *Foreign Key* |
| tgl\_review | *Varchar* | 20 |  |
| rating | *TinyInteger* | 5 |  |
| keterangan | *Text* |  |  |

1. Tabel *user*

Tabel *user* digunakan untuk menyimpan data *user* yang telah mendaftarkan diri melalui sistem. Deskripsi tabel *user* dapat dilihat pada Tabel 4.22.

Tabel 4.22 Deskripsi Tabel *user*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | **Tipe Data** | ***Length*** | ***Key*** |
| id\_user | *Integer* | - | *Primary Key* |
| nama\_user | *Varchar* | 100 |  |
| nohp\_user | *Char* | 15 |  |
| email\_user | *Varchar* | 50 | *Unique* |
| password\_user | *Varchar* | 20 |  |
| foto\_user | *Text* |  |  |
| role | *Char* | 10 |  |
| status | *Char* | 10 |  |

### Perancangan Simple Additive Weighting (SAW) dan Fuzzy Logic

Metode *Simple Additive Weighting* adalah sebuah metode yang dikenal dengan istilah metode penjumlahan terbobot dan merupakan salah satu penyelesaian dari metode *Fuzzy Multiple Attribute Decision Making* (FMADM). Metode FMADM ini merupakan metode yang menggunakan logika *Fuzzy* dengan tujuan untuk menghasilkan sebuah alternatif yang optimal dari berbagai alternatif yang ada dengan kriteria-kriteria tertentu. Penggunaan metode SAW dengan logika *Fuzzy* ini dapat membantu mempertemukan klien dan EO dengan memberikan rekomendasi kandidat EO yang sesuai dengan preferensi klien dari kriteria-kriteria yang telah ditentukan.

Konsep dasar dari penggunaan metode SAW adalah menghasilkan sebuah penjumlahan terbobot berdasarkan *rating* kinerja pada setiap alternatif beserta atribut yang dimilikinya. Metode SAW membutuhkan sebuah proses berupa normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua *rating* alternatif yang ada. Alternatif yang dimaksud dalam sistem usulan ini adalah para EO. Berikut adalah langkah-langkah dalam menggunakan metode SAW:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan beserta jenisnya

Berikut adalah empat kriteria wajib yang telah ditetapkan dalam proses pemilihan kandidat EO yang dapat dilihat pada Tabel 4.23.

Tabel 4.23 Tabel Kriteria

|  |  |
| --- | --- |
| **Kriteria** | **Jenis** |
| *Budget* (C1) | *Cost* |
| Reputasi (C2) | *Benefit* |
| Fasilitas dan layanan (C3) | *Benefit* |
| Konsep dan dekorasi (C4) | *Benefit* |

1. Menentukan nilai bobot dan fungsi keanggotaan *fuzzy* pada setiap kriteria

Setiap kriteria yang telah ditetapkan akan diberi nilai bobot dan fungsi keanggotaan *fuzzy*. Nilai yang akan diberi berdasarkan *range* dari 0 hingga 1, dimana 0 merupakan nilai terendah dan 1 merupakan nilai tertinggi. Pemberian nilai tersebut disesuaikan dengan teori logika *fuzzy*. Pembagian bobot untuk kriteria *budget* akan dibagi menjadi lima bagian sedangkan untuk kriteria reputasi, fasilitas dan layanan, konsep dan dekorasi akan dibagi menjadi empat bagian untuk mempermudah admin dalam memberikan penilaian *fuzzy*. Berikut adalah tabel-tabel yang merangkum pemberian nilai pada setiap kriteria:

1. *Budget*

Pembagian bobot kriteria *budget* dilihat berdasarkan harga dari paket yang disediakan oleh setiap alternatif (EO). Bobot kriteria *budget* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Bobot *Budget*

|  |  |  |
| --- | --- | --- |
| ***Budget*** | **Bilangan *Fuzzy*** | **Nilai** |
| < 5.000.000 | Sangat Rendah | 0.2 |
| 5.000.000 – 10.000.000 | Rendah | 0.4 |
| > 10.000.000 – 20.000.000 | Cukup | 0.6 |
| > 20.000.000 – 30.000.000 | Tinggi | 0.8 |
| > 30.000.000 | Sangat Tinggi | 1 |

1. Reputasi

Pembagian bobot kriteria reputasi dilihat berdasarkan *rating* dari setiap alternatif (EO). Bobot kriteria reputasi dapat dilihat pada Tabel 4.25.

