

## ***MAINTENANCE PORTAL WEB MARKETPLACE EVENT PERGURUAN TINGGI MENGUNAKAN METODE ITERATIVE DAN INCREMENTAL***

### ***MAINTENANCE A CROWDSOURCING COLLEGE EVENT WEB PORTAL ON PUBLICATION MODULE USING ITERATIVE AND INCREMENTAL METHOD***

Muhammad Edwin Baihaqi<sup>1</sup>, Dr. Irfan Darmawan<sup>2</sup>, Faisal Mufied Al Anshary<sup>3</sup>

<sup>1,2,3</sup>Information System, Department of Industrial and System Engineering Telkom University

<sup>1</sup>m.edwinbaihaqi@gmail.com, <sup>2</sup>irfandarmawan@telkomuniversity.ac.id,

<sup>3</sup>faisalmufied@telkomuniversity.ac.id

---

#### **Abstrak**

*Maintenance* merupakan salah satu hal yang terpenting pada pengembangan suatu sistem perangkat lunak. *Maintenance* dilakukan untuk meningkatkan kualitas dari sistem seiring dengan perkembangan waktu dan perubahan *requirement* dari pengguna. Saat ini terdapat suatu pengembangan portal *web event* perguruan tinggi bernama *acarakampus* yang memanfaatkan konsep *marketplace*. Seperti sistem perangkat lunak lainnya, portal *web acarakampus* membutuhkan *maintenance* untuk meningkatkan kualitasnya.

Beberapa hal ditemukan sebagai bahan untuk dilakukannya *maintenance*. Pada sistem yang telah ada saat ini belum menyediakan fitur yang dapat mengolah pola perilaku pengunjung situs atau keinginan pengunjung akan suatu jenis *event* sehingga tidak adanya fitur rekomendasi *event* yang diberikan kepada pengguna portal *web acarakampus* berdasarkan suatu indikator tertentu. Selain itu dibutuhkan perubahan dan peningkatan dari fitur-fitur yang telah ada.

Untuk mengatasi permasalahan tersebut dibutuhkan rancangan yang dapat menjadi acuan untuk melakukan *maintenance* portal *web acarakampus*. Hasil dari penelitian ini adalah rancangan *maintenance* portal *web marketplace event* perguruan tinggi dengan menggunakan metode *iterative & incremental*.

**Kata kunci :** *maintenance, event, rancangan aplikasi website, iterative & incremental, rekomendasi, rekomendasi berbasis tag.*

---

#### **Abstract**

*Maintenance is one of the most important things in developing a software system. Maintenance is done to improve the quality of the system along with the development of time and changes in requirements of the user. Currently there is a development of a college event web portal called Acarakampus which utilizes the concept of marketplace. Like other software systems, the Acarakampus web portal requires maintenance to improve its quality.*

*Some things were found as materials for maintenance. The existing system does not provide features that can process site visitor behavior patterns or visitor desires for a type of event so that there is no event recommendation feature provided to users of the Acarakampus web portal based on a particular indicator. In addition, changes and improvements to existing features are needed.*

*To overcome these problems, a design is needed that can be used as a reference for maintenance of the Acarakampus web portal. The result of this study is a maintenance design of the college's event marketplace web portal using iterative & incremental methods.*

**Keyword :** *maintenance, event, website application design, iterative & incremental, tag-based recommendations, recommendations.*

---

## 1. Pendahuluan

*Maintenance* merupakan salah satu hal yang terpenting pada pengembangan suatu sistem perangkat lunak. *Maintenance* dilakukan untuk meningkatkan kualitas dari sistem seiring dengan perkembangan waktu dan perubahan *requirement* dari pengguna. *Maintenance* perangkat lunak adalah suatu proses modifikasi pada produk perangkat lunak setelah rilis untuk memperbaiki kesalahan, meningkatkan performansi atau atribut lain, atau untuk beradaptasi dengan lingkungan yang berubah [1].

Saat ini terdapat suatu pengembangan portal web *event* perguruan tinggi bernama Acarakampus [2] [3] yang memanfaatkan konsep *marketplace*. Karena berbentuk portal web maka dapat diklasifikasikan sebagai *electronic marketplace*. *Electronic marketplace* adalah suatu sistem informasi interorganisasional yang memungkinkan berbagai penjual, pembeli dan *stakeholder* lainnya untuk berkomunikasi dan bertransaksi melalui ruang pasar sentral yang dinamis dan didukung oleh berbagai layanan tambahan [2]. Portal web acarakampus ini menjadi wadah untuk mempublikasikan *event-event* yang diselenggarakan oleh suatu organisasi atau lembaga yang berhubungan dengan perguruan tinggi atau menargetkan kepada calon peserta dari masyarakat perguruan tinggi. Portal web ini pun menyediakan layanan transaksi jual beli tiket *event* kepada pengunjung situs.

