**SKPL** - 001

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

APE (APLIKASI PENCARIAN *EVENT*)

untuk:

DIBOY DIMBOY

Dipersiapkan oleh:

|  |  |  |
| --- | --- | --- |
| Dimas Bayu Nugraha | 1301181096 | Project Leader |
| Rivan Nur Ihsan | 1301180081 | Analyst |
| Alif Ranadian | 1301184255 | Analyst |
| Muhammad Rafif | 1301184124 | Designer |

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  -  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *SKPL-001* | | 35 |
| Revisi | *-* | *Tgl: 6 MARET 2020* |

# Daftar Perubahan

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | * Merivisi Class Diagram menjadi lebih lengkap * Merivisi ERD menjadi lebih lengkap * Merubah penamaan functional requirement dari FR-LG menjadi FR-01 dan seterusnya * Penambahan use case scenario update password |
| B | * Perubahan penamaan dari kata Register/Registrasi menjadi Registrasi Akun * Merevisi Use Case Diagram pada bagian use case Register menjadi Registrasi Akun dan use case Baca Validasi Event menjadi Lihat Hasil Validasi Event |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| TGL |  |  |  |  |  |  |  |  |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

# Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

# Daftar Isi

[Daftar Perubahan 1](#_Toc38229176)

[Daftar Halaman Perubahan 2](#_Toc38229177)

[Daftar Isi 3](#_Toc38229178)

[1. Pendahuluan 4](#_Toc38229179)

[1.1 Tujuan Penulisan Dokumen 4](#_Toc38229180)

[1.2 Ruang Lingkup / Cakupan Dokumen 4](#_Toc38229181)

[1.3 Definisi, Singkatan, dan Akronim 4](#_Toc38229182)

[1.4 Referensi 5](#_Toc38229183)

[2. Deskripsi Global Perangkat Lunak 6](#_Toc38229184)

[2.1 Statement of Objective Perangkat Lunak 6](#_Toc38229185)

[2.2 Perspektif dan Fungsi Perangkat Lunak 6](#_Toc38229186)

[2.3 Profil dan Karakteristik Pengguna 7](#_Toc38229187)

[2.4 Lingkungan Operasi 8](#_Toc38229188)

[2.5 Batasan Perangkat Lunak / Sistem 8](#_Toc38229189)

[2.6 Asumsi dan Dependensi 8](#_Toc38229190)

[3. Deskrpsi Rinci Perangkat Lunak 10](#_Toc38229191)

[3.1 Deskripsi Kebutuhan 10](#_Toc38229192)

[3.1.1 Kebutuhan Fungsional 10](#_Toc38229193)

[3.1.2 Kebutuhan Non-Fungsional 11](#_Toc38229194)

[3.2 Pemodelan Analisis 11](#_Toc38229195)

[3.2.1 Usecase Diagram 11](#_Toc38229196)

[3.2.2 Usecase Scenario 12](#_Toc38229197)

[3.2.3 Class Diagram 29](#_Toc38229198)

[4. Kebutuhan Antarmuka Eksternal 30](#_Toc38229199)

[4.1 Antarmuka Pengguna 30](#_Toc38229200)

[4.2 Antarmuka Perangkat Keras 31](#_Toc38229201)

[4.3 Antarmuka Perangkat Lunak 32](#_Toc38229202)

[4.4 Antarmuka Komunikasi 32](#_Toc38229203)

[5. Requirements Lain 33](#_Toc38229204)

[5.1 Database 33](#_Toc38229205)

[5.2 MySQL 33](#_Toc38229206)

[5.3 Apache 33](#_Toc38229207)

[Lampiran A: Daftar Kata-Kata Sukar 34](#_Toc38229208)

[Lampiran B: Analysis Models 35](#_Toc38229209)

# Pendahuluan

## Tujuan Penulisan Dokumen

Isi dari dokumen ini merupakan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau dengan nama lain Software Requirement Specification (SRS) beserta penjelsannya. Adapun SKPL ini menggunakan pendekatan berorientasi proses dalam perangkat lunak yang akan dibuat.

## Ruang Lingkup / Cakupan Dokumen

Penulisn dokumen SKPL ini memiliki tujuan yaitu menjelaskan tahapan proses yang terjadi dalam pembuatan perangkat lunak yang akan dibangun. Dari sudut pandang pihak pengembang, SKPL ini dapat digunakan sebahai acuan dalam pengembangan perangkat lunak secara detail sehingga perangkat lunak yang dibangun nantinya sesuai dengan kebutuhan yang dibutuhkan oleh pengguna (*user*). Dengan adanya dokumen SKPL ini diharapkan dalam pengembangan perangkat lunak dapat lebih fokus terhadap tujuan dari pengembangan perangkat lunak itu sendiri. Sedangkan bila dilihat dari sudut pandang pengguna (*user*), dokumen SKPL ini dapat digunakan untuk mencatat spesifikasi kebutuhan perangkat lunak yang akan dibangun dan harapan kebutuhan yang diinginkan pengguna untuk perangkat lunak yang akan dibangu ini.