Tabel 4.25 Bobot Reputasi

|  |  |  |
| --- | --- | --- |
| **Reputasi** | **Bilangan *Fuzzy*** | **Nilai** |
| 0 – 1.49 | Rendah | 0.4 |
| 1.5 – 2.99 | Cukup | 0.6 |
| 3.0 – 4.49 | Tinggi | 0.8 |
| 4.5 – 5.0 | Sangat Tinggi | 1 |

1. Fasilitas dan layanan

Pembagian bobot kriteria fasilitas dan layanan dilihat berdasarkan banyaknya jenis fasilitas dan layanan yang disediakan oleh setiap alternatif (EO). Bobot kriteria fasilitas dan layanan dapat dilihat pada Tabel 4.26.

Tabel 4.26 Bobot Fasilitas dan Layanan

|  |  |  |
| --- | --- | --- |
| **Fasilitas dan layanan** | **Bilangan *Fuzzy*** | **Nilai** |
| Tidak Banyak | Rendah | 0.4 |
| Cukup | Cukup | 0.6 |
| Banyak | Tinggi | 0.8 |
| Sangat Banyak | Sangat Tinggi | 1 |

1. Konsep dan dekorasi

Pembagian bobot kriteria konsep dan dekorasi dilihat berdasarkan konsep dan dekorasi pada *portfolio* dari setiap alternatif (EO). Bobot kriteria konsep dan dekorasi dapat dilihat pada Tabel 4.27.

Tabel 4.27 Bobot Konsep dan Dekorasi

|  |  |  |
| --- | --- | --- |
| **Konsep dan dekorasi** | **Bilangan *Fuzzy*** | **Nilai** |
| Kurang Baik | Rendah | 0.4 |
| Cukup Baik | Cukup | 0.6 |
| Baik | Tinggi | 0.8 |
| Sangat Baik | Sangat Tinggi | 1 |

1. Menentukan rating kecocokan dari berbagai kriteria pada setiap alternatif

Setelah menyelesaikan pembagian bobot setiap kriteria, langkah selanjutnya adalah menyesuaikan dan mengkonversikan kriteria EO dengan nilai bobot yang telah ditentukan. Sebagai contoh, terdapat lima EO dengan kategori yang sama yang telah dipilih oleh klien. Berikut adalah data kriteria dari masing-masing EO tersebut:

Tabel 4.28 Kriteria Masing-masing EO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Budget*** | **Reputasi** | **Fasilitas dan layanan** | **Konsep dan dekorasi** |
| EO 1 | 3.000.000 | 3.0 | Cukup | Sangat Baik |
| EO 2 | 12.000.000 | 4.6 | Banyak | Baik |
| EO 3 | 5.000.000 | 1.4 | Sangat Banyak | Cukup Baik |
| EO 4 | 32.000.000 | 4.2 | Tidak Banyak | Baik |
| EO 5 | 21.000.000 | 2.5 | Banyak | Kurang Baik |

Tabel 4.29 menunjukkan kriteria-kriteria EO yang telah disesuaikan dengan nilai bobot yang telah ditentukan sebelumnya. Berikut hasil dari konversi tersebut:

Tabel 4.29 Konversi dari Kriteria EO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** |
| A1 | 0.2 | 0.8 | 0.6 | 1 |
| A2 | 0.6 | 1 | 0.8 | 0.8 |
| A3 | 0.4 | 0.4 | 1 | 0.6 |
| A4 | 1 | 0.8 | 0.4 | 0.8 |
| A5 | 0.8 | 0.6 | 0.8 | 0.4 |

Keterangan:

* A1 – A5 = Alternatif (EO)
* C1 – C4 = Kriteria

1. Membentuk matriks keputusan (X)

Setelah melakukan konversi, langkah selanjutnya adalah menyusun nilai-nilai tersebut ke dalam matriks sehingga terbentuk matriks keputusan (X).