Seperti sistem perangkat lunak lainnya, portal web acarakampus membutuhkan *maintenance* untuk meningkatkan kualitasnya. Beberapa hal ditemukan sebagai bahan untuk dilakukannya *maintenance*. Pada sistem yang telah ada saat ini belum menyediakan fitur yang dapat mengolah pola perilaku pengunjung situs atau keinginan pengunjung akan suatu tipe *event* sehingga tidak adanya fitur rekomendasi *event* yang diberikan kepada pengguna portal web acarakampus berdasarkan suatu indikator tertentu. Selanjutnya adalah suatu *event* hanya bisa terdaftar dalam satu kategori sementara dalam satu *event* memungkinkan memiliki lebih dari satu kategori *event* dalam penyelenggaraannya. Kondisi selanjutnya adalah suatu *event* hanya bisa terafiliasi dengan satu perguruan tinggi sementara dalam suatu *event* memungkinkan terdiri dari terafiliasi dari berbagai perguruan tinggi yang bekerja sama untuk menyelenggarakan suatu *event*. Pada sistem yang telah ada saat ini juga tidak memiliki fitur untuk mengelola organisasi sehingga bisa saja pengguna tersebut mengatasnamakan organisasi yang telah terdaftar dalam sistem dimana pengguna tersebut tidak memiliki wewenang akan organisasi tersebut. Selanjutnya sistem saat ini juga belum memiliki diversifikasi media untuk memperluas publikasi *event*.

Berdasarkan pemaparan diatas, peneliti akan merancang *maintenance* portal web *marketplace event* perguruan tinggi. *Maintenance* yang dilakukan berupa penambahan fitur personalisasi berdasarkan rekomendasi berbasis *tag*. Fitur personalisasi bertujuan agar pengunjung situs yang menjadi *member* mendapatkan informasi *event-event* yang sesuai dengan *member*. Selanjutnya penambahan fitur mengelola organisasi. Dengan adanya fitur ini maka suatu organisasi yang terdaftar dalam sistem akan memiliki minimal satu orang penanggung jawab dan meminimalisir penggunaan nama organisasi yang bukan menjadi wewangnya. Penambahan fitur selanjutnya adalah penambahan *Line Chatbot* sebagai diversifikasi media untuk publikasi event. Dengan menggunakan *Chatbot* maka data event yang diakses melalui aplikasi media sosial *Line* akan sama dengan data yang terdapat pada sistem dan data yang ditampilkan merupakan data terbaru. Selain penambahan ada pula perbaikan yang meliputi satu event yang dapat memilih lebih dari satu kategori dan juga dapat berafiliasi dengan lebih dari 1 perguruan tinggi setelah *maintenance* dilakukan. *Maintenance* dilakukan dengan menggunakan metode *iterative* dan *incremental* karena perubahan-perubahan pada tahap *maintenance* ini bersifat dinamis sehingga cocok menggunakan metode ini yang dapat menangani perubahan *requirement* saat melakukan pengembangan.

## 2. Dasar Teori

### 2.1 Software Maintenance

*Maintenance* perangkat lunak berdasarkan IEEE Standard 1219-1992 adalah suatu proses modifikasi pada produk perangkat lunak setelah rilis untuk memperbaiki kesalahan, meningkatkan performansi atau atribut lain, atau untuk beradaptasi dengan lingkungan yang berubah [1].

Kegiatan *maintenance* perangkat lunak dapat dibagi menjadi 4 jenis yaitu [3]:

1. *Corrective maintenance*

*Corrective maintenance* adalah modifikasi perangkat lunak yang dilakukan setelah perangkat lunak tersebut telah rilis untuk memperbaiki masalah-masalah yang ditemukan.

## 2. *Preventive maintenance*

*Preventive maintenance* adalah modifikasi perangkat setelah rilis untuk mendeteksi dan memperbaiki kesalahan laten dalam perangkat lunak sebelum mereka menjadi kesalahan operasional.

## 3. *Adaptive maintenance*

*Adaptive maintenance* adalah modifikasi perangkat lunak yang dilakukan setelah rilis untuk menjaga produk perangkat lunak dapat digunakan dalam lingkungan yang berubah atau yang mengubah lingkungan.

## 4. *Perfective maintenance*

*Perfective maintenance* adalah modifikasi perangkat lunak setelah rilis untuk mendeteksi dan memperbaiki kesalahan laten dalam produk perangkat lunak sebelum dimanifestasikan sebagai kegagalan.

## 2.2 Electronic Marketplace

*Marketplace* adalah suatu tempat yang mempertemukan penjual dan pembeli dalam suatu waktu untuk melakukan transaksi jual beli. Sementara itu definisi *electronic marketplace* adalah suatu sistem informasi interorganisasional yang memungkinkan berbagai penjual, pembeli dan *stakeholder* lainnya untuk berkomunikasi dan bertransaksi melalui ruang pasar sentral yang dinamis dan didukung oleh berbagai layanan tambahan [2].

## 2.3 Tag-Based Recommendation

*Tag-based recommendation* adalah algoritme rekomendasi yang memprediksi preferensi pengguna untuk item berdasarkan preferensi tersirat mereka akan suatu *tag*. Sistem pemberian tag menawarkan kepada pengguna cara alternatif untuk mengatasi tugas rekomendasi dan tugas memprediksi.