## Definisi, Singkatan, dan Akronim

Berikut adalah daftar definisi dan istilah penting yang digunakan dalam dokumen SKPL ini:

* SKPL: Spesifikasi Kebutuhan Perangkat Lunak Dokumen hasil analisis yang berisi kebutuhan perangkat lunak.
* IEEE: *Institute of Electrical and Electronics Engineering*, standar inteernasional untuk pengembangan dan perancangan produk.
* ANSI: *American National Standard Institute* adalah Lembaga Standardisasi di Amerika.
* *Use Case Diagram*: Gambaran grafik dari beberapa atau semua aktor, *Use case* dan interaksinya.
* TBD: *To be defined*.
* ERD*: Entity Relational Diagram*
* APPL: Analisan dan Perancangan Perangkat Lunak
* Website: Suatu halaman web yang saling berhubungan yang umum nya berisikan kumpulan informasi berupa data teks, gambar, video, audio.
* Internet*: Interconnection-networking* merupakan seluruh jaringan komputer yang saling terhubung menggunakan standar *global Transmission Control Protocol/Internet Protocol* (TCP/IP) sebagai protokol pertukaran paket untuk melayani miliaran pengguna di seluruh dunia.
* Database: Memiliki arti Pangkalan Data merupakan kumpulan informasi yang disimpan menjadi satu di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.
* *Username*: Sebuah identitas berupa nama yang dibuat oleh pengguna, diisi disaat hendak melakukan proses registrasi atau *log in* akun. Biasanya setiap individu memiliki *username* yang berbeda-beda untuk masuk ke sistem.
* *Password*: Sebuah identitas yang digunakan untuk pengamanan akun antar user ke sistem. Password sendiri memiliki kombinasi yang berbeda untuk setiap pengguna nya tergantung dari pengguna inputkan.

## Referensi

* Vikri, C. (2017). Dokumen Spesifikasi Kebutuhan Perangkat Lunak. Bandung: Prodi Informatika, Universitas Telkom.
* Kusumayadhi, N.I. (2016). Dokumen Spesifikasi Kebutuhan Perangkat Lunak. Bandung: Prodi Informatika, Universitas Telkom.

# Deskripsi Global Perangkat Lunak

## Statement of Objective Perangkat Lunak

Perangkat lunak yang dibuat dalam SKPL ini merupakan sebuah perangkat lunak dari sistem yang sudah ada, yaitu sistem pencarian *event* yang terdapat pada web Eventbrite.com, dimana web tersebut menyediakan fitur utama yaitu pencarian *event* dan mendaftar *event* yang ingin diikuti dengan cakupan user bersifat umum. Perangkat lunak Aplikasi Pencarian *Event* (APE) ini merupakan perangkat lunak berbasis aplikasi yang diakses melalui website yang digunakan untuk mempermudah pencarian *event* yang ada disekitar Telkom University. Dalam aplikasi ini meliputi Pembuatan *Event*, Pencarian *Event*, Informasi *Event*, *Join* *Event*, dan lain-lain.

Dengan adanya sistem pencarian *event* berbasis web di sekitaran lingkungan kampus Telkom University ini diharapkan Mahasiswa dapat lebih mudah mengetahui informasi tentang *event*-*event* apa saja yang berlangsung di sekitaran lingkungan Telkom University untuk diikuti agar dapat menambah poin Transkrip Keaktifan Mahasiswa, serta diharapkan Pembuat *Event* lebih mudah dalam mengadakan *event* dan melakukan promosi *event* yang diadakan serta dapat diketahui oleh banyak masyarakat di sekitar lingkungan Telkom University.

## Perspektif dan Fungsi Perangkat Lunak

Perangkat lunak berbasis *website* ini dibuat berdasarkan dari permasalahan yang dihadapi oleh *client* (Diboy Dimboy), diantaranya adalah :

1. *Client* kesulitan dalam mencari informasi tentang *event*-*event* yang diadakan di sekitaran lingkungan Telkom University.
2. *Client* kesulitan mencari informasi *event* di media social, karena informasi yang diterima di media sosial terlalu distraktif sehingga menjadi tidak fokus dalam melakukan pencarian.

Sehingga dengan dibangunnya Aplikasi Pencarian *Event* berbasis *web* ini, diharapkan dapat membantu semua pihak yang terkait dalam Aplikasi Pencarian *Event* agar menjadi lebih mudah dalam mencari *event* maupun membuat *event*. Perangkat lunak ini memiliki beberapa fungsi utama, antara lain:

* Registrasi Akun
* *Login*
* *Update* *Password*
* *Update* *Profile*
* *Input* *Event*
* *Edit* *Event*
* Hapus *Event*
* *Input* validasi *Event*
* Lihat hasil validasi *Event*
* Lihat *Event*
* Cari *Event*
* *Join* *Event*
* Buat *Form* *event*
* *Input* *Form* *event*
* *Edit* *Form* *event*
* Hapus *Form* *event*

## Profil dan Karakteristik Pengguna

**Tabel 1 Karakteristik Pengguna**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategori Pengguna** | **Tugas** | **Hak Akses ke Aplikasi** | **Kemampuan yang harus dimiliki** |
| Admin | Memvalidasi *event* yang telah dibuat oleh Pembuat *Event* | * *Login* * *Update* *Profile* * *Update* *Password* * Lihat *Event* * Kirim hasil validasi *event* | * Mengoperasikan computer dan aplikasi * Kemampuan menganalisis kebenaran dari suatu *event* yang telah dibuat untuk di validasi. |
| Pembuat *Event* | Membuat *event* yang ingin diselenggarakan dan membuat *Form* pendaftaran *event* jika perlu | * *Login* * *Update* *Profile* * *Update* *Password* * *Input* *Event* * *Edit* *Event* * Hapus *Event* * Lihat hasil validasi *event* * Buat *Form* *Event* * Edit *Form* *Event* * Hapus *Form* *Event* | * Mengoperasikan computer dan aplikasi |
| Mahasiswa | Melihat mencari dan mengikuti dari daftar-daftar *event* yang ditampilkan dari pembuat *event* | * *Login* * Update Profile * Update Password * Lihat *Event* * Mencari *Event* * *Join* *Event* * Input *Form* *Event* | * Mengoperasikan computer dan aplikasi |

## Lingkungan Operasi

Perangkat Lunak yang dibuat dalam SKPL ini akan digunakan di sekitaran lingkungan kampus Telkom University dengan implementasi aplikasi berbasis web, sehingga perangkat lunak dapat diakses kapan saja, dimana saja, dan oleh siapa saja. Mahasiswa dan Pembuat *Event* merupakan *actor* utama yang nantinya dapat berhubungan langsung dengan aplikasi ini. Oleh karena itu, aplikasi ini dapat diakses menggunakan perangkat keras seperti laptop, computer, maupun smartphone dengan sistem operasi Windows XP, Windows 7, Windows 10, Mac OS, iOS, sampai Android dengan bantuan browser seperti Mozilla Firefox, Microsoft Edge, Opera, Google Chrome, UC Browser, dan browser setara lainnya.