Matriks X =

1. Menentukan bobot preferensi dari setiap kriteria (W)

Selanjutnya adalah proses penentuan bobot untuk setiap kriteria oleh pembuat keputusan (klien). Bobot tersebut berupa persentase kepentingan yang akan diberikan untuk setiap kriteria dari kandidat EO yang dicari oleh klien dan totalnya harus mencapai 100%. Semakin tinggi persentase yang diberikan, maka semakin penting kriteria tersebut. Sebagai contoh, Klien A lebih mementingkan *budget* daripada kriteria lain dan reputasi memiliki bobot paling rendah maka pembobotannya adalah sebagai berikut:

1. *Budget* = 50%
2. Reputasi = 10%
3. Fasilitas dan layanan = 20%
4. Konsep dan dekorasi = 20%

Vektor bobot (W) =

1. Melakukan normalisasi matriks keputusan (X)

Tahap selanjutnya adalah melakukan perhitungan normalisasi dengan rumus SAW yang sesuai dengan jenis dari setiap kriteria. Dari empat kriteria yang telah ditetapkan diatas, terdapat satu kriteria yang termasuk atribut biaya (*cost*) yaitu kriteria *budget* sedangkan tiga kriteria lainnya termasuk atribut keuntungan (*benefit*). Untuk melakukan perhitungan kriteria dengan atribut biaya (*cost*), rumus yang dipakai adalah:

rij =

Berikut contoh perhitungan pada kriteria *budget*:

R11 = = 1

R12 = = 0.33

R13 = = 0.5

R14 = = 0.2

R15 = = 0.25

Sedangkan untuk melakukan perhitungan kriteria dengan atribut keuntungan (*benefit*), rumus yang dipakai adalah:

rij =

Berikut salah satu contoh perhitungan pada kriteria reputasi:

R21 = = 0.8

R22 = = 1

R23 = = 0.4

R24 = = 0.8

R25 = = 0.6

Hasil matriks ternormalisasi R sebagai berikut:

Matriks R =

1. Menghasilkan nilai preferensi atau peringkat dari setiap alternatif (V)

Tahap terakhir dalam menghasilkan sebuah peringkat, diperlukan untuk melakukan perhitungan yang melibatkan perkalian antara matriks R dan nilai bobot (W) yang telah ditentukan sebelumnya. Rumus yang digunakan adalah:

Vi =

Berikut hasil perhitungan dengan menggunakan rumus tersebut:

V1 = = 0.9

V2 = = 0.59

V3 = = 0.61

V4 = = 0.42

V5 = = 0.43

Dari hasil perhitungan diatas, diperoleh peringkat dari setiap alternatif (EO) yang kemudian diurutkan berdasarkan nilai yang terbesar. Nilai Vi yang lebih besar mengindikasikan bahwa Ai lebih direkomendasikan. Hasil peringkat dapat dilihat pada Tabel 4.30.

Tabel 4.30 Hasil Peringkat

|  |  |  |
| --- | --- | --- |
| **Alternatif** | **Total Nilai** | **Peringkat** |
| EO 1 | 0.9 | 1 |
| EO 3 | 0.61 | 2 |
| EO 2 | 0.59 | 3 |
| EO 5 | 0.43 | 4 |
| EO 4 | 0.42 | 5 |