Algoritme *tag-based recommendation* dapat di klasifikasikan menjadi 2 yaitu [4]:

### 1. *Implicit Tag-Based Algorithms*

*Implicit Tag-Based Algorithms* adalah algoritme rekomendasi yang menggunakan data implisit untuk mendukung situs yang tanpa menggunakan sistem *item ratings*. Sebagai *input*, algoritme ini menggunakan preferensi *tag* yang disimpulkan oleh semua data implisit. Sebagai *output*, algoritme ini menghasilkan skor yang cocok untuk peringkat *item* dalam daftar rekomendasi. Hasil dari algoritme ini cocok untuk tugas rekomendasi tapi tidak cocok untuk tugas memprediksi.

*Implicit Tag-Based Algorithms* sendiri dapat dibagi menjadi 2 jenis:

#### a. *Implicit-tag Algorithms*

Algoritma ini mengkalkulasikan hubungan antara preferensi pengguna dengan *tag* dan suatu objek secara implisit hanya berdasarkan informasi yang ada.

#### b. *Implicit-tag-pop Algorithms*

Algoritma ini mengkalkulasikan hubungan antara preferensi pengguna dengan *tag* dan suatu objek secara implisit berdasarkan informasi yang ada dan ditambah faktor popularitas suatu objek seperti jumlah yang dilihat dan jumlah tag pada objek tersebut.

### 2. *Explicit Tag-Based Algorithms*

*Explicit Tag-Based Algorithms* adalah algoritme rekomendasi yang menggunakan data implisit dan eksplisit untuk mendukung situs yang menggunakan sistem *item ratings*. Hasil dari algoritme ini cocok untuk tugas rekomendasi dan tugas memprediksi.

*Explicit Tag-Based Algorithms* sendiri dapat dibagi menjadi 3 jenis:

#### a. *Cosine-tag Algorithms*

Algoritme ini memprediksi berdasarkan *rating* yang diberikan pengguna terhadap suatu objek dan diproses dengan preferensi pengguna terhadap suatu objek *tag*.

b. *Linear-tag Algorithms*

Algoritme ini memprediksi berdasarkan *rating* suatu objek dan diproses dengan preferensi pengguna terhadap suatu objek *tag*.

c. *Regress-tag Algorithms*

Algoritme ini hampir sama dengan *Linear-tag* hanya saja algoritme ini ikut mempertimbangkan hubungan antar *tag* karena dimungkinkan terjadi nama *tag* yang sama tetapi memiliki maksud yang sama

## 2.4 Event

*Event* adalah suatu kegiatan yang diselenggarakan untuk memperingati hal-hal penting sepanjang hidup manusia baik secara individu atau kelompok yang terikat secara adat, budaya, tradisi dan agama yang diselenggarakan untuk tujuan tertentu serta melibatkan lingkungan masyarakat yang diselenggarakan pada waktu tertentu. [1].

## 2.5 Iterative & Incremental

*Iterative development* adalah sebuah strategi untuk mengembangkan sistem yang memungkinkan untuk pengerjaan ulang pada bagian dari sistem untuk menghilangkan kesalahan atau melakukan perbaikan berdasarkan masukan dari pengguna [3]. *Incremental development* adalah pengembangan sistem dalam serangkaian peningkatan sepanjang skala waktu proyek [3]. Dengan menggabungkan dua konsep tersebut maka diharapkan proses pembangunan aplikasi dapat beradaptasi dengan cepat sesuai kebutuhan sehingga pada akhirnya dapat meningkatkan kualitas aplikasi.