## Batasan Perangkat Lunak / Sistem

1. Aplikasi Pencarian *Event* ini dilengkapi dengan penggunaan Bahasa Indonesia hampir di setiap bagian tampilan Aplikasi, adapun beberapa istilah dalam Bahasa asing itu sudah dibakukan.
2. Pada Aplikasi Pencarian *Event* ini hanya menampilkan informasi *event* dan melakukan *Form* registrasi *event* jika diperlukan (*optional*) tidak sampai melakukan transaksi pembayaran untuk mengikuti *event* tersebut, karena itu sudah diluar dari batasan sistem.

## Asumsi dan Dependensi

Asumsi:

* Sistem Aplikasi Pencarian Event berbasis web dapat di akses oleh siapapun, dimanapun, dan kapanpun dengan syarat utama harus terhubung dengan jaringan internet ;
* Data informasi dapat di akses secara realtime ;
* Pembuat Event dapat mengetahui siapa saja yang mendaftar event yang telah dibuatnya ;

Dependensi:

* Semua actor yang menggunakan Aplikasi Pencarian Event, harus Login saat hendak menggunakan sistem.

# Deskrpsi Rinci Perangkat Lunak

## Deskripsi Kebutuhan

### Kebutuhan Fungsional

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Kebutuhan | Fungsi | Deskripsi |
| 1. | FR-01 | *Login* | Fungsi ini digunakan oleh semua *actor* untuk dapat berinteraksi di dalam sistem. |
| 2. | FR-02 | Input *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menginputkan *event* ke sistem |
| 3. | FR-03 | Edit *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk memperbaharui atau mengubah informasi *event* yang telah dibuat. |
| 4. | FR-04 | Hapus *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menghapus *event* yang telah dibuat. |
| 5. | FR-05 | Lihat *Event* | Fungsi ini digunakan oleh *actor* Mahasiswa dan Admin untuk dapat melihat list-list dan isi dari suatu *event* yang telah dibuat *actor* Pembuat *Event* |
| 6. | FR-06 | Validasi *Event* | Fungsi ini digunakan oleh *actor* Admin untuk memvalidasi kebenaran terkait *event* yang telah dibuat oleh *actor* Pembuat *Event* |
| 7. | FR-07 | Lihat Hasil Validasi *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk melihat hasil validasi dari *actor* Admin terhadap *event* yang telah dibuat. |
| 8. | FR-08 | Buat *Form* *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk membuat *Form* pendaftarn *event* untuk *actor* Mahasiswa |
| 9. | FR-09 | Edit *Form* *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk mengedit atau mengubah informasi *Form*at pada *Form* pendaftaran *event* yang telah dibuat |
| 10. | FR-10 | Hapus *Form* *Event* | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menghapus atau menghilangkan *Form* dari *event* yang telah dibuat. |
| 11. | FR-11 | Mencari *Event* | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mencari daftar-daftar *event* yang telah di post atau dibuat oleh *actor* Pembuat *Event* |
| 12. | FR-12 | *Join* *Event* | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mendaftar dan mengikuti *event* yang diinginkan. |
| 13. | FR-13 | Input *Form* *Event* | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mengisi *Form* pendaftaran *Event* jika ingin mengikuti *event* tersebut. |
| 14. | FR-14 | Update Profile | Fungsi ini digunakan oleh semua *actor* untuk memperbaharui informasi data diri. |
| 15. | FR-15 | Update Password | Fungsi ini digunakan oleh semua *actor* untuk memperbaharui kata sandi yang lama. |
| 16. | FR-16 | Registrasi Akun | Fungsi ini digunakan oleh *actor* Pembuat *Event* dan Mahasiswa untuk melakukan registrasi akun agar bisa mengakses sistem |

### Kebutuhan Non-Fungsional

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Kode Kebutuhan | Deskripsi |
| 1. | Security  Safety | NFR-01 | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menginputkan *Event* ke dalam sistem. |
| 2. | Identification Security | NFR-02 | Fungsi ini digunakan oleh *actor* Admin untuk mengidentifikasi *event* yang telah dibuat oleh *actor* Pembuat *Event* benar adanya secara factual. |
| 3. | Usability | NFR-03 | Fungsi ini memiliki tampilan antar muka sistem yang mudah dipahami untuk semua *actor*. |
| 4. | Usability | NFR-04 | Fungsi ini membuat perangkat lunak berbasis web menjadi responsive sehingga bisa mendukung di semua media browser seperti Chrome, Microsoft Edge, Mozilla Firefox, UC Browser, Opera dan lain-lain. |