### 4.3.3 Perancangan Lapisan Antarmuka

# BAB V KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

<Times New Roman 12, spasi 1.5pt, alignment: justify>

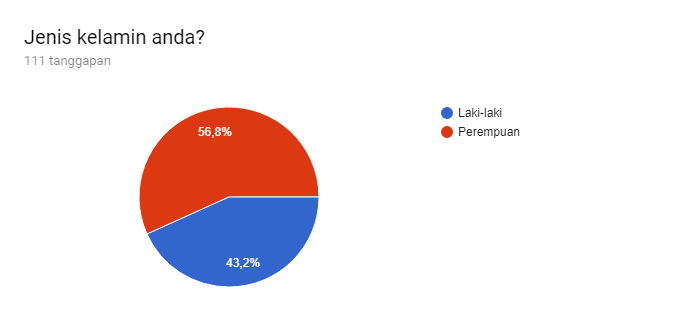
## 5.2 Saran

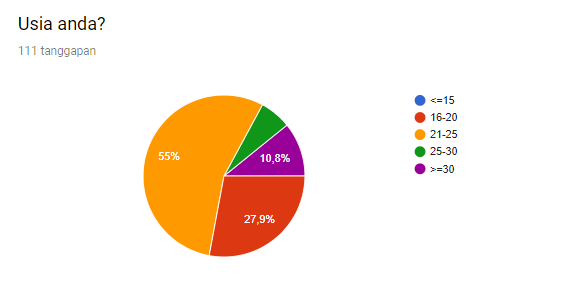
<Times New Roman 12, spasi 1.5pt, alignment: justify>

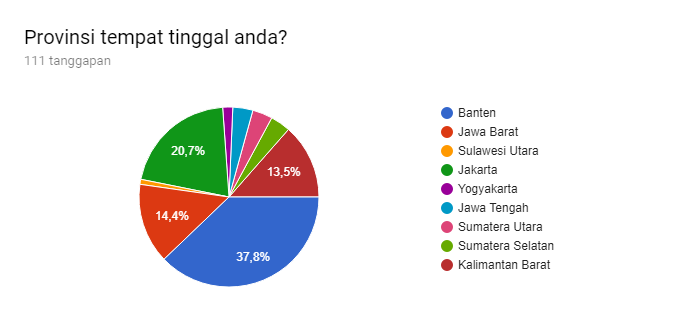
# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | I. K. Suseno, Cara Pinter Jadi Event Organizer, Yogyakarta: Galang Press, 2005. |
| [2] | A. Dennis, B. H. Wixom and D. Tegarden, System Analysis & Design: An Object-Oriented Approach with UML, 5th Edition, USA: Wiley, 2015. |
| [3] | B. Shneiderman and C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th Edition, Boston: Pearson, 2004. |
| [4] | T. Connolly and C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition, England: Pearson, 2015. |
| [5] | R. Nixon, Learning PHP, MySQL, and JavaScript, 4th Edition, USA: O'Reilly, 2015. |
| [6] | "What can PHP do?," PHP, 2001. [Online]. Available: http://php.net/manual/en/intro-whatcando.php. [Accessed 20 Juli 2018]. |
| [7] | K. Tatroe, P. MacIntyre and R. Lerdorf, Programming PHP, 3rd Edition, USA: O'Reilly, 2013. |
| [8] | "What is JavaScript?," MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/What\_is\_JavaScript. [Accessed 20 Juli 2018]. |
| [9] | S. Nidhra and J. Dondeti, "Black Box and White Box Testing: A Literature Review," *International Journal of Embedded Systems and Applications,* vol. II, no. 2, pp. 1-22, 2012. |
| [10] | S. K. Boel and D. C. Kecmanovic, "What is an Information System?," *Hawaii International Conference on System Sciences,* no. 48, pp. 1-11, 2015. |
| [11] | A. Noor, Manajemen Event, Bandung: Alfabeta, 2013. |
| [12] | J. Goldblatt, Special Events: Creating and Sustaining a New World for Celebration, New York: Wiley, 2013. |