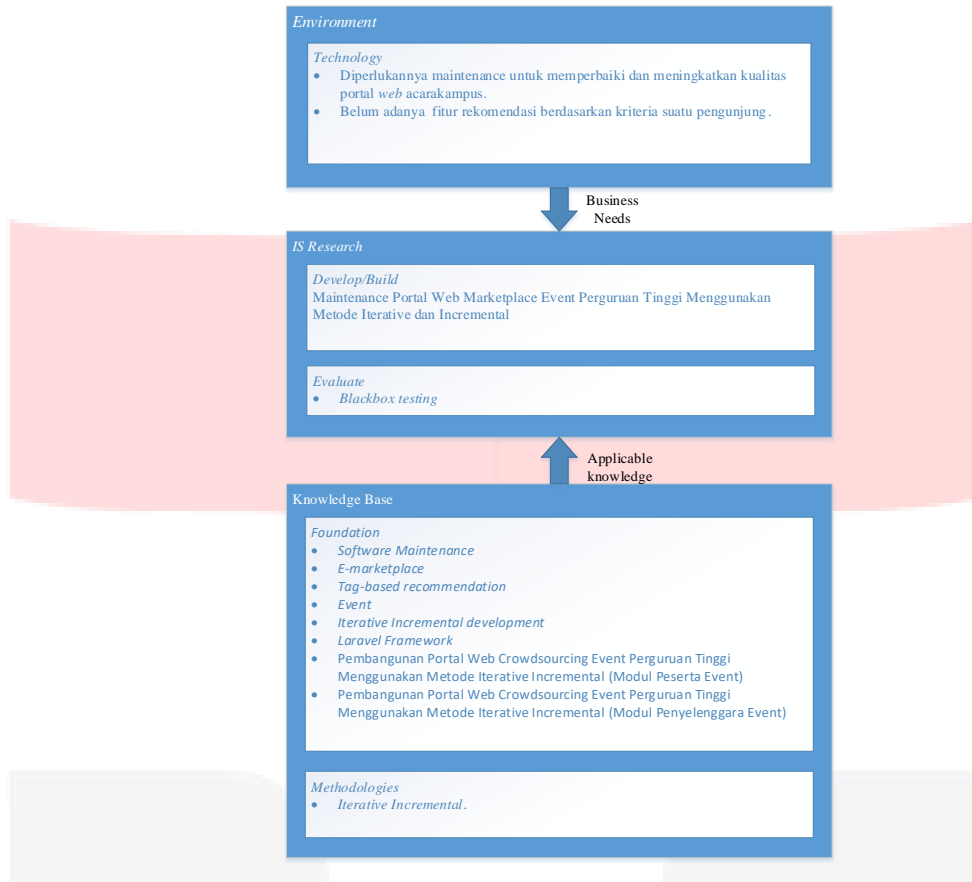
## 2.6 Laravel

Laravel salah satu framework aplikasi berbasis web dengan dasar PHP . Di desain untuk membangun aplikasi web dengan mengikuti pedoman MVC (Model , View , Controller ). Laravel pertama kali dikembangkan oleh Taylor Otwell. Laravel merupakan framework PHP yang bersifat opensource [4]. Laravel dapat dikombinasikan dengan bahasa pemrograman lainnya seperti Ruby on Rails, ASP.NET MVC, dan Sinatra.

## 3. Metodologi Penelitian

### 3.1 Model Konseptual

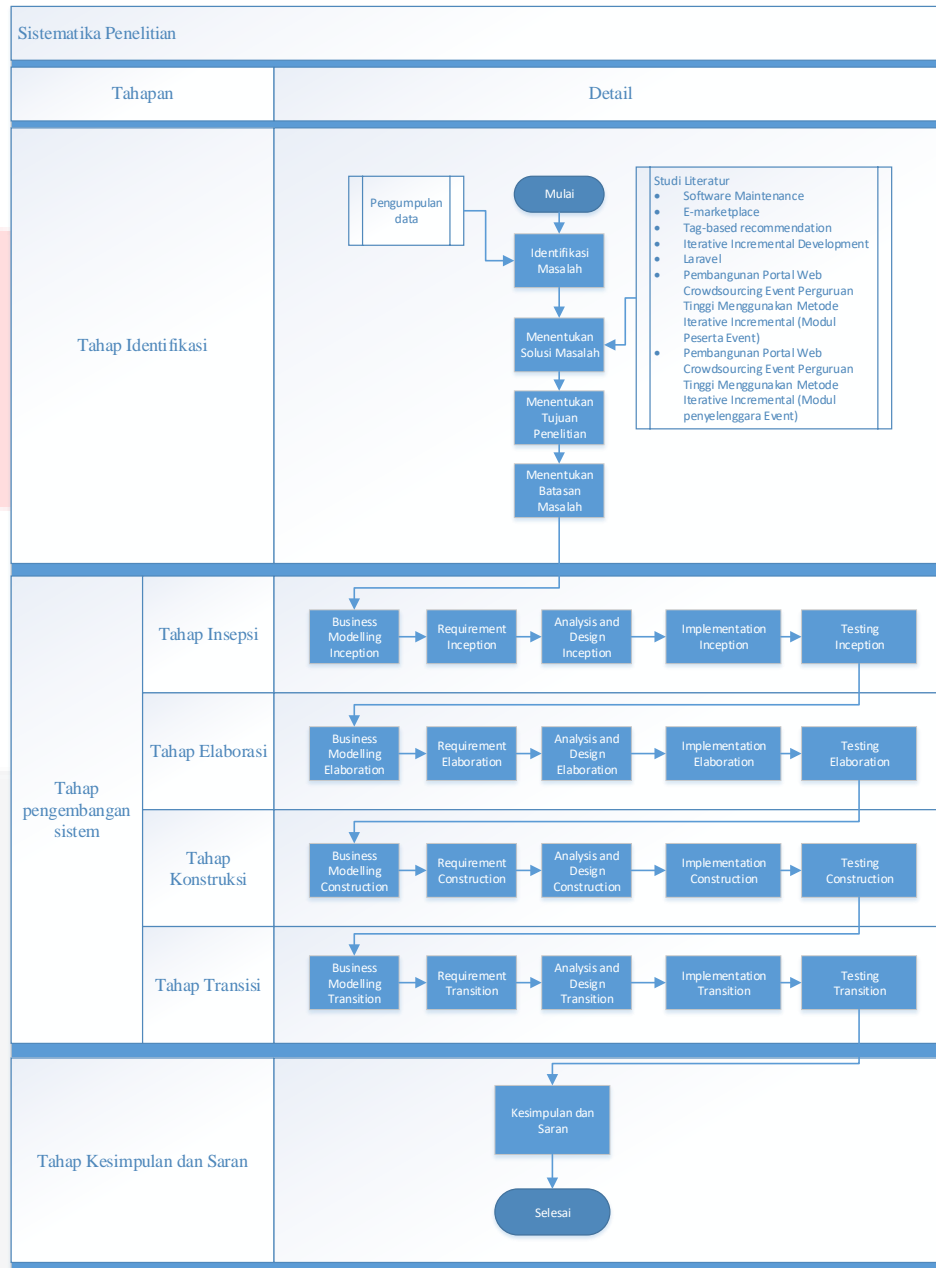
Model konseptual adalah model yang digunakan sebagai dasar untuk memahami melaksanakan dan mengevaluasi suatu riset berdasarkan disiplin sistem informasi [5].