## Pemodelan Analisis

### Usecase Diagram

A close up of a map

Description automatically generated

### Usecase Scenario

#### Usecase Scenario #1

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Registrasi Akun | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* dan Mahasiswa untuk melakukan registrasi akun agar bisa mengakses sistem. | |
| Pre-Kondisi | Pembuat *Event* dan Mahasiswa belum terdaftar memiliki akun APE. | |
| Post-Kondisi | Pembuat *Event* dan Mahasiswa telah melakukan pendaftaran dan memiliki akun. | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Membuka Halaman Registrasi |  |
|  | 1. Menampilkan Tampilan dari halaman registrasi |
| 1. Melengkapi data diri, lalu klik tombol registrasi |  |
|  | 1. Melakukan Validasi data |
|  | 1. Menampilkan pesan notifikasi “Registrasi Berhasil” |
|  | 1. Mengarahkan *actor* ke halaman utama |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Membuka Halaman Registrasi |  |
|  | 1. Menampilkan Tampilan dari halaman registasi |
|  | 1. Melengkapi data diri, lalu klik tombol registrasi |  |
|  |  | 1. Menampilkan pesan notifikasi “Registrasi Gagal” karena data email yang dipakai sudah terdaftar |

#### Usecase Scenario #2

|  |  |  |
| --- | --- | --- |
| Nama Use Case | *Login* | |
| Deskripsi | Sistem mengindentifikasi dan mengautentifikasi *actor* siapa saja yang masuk untuk memeberikan otoritas di dalam sistem (sesuai porsinya) | |
| Pre-Kondisi | Pembuat *Event*, Mahasiswa, Admin ingin menggunakan website dan belum melakukan *Login* | |
| Post-Kondisi | Pembuat *Event*, Mahasiswa, Admin telah melakukan *Login* dan telah diauntentifikasi oleh sistem. Semua *actor* dapat melakukan aktivitas di dalam website | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman website dan klik tombol *Login* |  |
|  | 1. Menampilkan tampilan dari halaman *Login* |
| 1. Aktor memasukkan username dan password, kemudian klik tombol *Login* |  |
|  | 1. Sistem melakukan validasi data username dan password dengan data yang tersimpan di dalam database |
| 1. Aktor dapat beraktifitas di dalam website |  |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman website dan klik tombol *Login* |  |
|  | 1. Menampilkan tampilan dari halaman *Login* |
|  | 1. Aktor memasukkan username dan password, kemudian klik tombol *Login* |  |
|  |  | 1. Sistem melakukan validasi data username dan password dengan data yang tersimpan di dalam database |
|  |  | 1. Menampilkan pesan notifikasi “*Login* Gagal” karena data username / password tidak sesuai atau tidak ada di dalam database |

#### Usecase Scenario #3

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menginputkan atau mendaftarkan *event* yang dimiliki ke sistem | |
| Pre-Kondisi | Pembuat *event* telah melakukan *Login* dan memasuki halaman utama website, *event* belum di input atau didaftarkan ke dalam sistem | |
| Post-Kondisi | *Event* telah diinputkan atau didaftarkan ke dalam sistem dan di simpan ke dalam database | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman input *event* |  |
|  | 1. Sistem menampilkan tampilan halaman input *event* |
| 1. Aktor menginputkan data *event* |  |
|  | 1. Sistem menerima dan mengecek data yang di inputkan oleh *actor* Pembuat *Event* apakah sudah benar atau tidak |
|  |  | 1. Memasukkan data *event* yang telah diinputkan ke dalam database kemudian diteruskan ke *actor* Admin untuk di validasi |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman input *event* |  |
|  | 1. Sistem menampilkan tampilan halaman input *event* |
|  | 1. Aktor menginputkan data *event* |  |
|  |  | 1. Sistem menerima dan mengecek data yang di inputkan oleh *actor* Pembuat *Event* apakah sudah benar atau tidak |
|  |  | 1. Menampilkan notifikasi “Gagal menginput *Event*” karena *Form*at inputan masih belum benar |

#### Usecase Scenario #4

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Edit *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk memperbaharui atau mengubah informasi *event* yang telah dibuat. | |
| Pre-Kondisi | Pembuat *event* telah melakukan *Login* dan memasuki halaman utama website, *event* telah di inputkan atau telah didaftarkan | |
| Post-Kondisi | Informasi *Event* yang telah di inputkan telah diperbaharui kemudian di simpan ke dalam database | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
| 1. Aktor memilih *event* mana yang ingin diperbaharui informasi nya, kemudian klik tombol edit *event* |  |
| 1. Aktor menginputkan informasi *event* yang terbaru |  |
|  |  | 1. Sistem menerima dan mengecek data terbaru yang di inputkan oleh *actor* Pembuat *Event* apakah sudah benar atau tidak |
|  |  | 1. Memasukkan data *event* yang terbaru yang telah diinputkan ke dalam database |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
|  | 1. Aktor memilih *event* mana yang ingin diperbaharui informasi nya, kemudian klik tombol edit *event* |  |
|  | 1. Aktor menginputkan informasi *event* yang terbaru |  |
|  |  | 1. Sistem menerima dan mengecek data terbaru yang di inputkan oleh *actor* Pembuat *Event* apakah sudah benar atau tidak |
|  |  | 1. Menampilkan notifikasi “Gagal memperharui *Event*” karena *Form*at inputan masih belum benar |

#### Usecase Scenario #5

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Hapus *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menghapus *event* yang telah dibuat. | |
| Pre-Kondisi | Pembuat *Event* telah melakukan *Login* dan *event* yang ingin dihapus belum terhapus dari sistem dan database | |
| Post-Kondisi | *Event* telah dipilih dan telah terhapus dari sistem dan database | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan list *event* |
| 1. Aktor memilih *event* yang ada di dalam list, kemudian klik tombol “Hapus” |  |
|  | 1. Sistem menampilkan pesan konfirmasi “Apakah anda yakin *event* ini ingin dihapus?” |
|  | 1. Aktor mengklik “Ya” pada pesan konfirmasi |  |
|  |  | 1. Sistem menghapus *event* yang telah dipilih |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan list *event* |
|  | 1. Aktor memilih *event* yang ada di dalam list, kemudian klik tombol “Hapus” |  |
|  |  | 1. Sistem menampilkan pesan konfirmasi “Apakah anda yakin *event* ini ingin dihapus?” |
|  | 1. Aktor mengklik “Tidak” pada pesan konfirmasi |  |