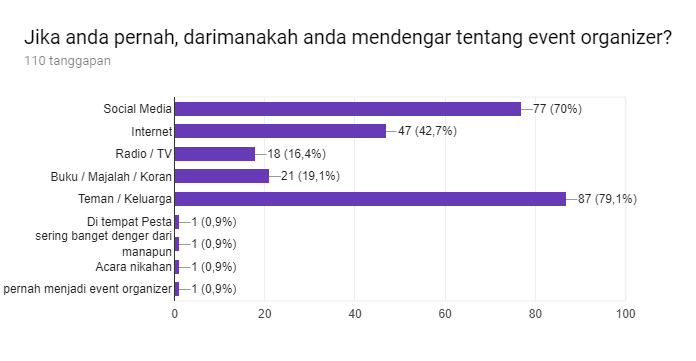
LAMPIRAN 1  
HASIL SURVEI

****

****

****

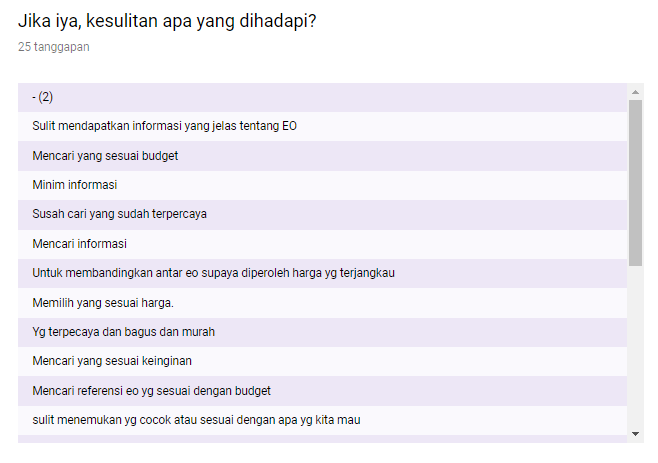
****

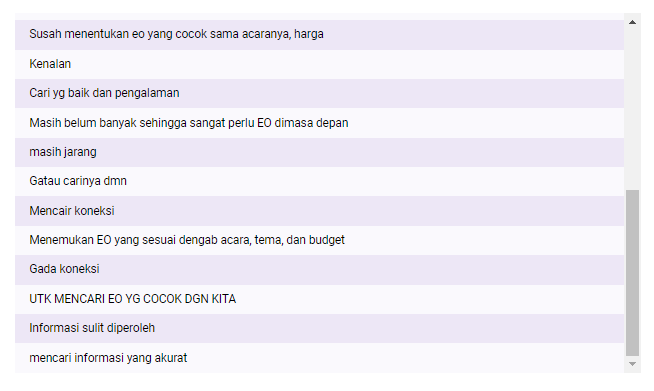
****

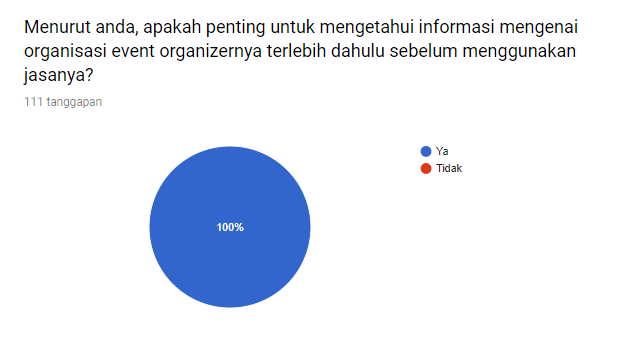
****

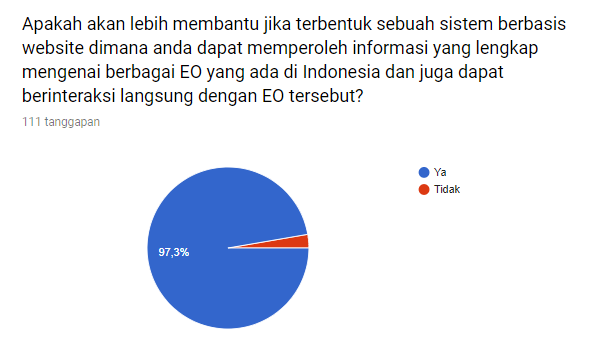
****

****









****