Gambar 3-1. Model konseptual

### 3.2 Sistematika Pemecahan Masalah

Sistematika pemecahan masalah menggambarkan tahapan penelitian yang harus diselesaikan.



Gambar 3-2 Sistematisasi Penelitian

#### 4. Pembahasan

##### 4.1 Identifikasi Aktor

Identifikasi aktor berisi informasi tentang aktor yang terlibat pada sistem. Identifikasi aktor dilakukan diawal guna mempermudah perancangan fitur dan tampilan yang sesuai untuk aktor-aktor yang terlibat. Berikut adalah identifikasi aktor pada perancangan ini.

Tabel 4-1 Identifikasi Aktor

No.	Nama Aktor	Deskripsi
1.	Pengguna	Seluruh pengunjung situs baik terdaftar maupun yang tidak terdaftar.

2.	Member	Pengunjung situs yang telah mendaftarkan data diri pada situs.
3.	Admin Organisasi	Seorang member situs yang mempunyai akses penuh terhadap suatu organisasi yang terdaftar pada situs.
4	Administrator	Aktor yang memiliki akses khusus sebagai administrator sistem

#### 4.2 Tahap Insepsi

Pada tahap insepsi ini kebutuhan aktor dianalisis untuk dijadikan acuan perancangan. Berikut ini adalah kebutuhan *maintenance* terhadap sistem yang telah ada.

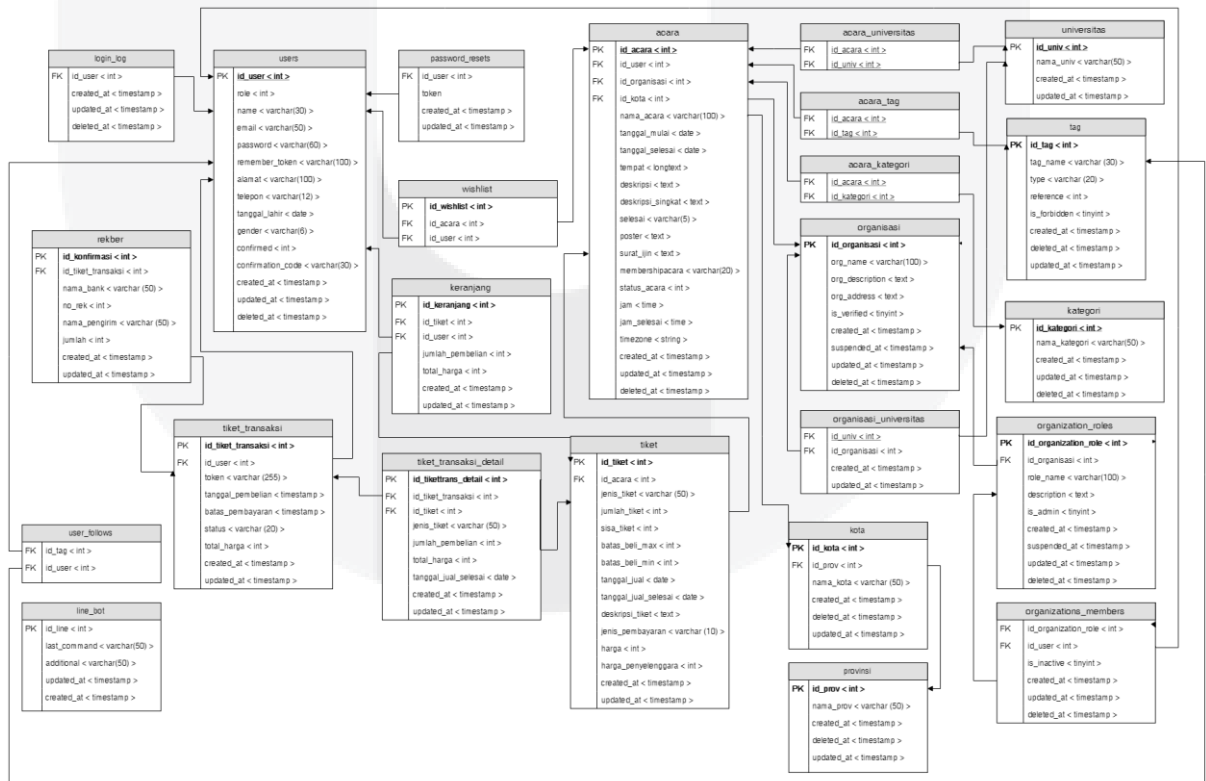
Tabel 4-2 Kebutuhan Maintenance.