#### Usecase Scenario #6

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Mahasiswa dan Admin untuk dapat melihat list-list dan isi dari suatu *event* yang telah dibuat *actor* Pembuat *Event* | |
| Pre-Kondisi | Aktor belum melakukan *Login* dan belum melihat list *event* | |
| Post-Kondisi | Aktor telah melakukan *Login* dan telah melihat daftar dari list *event* yang telah dibuat oleh aktor Pembuat *Event* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor (Admin) melakukan *Login* dan memasuki ke halaman utama |  |
| 1. Aktor membuka halaman request *event* |  |
|  | 1. Sistem menampilkan tampilan halaman request *event* |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor (Mahasiswa) melakukan *Login* dan memasuki ke halaman utama |  |
|  | 1. Sistem menampilkan daftar list *event* yang telah dibuat oleh aktor Pembuat *Event* di halaman utama |

#### Usecase Scenario #7

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Validasi *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Admin untuk memvalidasi kebenaran terkait *event* yang telah dibuat oleh *actor* Pembuat *Event* | |
| Pre-Kondisi | Admin telah melakukan *Login*, *event* yang dibuat oleh Pembuat *Event* belum dibaca dan belum di validasi | |
| Post-Kondisi | *Event* yang dibuat oleh Pembuat *Event* telah dibaca dan divalidasi | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman request *event* |  |
|  | 1. Sistem menampilkan tampilan halaman request *event* |
| 1. Melihat list-list *event* yang di request oleh *actor* Pembuat *Event* |  |
| 1. Memilih *event* yang direquest untuk di validasi, kemudian klik tombol “Lihat” untuk melihat isi dari *Event* |  |
|  |  | 1. Sistem menampilkan tampilan isi dari *Event* |
|  | 1. Membaca isi *event* tersebut, kemudian mengklik tombol “Terima” |  |
|  |  | 1. Sistem menampilkan pesan konfirmasi “Apakah anda yakin *event* ini sudah benar ?” |
|  | 1. Aktor mengklik tombol “Ya” pada pesan konfirmasi |  |
|  |  | 1. Sistem menyimpan *event* tersebut ke dalam database dan mengirim notifikasi kepada *actor* Pembuat *Event* bahwa *event* telah berhasil divalidasi |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman request *event* |  |
|  | 1. Sistem menampilkan tampilan halaman request *event* |
|  | 1. Melihat list-list *event* yang di request oleh *actor* Pembuat *Event* |  |
|  | 1. Memilih *event* yang direquest untuk di validasi, kemudian klik tombol “Lihat” untuk melihat isi dari *Event* |  |
|  |  | 1. Sistem menampilkan tampilan isi dari *Event* |
|  | 1. Membaca isi *event* tersebut, kemudian mengklik tombol “Tolak” |  |

#### Usecase Scenario #8

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Lihat hasil validasi *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk melihat hasil validasi dari *actor* Admin terhadap *event* yang telah dibuat. | |
| Pre-Kondisi | Aktor Pembuat *Event* telah melakukan *Login* dan belum melihat pesan notifikasi validasi *event* | |
| Post-Kondisi | Aktor telah melihat dan membaca pesan notifikasi *event* yang telah divalidasi oleh Admin | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman pesan notifikasi |  |
|  | 1. Sistem menampilkan tampilan halaman pesan notifikasi |
| 1. Aktor melihat list pesan notifikasi yang masuk, kemudian klik pesan notifikasi yang terbaru |  |
|  | 1. Sistem menampilkan tampilan isi dari pesan notifikasi yang terbaru |
|  | 1. Aktor telah membaca isi pesan notifikasi tersebut |  |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman pesan notifikasi |  |
|  | 1. Sistem menampilkan tampilan halaman pesan notifikasi |
|  | 1. Aktor melihat list pesan notifikasi yang masuk |  |
|  | 1. Aktor tidak melihat ada pesan notifikasi terbaru |  |

#### Usecase Scenario #9

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Buat *Form* *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk membuat *Form* pendaftarn *event* untuk *actor* Mahasiswa | |
| Pre-Kondisi | Aktor Pembuat *Event* telah melakukan *Login*, *event* belum dipilih dan belum ada *Form* pendaftaran di dalamnya | |
| Post-Kondisi | *Event* telah dipilih dan terdapat *Form* pendaftaran di dalamnya | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
| 1. Aktor melihat list *event* dan memilih *event* yang akan di tambahkan *Form* pendaftaran dengan cara mengklik tombol “Tambah *Form*” |  |
|  | 1. Menampilkan tampilan halaman pembuatan *Form* *event* |
|  | 1. Membuat *Form*at *Form* *event* yang nantinya akan diinputkan oleh *actor* Mahasiswa |  |
|  | 1. Mengklik tombol “Simpan” |  |
|  |  | 1. Menyimpan perububahan ke dalam database |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
|  | 1. Aktor melihat list *event* dan memilih *event* yang akan di tambahkan *Form* pendaftaran dengan cara mengklik tombol “Tambah *Form*” |  |
|  |  | 1. Menampilkan tampilan halaman pembuatan *Form* *event* |
|  | 1. Membuat *Form*at *Form* *event* yang nantinya akan diinputkan oleh *actor* Mahasiswa |  |
|  | 1. Mengklik tombol “Batal” |  |