No.	Nama <i>Maintenance</i>	Tipe <i>Maintenance</i>
1	<i>Upgrading</i> versi <i>Laravel</i> dari <i>Laravel</i> 4.2 ke <i>Laravel</i> 5.4.	<i>Adaptive</i>
2	Perubahan fitur <i>login</i> .	<i>Perfective</i>
3	Penambahan fitur registrasi organisasi	<i>perfective</i>
4	Penambahan fitur list “organisasi saya”	<i>perfective</i>
5	Penambahan fitur list organisasi	<i>perfective</i>
6	Penambahan fitur Edit organisasi	<i>perfective</i>
7	Penambahan fitur <i>Delete</i> organisasi	<i>perfective</i>
8	Perubahan pada fitur membuat <i>event</i> .	<i>Perfective</i>
9	Penambahan fitur list <i>event</i> saya.	<i>Perfective</i>
10	Perubahan pada fitur detail <i>event</i> .	<i>Perfective</i>
11	Perubahan pada fitur list <i>event</i>	<i>Perfective</i>
12	Perubahan pada fitur pencarian <i>event</i> .	<i>Perfective</i>
13	Perubahan fitur mengubah informasi <i>event</i> .	<i>Perfective</i>

14	Perubahan fitur hapus <i>event</i>	<i>Perfective</i>
15	Penambahan fitur list ketertarikan saya.	<i>Perfective</i>
16	Penambahan fitur tambah ketertarikan.	<i>Perfective</i>
17	Penambahan fitur hapus ketertarikan.	<i>Perfective</i>
18	Penambahan fitur personalisasi.	<i>Perfective</i>
19	Penambahan fitur lihat event yang akan datang via LINE <i>chatbot</i> .	<i>Perfective</i>
20	Penambahan fitur lihat event berdasarkan kategori via LINE <i>chatbot</i> .	<i>Perfective</i>

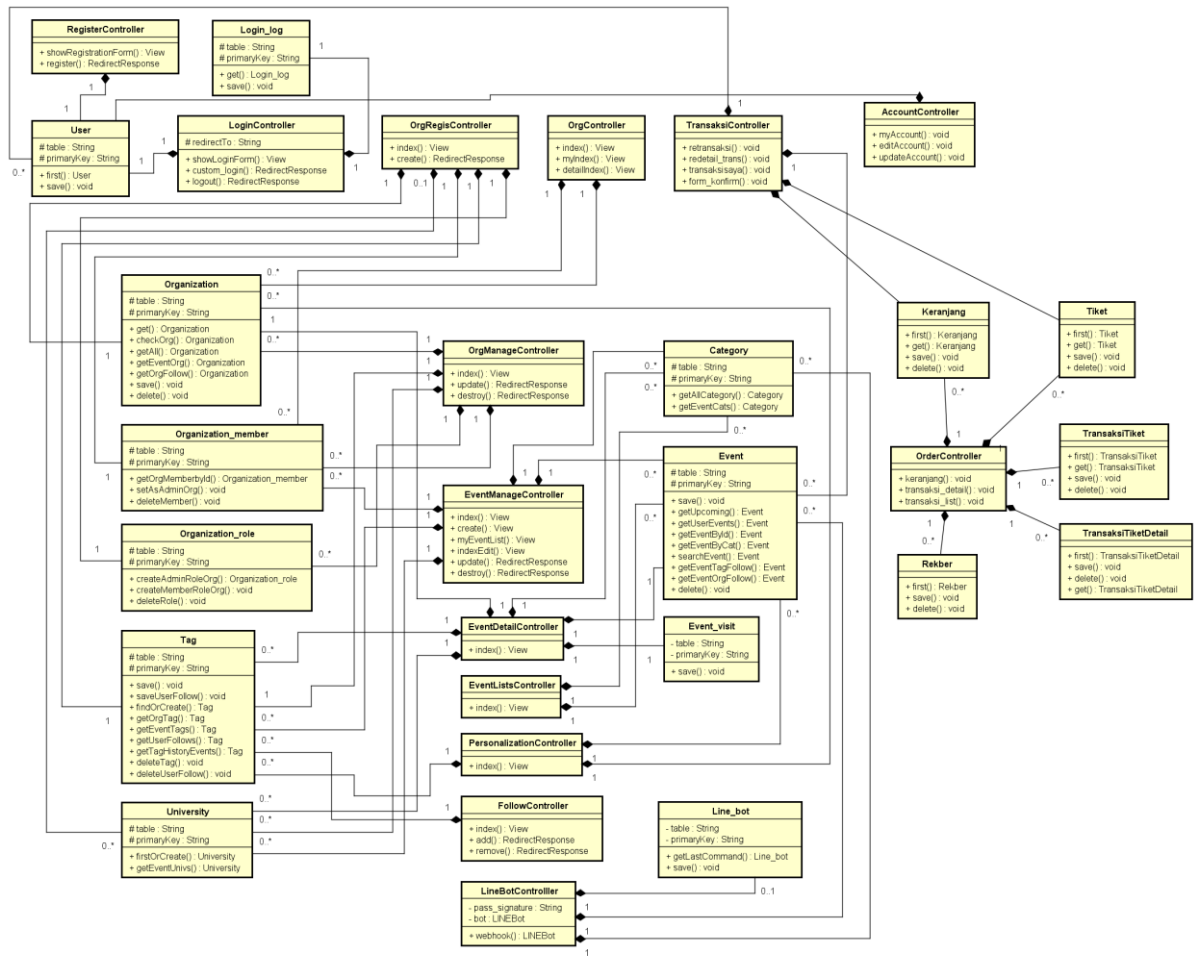
### 4.3 Tahap Elaborasi

Tahap elaborasi merupakan tahap perancangan berdasarkan analisis yang telah didapat pada tahap inerspsi. Pada tahap ini dapat digambarkan menggunakan *physical data model* dan *class diagram*.



Gambar 4-1 *physical data model*





Gambar 4-2 Class Diagram

#### 4.4 Tahap Konstruksi

Pada tahap konstruksi ini *maintenance website* dilakukan dengan merujuk pada analisis sistem pada tahap elaborasi. Pembangunan dilakukan menggunakan bahasa pemrograman PHP dengan framework Laravel 5.4. Aplikasi *database* yang digunakan adalah PostgreSQL 9.2. *Website* yang dibangun dapat diakses melalui *browser* seperti Chrome atau firefox. *Website* pun terintegrasi dengan media sosial Line.

#### 4.5 Tahap Transisi

Tahap transisi merupakan tahap pengujian terhadap aplikasi yang telah dibangun pada tahap konstruksi. Testing dilakukan dengan menggunakan *unit testing*. Pengujian ini dilakukan untuk mengecek fungsionalitas masing-masing fitur.

Tabel 4-3 Tabel Hasil Pengujian

No	Kode Testing	Fitur	Hasil yang diharapkan	Hasil yang sebenarnya	Sukses
1	TC-01	Registrasi	Dapat melakukan registrasi dengan email yang belum terdaftar.	Dapat melakukan registrasi dengan email yang belum terdaftar.	Ya
2	TC-02	Login	Dapat melakukan login dengan menggunakan <i>email</i> dan <i>password</i> yang telah terdaftar dan selanjutnya pengunjung	Dapat melakukan login dengan menggunakan <i>email</i> dan <i>password</i> yang telah terdaftar dan	Ya

			terdaftar sebagai member.	selanjutnya pengunjung terdaftar sebagai member.	
3	TC-03	Logout	Dapat melakukan login dengan memilih menu logout dan pengunjung tidak teridentifikasi sebagai member.	Dapat melakukan login dengan memilih menu logout dan pengunjung tidak teridentifikasi sebagai member.	Ya
	TC-04	Registrasi organisasi	Dapat melakukan registrasi organisasi dan data organisasi yang tersimpan sesuai dengan data yang didaftarkan.	Berhasil melakukan proses registrasi dan data organisasi yang tersimpan sesuai dengan data yang didaftarkan.	Ya
	TC-05	Melihat list "organisasi saya".	Dapat melihat list organisasi yang member tergabung didalamnya.	Berhasil melihat list organisasi yang member tergabung didalamnya.	Ya
	TC-06	Melihat list organisasi	Dapat melihat list organisasi secara keseluruhan.	Berhasil melihat list organisasi secara keseluruhan.	Ya
	TC-07	Melihat detail organisasi	Dapat melihat detail suatu organisasi.	Berhasil melihat detail suatu organisasi.	Ya
	TC-08	Edit organisasi	Dapat melakukan edit data organisasi dan data tersimpan sesuai dengan data yang diubah.	Berhasil melakukan edit data organisasi dan data tersimpan sesuai dengan data yang diubah.	Ya
	TC-09	Delete organisasi	Dapat menghapus organisasi dan organisasi yang dihapus tidak tampil pada halaman <i>list</i> organisasi.	Berhasil menghapus organisasi dan organisasi yang dihapus tidak tampil pada halaman <i>list</i> organisasi	Ya
	TC-10	Membuat Event	Admin organisasi dapat membuat event baru dan data event yang tersimpan sesuai dengan data yang didaftarkan.	Admin organisasi berhasil membuat event baru dan data event yang tersimpan sesuai dengan data yang didaftarkan.	Ya
	TC-11	Melihat list event saya.	Member dapat melihat list event yang dibuat oleh member.	Member berhasil melihat list event yang dibuat oleh member.	Ya
	TC-12	Melihat detail event	Pengguna dapat melihat informasi terkait suatu event.	Pengguna berhasil melihat informasi terkait suatu event.	Ya
	TC-13	Melihat list event	Pengguna dapat melihat seluruh event yang terdaftar pada situs.	Pengguna berhasil melihat seluruh event yang terdaftar pada situs.	Ya