#### Usecase Scenario #10

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Edit *Form* *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk mengedit atau mengubah informasi *Form*at pada *Form* pendaftaran *event* yang telah dibuat | |
| Pre-Kondisi | Aktor Pembuat *Event* telah melakukan *Login* dan *Form*at *Form* yang ingin dirubah belum terubah | |
| Post-Kondisi | *Event* yang ingin diperbaharui *Form*nya telah berubah dan disimpan ke dalam database | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
| 1. Aktor melihat data list *event* yang telah dibuat kemudian memilih *event* yang ingin diedit *Form* nya, dengan cara mengklik tombol “Edit *Event*” |  |
|  | 1. Sistem menampilkan tampilan halaman Edit *Event* |
|  | 1. Aktor mengklik tombol “Edit *Form*” |  |
|  | 1. Aktor mengedit *Form* sesusai yang diinginkan, kemudian klik tombol “Simpan” untuk menyimpan perubahan |  |
|  |  | 1. Sistem menyimpan perubahan *Form* tersebut kedalam database *Event* |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
|  | 1. Aktor melihat data list *event* yang telah dibuat kemudian memilih *event* yang ingin diedit *Form* nya, dengan cara mengklik tombol “Edit *Event*” |  |
|  |  | 1. Sistem menampilkan tampilan halaman Edit *Event* |
|  | 1. Aktor mengklik tombol “Edit *Form*” |  |
|  | 1. Aktor tidak jadi mengedit *Form* dan mengklik tombol “Batal” |  |

#### Usecase Scenario #11

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Hapus *Form* *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Pembuat *Event* untuk menghapus atau menghilangkan *Form* dari *event* yang telah dibuat. | |
| Pre-Kondisi | Aktor telah melakukan *Login* dan *Form* *event* yang dipilih belum terhapus dari database *Event* | |
| Post-Kondisi | *Form* *Event* yang telah di pilih telah dihapus dari sistem dan database *Event* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
| 1. Aktor melihat data list *event* yang telah dibuat kemudian memilih *event* yang ingin diedit *Form* nya, dengan cara mengklik tombol “Edit *Event*” |  |
|  | 1. Sistem menampilkan tampilan halaman Edit *Event* |
|  | 1. Aktor mengklik tombol “Hapus *Form*” |  |
|  |  | 1. Sistem menampilkan pesan konfirmasi “Apakah anda yakin *event* ini ingin dihapus?” |
|  | 1. Aktor mengklik tombol “Ya” pada pesan konfirmasi |  |
|  |  | 1. Sistem menghapus *Form* *event* yang dipilih dari database. |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor membuka halaman list *event* |  |
|  | 1. Sistem menampilkan tampilan halaman list *event* |
|  | 1. Aktor melihat data list *event* yang telah dibuat kemudian memilih *event* yang ingin diedit *Form* nya, dengan cara mengklik tombol “Edit *Event*” |  |
|  |  | 1. Sistem menampilkan tampilan halaman Edit *Event* |
|  | 1. Aktor mengklik tombol “Hapus *Form*” |  |
|  |  | 1. Sistem menampilkan pesan konfirmasi “Apakah anda yakin *event* ini ingin dihapus?” |
|  | 1. Aktor mengklik tombol “Tidak” pada pesan konfirmasi |  |

#### Usecase Scenario #12

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Mencari *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mencari daftar-daftar *event* yang telah di post atau dibuat oleh *actor* Pembuat *Event* | |
| Pre-Kondisi | Aktor telah melakukan *Login* dan belum mencari *event* yang ingin dicari | |
| Post-Kondisi | Aktor telah mencari *event* yang diinginkan dan sistem menampilkan informasi *event* yang dicari | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
|  |  |
| 1. Aktor mengklik tombol pencarian di halaman utama |  |
| 1. Aktor menginputkan nama *Event* yang ingin dicari |  |
|  | 1. Sistem menampilkan informasi *Event* yang dicari sesuai dengan inputan |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor mengklik tombol pencarian di halaman utama |  |
| 1. Aktor menginputkan nama *Event* yang ingin dicari |  |
|  |  | 1. Sistem menampilkan pesan bahwa *event* yang dicari tidak ditemukan |

#### Usecase Scenario #13

|  |  |  |
| --- | --- | --- |
| Nama Use Case | *Join* *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mendaftar dan mengikuti *event* yang diinginkan. | |
| Pre-Kondisi | Aktor sudah melakukan *Login* dan belum memilih *Event* yang ingin diikutinya | |
| Post-Kondisi | Aktor telah memilih *Event* dan telah mengklik tombol *Join* *Event* dan menyimpan aksi tersebut kedalam database | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor berada di halaman utama yang terdapat daftar list-list *event* yang ada |  |
| 1. Aktor memilih *event* dan mengklik tombol “*Join* *Event*” |  |
|  | 1. Sistem mengalihkan ke tampilan halaman dari isi *Event* tersebut |
| 1. Aktor mengisi *Form* dan mengklik tombol “*Join*” |  |

#### Usecase Scenario #2

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input *Form* *Event* | |
| Deskripsi | Fungsi ini digunakan oleh *actor* Mahasiswa untuk mengisi *Form* pendaftaran *Event* jika ingin mengikuti *event* tersebut. | |
| Pre-Kondisi | Aktor telah melakukan *Login*, belum melakukan *Join* *event* dan mengisi *Form* *Event* | |
| Post-Kondisi | Aktor telah *Join* *Event* dan telah mengisi *Form* *Event* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor berada di halaman utama yang terdapat daftar list-list *event* yang ada |  |
| 1. Aktor memilih *event* dan mengklik tombol “*Join* *Event*” |  |
|  | 1. Sistem mengalihkan ke tampilan halaman dari isi *Event* tersebut |
| 1. Aktor mengisi *Form* dan mengklik tombol “*Join*” |  |
|  |  | 1. Sistem memberikan pesan konfirmasi “Apakah data diri yang diinputkan pada *Form* sudah benar ?” |
|  | 1. Aktor mengklik tombol “Ya” pada pesan konfirmasi |  |
|  |  | 1. Sistem menyimpan inputan *Form* tersebut kedalam database |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor berada di halaman utama yang terdapat daftar list-list *event* yang ada |  |
| 1. Aktor memilih *event* dan mengklik tombol “*Join* *Event*” |  |
|  |  | 1. Sistem mengalihkan ke tampilan halaman dari isi *Event* tersebut |
|  | 1. Aktor mengisi *Form* dan mengklik tombol “*Join*” |  |
|  |  | 1. Sistem memberikan pesan konfirmasi “Apakah data diri yang diinputkan pada *Form* sudah benar ?” |
|  | 1. Aktor mengklik tombol “Tidak” pada pesan konfirmasi |  |

#### Usecase Scenario #15

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Update Profile | |
| Deskripsi | Fungsi ini digunakan oleh semua *actor* untuk memperbaharui informasi data diri. | |
| Pre-Kondisi | Aktor telah melakukan *Login* dan data profile belum berubah | |
| Post-Kondisi | Aktor telah memperbaharui data profile | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor telah melakukan *Login* dan memasuki halaman utama. |  |
| 1. Aktor mengklik menu Profile |  |
|  | 1. Sistem menampilkan tampilan halaman Profile Akun |
| 1. Aktor mengklik tombol “Edit Profile” |  |
|  |  | 1. Sistem menampilkan halaman “Edit Profile” |
|  | 1. Aktor menginputkan informasi data diri yang terbaru |  |
|  | 1. Aktor mengklik tombol “Simpan” |  |
|  |  | 1. Sistem menyimpan perubahan data diri yang terbaru kedalam database Akun |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor telah melakukan *Login* dan memasuki halaman utama. |  |
| 1. Aktor mengklik menu Profile |  |
|  |  | 1. Sistem menampilkan tampilan halaman Profile Akun |
|  | 1. Aktor mengklik tombol “Edit Profile” |  |
|  |  | 1. Sistem menampilkan halaman “Edit Profile” |
|  | 1. Aktor menginputkan informasi data diri yang terbaru |  |
|  | 1. Aktor mengklik tombol “Simpan” |  |
|  |  | 1. Sistem menampilkan pesan bahwa “Terdapat beberapa informasi yang diinputkan tidak valid/tidak sesuai *Form*at” |

#### Usecase Scenario #16

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Update Password | |
| Deskripsi | Fungsi ini digunakan oleh semua *actor* untuk memperbaharui kata sandi yang lama. | |
| Pre-Kondisi | Aktor telah melakukan *Login* dan kata sandi belum berubah | |
| Post-Kondisi | Aktor telah memperbaharui kata sandi | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Aktor telah melakukan *Login* dan memasuki halaman utama. |  |
| 1. Aktor mengklik menu Profile |  |
|  | 1. Sistem menampilkan tampilan halaman Profile Akun |
| 1. Aktor mengklik tombol “Edit Password” |  |
|  |  | 1. Sistem menampilkan halaman “Edit Password” |
|  | 1. Aktor menginputkan kata sandi terbaru |  |
|  | 1. Aktor mengklik tombol “Simpan” |  |
|  |  | 1. Sistem menyimpan perubahan kata sandi yang terbaru kedalam database Akun |
| Skenario Eksepsional (*Alternative flow*) |  | |
|  | Aktor | Sistem |
| 1. Aktor telah melakukan *Login* dan memasuki halaman utama. |  |
| 1. Aktor mengklik menu Profile |  |
|  |  | 1. Sistem menampilkan tampilan halaman Profile Akun |
|  | 1. Aktor mengklik tombol “Edit Password” |  |
|  |  | 1. Sistem menampilkan halaman “Edit Password” |
|  | 1. Aktor menginputkan kata sandi terbaru |  |
|  | 1. Aktor mengklik tombol “Simpan” |  |
|  |  | 1. Sistem menampilkan pesan bahwa “Password terlalu mudah atau Jangan menggunakan Password yang pernah digunakan sebelumnya” |

### Class Diagram

A close up of text on a white background

Description automatically generated

# Kebutuhan Antarmuka Eksternal

## Antarmuka Pengguna

Di dalam pembuatan Aplikasi Pencarian Event berbasis web ini, pengguna dapat berinteraksi dengan perangkat lunak melalui antarmuka berbasis web browser, dimana perangkat lunak dapat menampilkan menu beserta fungsinya kepada pengguna. Perangkat lunak mendapat masukan berupa klik untuk mengklik tombol fungsi yang ada di perangkat lunak melalui perangkat keras berupa mouse atau touchpad dan perangkat lunak dapat menerima masukan berupa teks dari pengguna melalui keyboard. Umpan balik atau keluaran dari perangkat lunak dapat dilihat oleh pengguna dengan menggunakan monitor secara langsung. Berikut ini penejelasan yang mencakup dari antarmuka Aplikasi Pencarian Event :

* Formulir Login untuk Pembuat Event dan Mahasiswa

Pembuat Event dan Mahasiswa dapat mengakses perangkat lunak ketika sudah menginputkan username dan password yang valid.

* Formulir Registrasi untuk Pembuat Event dan Mahasiswa

Terdapat Formulir registrasi kepada kepada Pembuat Event dan Mahasiswa untuk membuat akun pada perangkat lunak tersebut, inputan di dalam Formulir tersebut berupa input username, nama, email, password, dan no hp.

* Formulir Input Event

Setelah melakukan Login ke dalam website Aplikasi Pembuatan Event, pengguna Pembuat Event dapat melakukan proses penginputan event melalui halaman input Event dengan mengisi Formulir online yang berisi data-data terkait event yang hendak diinputkan.