	TC-14	Pencarian event	Pengguna dapat mencari event berdasarkan suatu kriteria.	Pengguna berhasil mencari event berdasarkan suatu kriteria	Ya
	TC-15	Mengubah data event	Admin organisasi dapat mengubah data event dan data tersimpan sesuai dengan data yang diubah.	Admin organisasi berhasil mengubah data event dan data tersimpan sesuai dengan data yang diubah.	Ya
	TC-16	Menghapus event	Admin organisasi dapat menghapus event dan event yang dihapus tidak tampil pada halaman <i>list</i> event.	Admin organisasi berhasil menghapus event dan event yang dihapus tidak tampil pada halaman <i>list</i> event.	Ya
	TC-17	Melihat list ketertarikan saya	Member dapat melihat list ketertarikan member dan melihat rekomendasi tag yang sesuai dengan <i>history</i> lihat event member.	Member berhasil melihat list ketertarikan member dan melihat rekomendasi tag yang sesuai dengan <i>history</i> lihat event member.	Ya
	TC-18	Tambah ketertarikan	Member dapat menambahkan suatu tag atau kategori atau organisasi ke dalam list ketertarikan.	Member berhasil menambahkan suatu tag atau kategori atau organisasi ke dalam list ketertarikan.	Ya
	TC-19	Hapus ketertarikan	Pengguna dapat melihat informasi terkait suatu event.	Pengguna berhasil melihat informasi terkait suatu event.	Ya
	TC-20	Melihat halaman personalisasi	Pengguna dapat melihat list event yang berhubungan dengan ketertarikan member.	Pengguna berhasil melihat list event yang berhubungan dengan ketertarikan member.	Ya
	TC-21	Lihat event yang akan datang via LINE chatbot	Pengguna dapat melihat list event yang akan datang melalui chat pada aplikasi Line	Pengguna dapat melihat list event yang akan datang melalui chat pada aplikasi Line	Ya
	TC-22	Lihat event berdasarkan kategori via LINE chatbot	Pengguna dapat melihat list event yang belum terselenggara berdasarkan kategori melalui chat pada aplikasi Line	Pengguna dapat melihat list event yang belum terselenggara berdasarkan kategori melalui chat pada aplikasi Line	Ya

## 5. Kesimpulan

Dengan adanya perancangan ini maka *maintenance* dapat dilakukan untuk memperbaiki dan meningkatkan kualitas portal *web*. Permasalahan yang ditemukan dapat diatasi dengan penambahan fitur baru maupun perbaikan fitur yang telah ada. Lalu dengan adanya penambahan fitur rekomendasi berbasis *tag* maka pengunjung yang telah terdaftar sebagai anggota dapat mengetahui informasi yang sesuai dengan kriterianya. Rekomendasi menggunakan algoritma *implicit-tag*. Algoritma *implicit-tag* dipilih karena rekomendasi menggabungkan referensi *tag event* dari

member dengan riwayat lihat detail *event* suatu member.

**Daftar Pustaka:**

- [1] IEEE, "EEE Standard for Software Maintenance," *IEEE Std 1219-1993*, pp. 1-45, 1993.
- [2] A. I. Damarani, *Pembangunan Portal Web Crowdsourcing Event Perguruan Tinggi Menggunakan Metode Iterative Incremental (Modul Peserta Event)*, Bandung: Telkom University, 2015.
- [3] M. Meilani, *Pembangunan Portal Web Crowdsourcing Event Perguruan Tinggi Menggunakan Metode Iterative Incremental (Modul Penyelenggara Event)*, Bandung: Telkom University, 2015.
- [4] R. Stockdale and C. Standing, "Benefits and barriers of electronic marketplace participation : an SME perspective," *Journal of Enterprise Information Management, Vol. 17 Issue: 4*, pp. 301-311, 2004.
- [5] IEEE, "ISO/IEC/IEEE International Standard for Software Engineering - Software Life Cycle Processes - Maintenance," *ISO/IEC 14764:2006*, pp. 1-58, 2006.
- [6] S. Sen, J. Vig and J. Riedl, "Tagommenders: Connecting Users to Items through Tags," in *WWW '09 Proceedings of the 18th international conference on World wide web*, Madrid, 2009.
- [7] A. Noor, *Manajemen Event*, Bandung: Alfabeta, 2009.
- [8] O. Benediktsson, D. Dalcher, K. Reed and M. Woodman, "COCOMO-Based Effort Estimation," *Software Quality Journal 11*, p. 265-281, 2003.
- [9] D. Naista, *Codeigniter vs Laravel, Kasus Membuat Website Pekerja*, Yogyakarta: Lokomedia, 2017.
- [10] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design Science In Information System Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, 2004.