* View data Event untuk Pembuat Event

Terdapat tampilan informasi berupa list yang menunjukkan data dari database Event yang telah dibuat. Di sebelah kanan setiap data terdapat ikon delete, edit, tambah Form yang apabila di klik maka akan menuju ke halaman Edit Event

* View data Event untuk divalidasi oleh Admin

Terdapat tampilan informasi berupa list di dalam halaman Admin di bagian menu List Event yang menunjukkan data dari database event yang telah dibuat oleh Pembuat Event. Di sebelah kanan setiap data terdapat ikon Lihat Detail Event, Terima, Tolak.

* View data Event untuk Mahasiswa

Setelah melakukan Login kemudian masuk ke halaman utama, terdapat tampilan informasi berupa list yang menunjukkan data dari database Event yang telah dibuat oleh Pembuat Event. Di paling atas halaman utama terdapat tombol pencarian untuk mencari Event secara spesifik dan di sebelah kanan setiap data terdapat ikon Join Event.

* View Pesan Notifikasi untuk Pembuat Event dan Mahasiswa

Setelah melakukan Login, terdapat ikon notifikasi yang berada pada header website di bagian paling kanan untuk melihat dan membaca pesan notifikasi yang masuk atau yang telah dikirim oleh sistem.

* Formulir Input Form Event untuk Mahasiswa

Ketika berada di halaman utama, maka akan ditampilkan berbagai macam informasi data dari database Event terkait event yang telah dibuat oleh Pembuat Event. Di sebelah kanan terdapat ikon Join Event, apabila di klik akan menuju ke Halaman dari isi Event tersebut, dan dibawahnya terdapat Formulir Form pendaftaran Event inputannya berupa Nama, Fakultas, Jurusan, Angkatan, Email dan No.HP.

## Antarmuka Perangkat Keras

Website Aplikasi Pencarian Event yang dibahas dalam SKPL ini dapat diakses melalui perang keras seperti :

1. Laptop atau Komputer, dengan minimal spesifikasi sebagai berikut:

- Layar resolusi berapapun

- Processor : Intel Pentium 4 2,8GHz atau yang terbaru;

- RAM : 512MB

- Windows XP, MacOS X

1. Gawai, dengan minimal spesifikasi minimal sebagai berikut:

* Layar resolusi berapapun
* ARM Dual Core Processor 1,5GHz
* RAM : 1GB
* Android 4.1.2 Ice Cream Sandwich, iOS 7

## Antarmuka Perangkat Lunak

Website Aplikasi Pencarian Event yang dibahas dalam SKPL ini dingaun untuk mempermudah proses Pencarian Event oleh Mahasiswa yang ingin mencari event-event yang sedang berlangsung di lingkungan kampus Telkom University melalui perantara Pembuat Event. Website yang akan dibangun menggunakan bahasa pemrograman HTML dan CSS. Website Aplikasi Pencarian Event ini dapat dibuka melalui berbagai Web Browser, seperti Mozila Firefox, Google Chrome, Opera, Microsoft Edge, UC Browser, dan web browser lainnya yang setara.

## Antarmuka Komunikasi

Sistem pemanfaatan teknologi Website Aplikasi Pencarian Event untuk masing masing role terhubung secara Client – Server dalam lingkup jaringan internet yang berbasis Transmission Control Protocol / Internet Protocol.

# Requirements Lain

## Database

Dalam proses pembuatan Website Aplikasi Pencarian Event ini, database digunakan untuk menyimpan data Event, data Mahasiswa, data Pembuat Event itu sendiri. Database memiliki peran yang penting dalam sistem ini, karena Mahasiswa perlu melihat data Event untuk melihat dan mengikuti Event sesuai dengan yang diinginkan.

## MySQL

Diperlukan pengaturan database agar database dapat berjalan sesuai dengan kebutuhan yang diinginkan, maka digunakan RDBMS MySQL dalam pengaturan database ini, karena MySQL merupakan perangkat lunak yang bersifat gratis atau bersifat sumber terbuka (open-source) sehingga tidak perlu mengeluarkan biaya tambahan untuk pengelolaan database. MySQL juka memiliki dukungan kompabilitas yang cukup baik sehingga bisa digunakan di berbagai sistem operasi seperti Windows, Linux, MacOS, Android dan iOS.

## Apache

Dalam pembuatan Website Aplikasi Pencarian Event ini, digunakan perangkat lunak web server Apache, untuk mengimplementasikan standar protokol HTTP agar memudahkan pengembang untuk membuat aplikasi sesuai dengan kebutuhan.

# Lampiran A: Daftar Kata-Kata Sukar

|  |  |
| --- | --- |
| **ISTILAH** | **DEFINISI** |
| Aplikasi | Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna |
| Open Source | Open Source memiliki arti yaitu Sumber Terbuka merupakan sistem pengembangan yang tidak dikoordinasi oleh suatu individu atau sebuah lembaga pusat melainkan oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (source code) yang tersebar dan tersedia bebas yang biasa diakses menggunakan fasilitas komunikasi internet. |
| Pengembang | Pengembang atau dalam bahasa Inggris disebut Developer merupakan seorang programmer yang sudah terlatih tidak hanya menyelesaikan masalah, namun juga membangun suatu produk. |
| Client | Seseorang yang menggunakan layanan dari seorang atau sebuah organisasi profesional. |
| Realtime | Realtime atau disebut juga Komputasi waktu nyata merupakan kondisi pengoperasian dari suatu sistem perangkat keras dan perangkat lunak yang dibatasi oleh rentang waktu dan memiliki tenggat waktu yang jelas, relatif terhadap waktu sautu operasi atau peristiwa yang terjadi. |

# Lampiran B: Analysis Models

