

# Penerapan Arsitektur MVC pada Web Dashboard dan Aplikasi Manajemen Informasi Fakultas Industri Kreatif (MI-FIK) Universitas Telkom

1<sup>st</sup> Leonardho R. Sitanggang  
Fakultas Informatika,  
Universitas Telkom  
Bandung, Indonesia  
leonardhositanggang@student.telkomuniversity.ac.id

2<sup>nd</sup> Mira Kania Sabariah  
Fakultas Informatika,  
Universitas Telkom  
Bandung, Indonesia  
mirakania@telkomuniversity.ac.id

3<sup>rd</sup> Monterico Adrian  
Fakultas Informatika,  
Universitas Telkom  
Bandung, Indonesia  
monterico@telkomuniversity.ac.id

**Abstrak** - MI-FIK adalah sebuah perangkat lunak yang dirancang untuk mengelola informasi bagi civitas akademika Fakultas Industri Kreatif, Telkom University. Perangkat lunak ini berbasis aplikasi mobile yang dapat digunakan oleh seluruh mahasiswa, dosen, dan staff dari Fakultas Industri Kreatif. Untuk menunjang pengelolaan data yang terdapat di dalam database pada aplikasi MI-FIK, maka dirancanglah perangkat lunak berbasis web dashboard yang dikelola oleh admin. Perancangan perangkat lunak ini dibagi menjadi sisi frontend dan sisi backend, untuk menghubungkan kedua sisi tersebut diperlukan arsitektur yang dapat mengelola dan menampilkan data. MVC atau Model View Controller merupakan sebuah pola desain yang berfungsi untuk menghubungkan struktur perancangan perangkat lunak dari sisi backend maupun frontend. Arsitektur ini memungkinkan perancangan perangkat lunak dilakukan secara terpisah berdasarkan fungsi logika, interaksi ke database, dan tampilan antarmuka. Selain itu diperlukan juga Black Box Testing untuk memastikan bahwa penerapan MVC sudah terlaksana dengan benar dan sesuai kebutuhan. Setelah semua fitur untuk tiap Functional Requirement telah selesai dikembangkan dan diuji dengan Black Box Testing, maka hasil yang didapat dari 50 Functional Requirement dan 9 Non-Functional Requirement yang telah diuji. Seluruh fitur telah memenuhi kebutuhan sesuai rancangan arsitektur yang sudah dibuat.

**Kata kunci** : MVC, Dashboard, Mobile, Black Box Testing, Framework

## I. PENDAHULUAN

### A. Latar Belakang

Perangkat lunak MI-FIK adalah sebuah proyek lanjutan dari perancangan purwarupa perangkat lunak pengelolaan informasi Fakultas Industri Kreatif, Telkom University yang telah didesain oleh Ceka Elgy Dwi Putra, mahasiswa program studi S1 Desain Komunikasi Visual angkatan 2018. Fungsi dari perangkat lunak MI-FIK adalah untuk mengelola informasi bagi civitas akademika Fakultas Industri Kreatif yang dirancang untuk menyelesaikan permasalahan yang sering terjadi di lingkungan Fakultas Industri Kreatif.

Berdasarkan hasil survei yang terdapat pada

proyek sebelumnya yang telah dikerjakan oleh Ceka Elgy Dwi Putra, terdapat beberapa masalah dalam penyebaran informasi di Fakultas Industri Kreatif. Mulai dari informasi yang menumpuk, informasi yang tidak terklasifikasi, informasi yang tidak dikelola dengan baik, penggunaan media informasi yang terlalu banyak, hingga sulitnya melihat kembali informasi yang sudah berlalu. Informasi tersebut meliputi informasi akademik seperti jadwal kuliah, informasi tentang fakultas, informasi kurikulum, jadwal ujian, jadwal kuliah, jadwal mengajar, jadwal pertemuan, event, seminar, dan lainnya.

Untuk itu diperlukan perangkat lunak berbasis web, yang ringan dan mudah saat proses pemeliharaan, mengingat perangkat lunak MI-FIK akan digunakan setiap saat. Perangkat lunak berbasis web lebih cocok untuk aplikasi berbentuk dashboard karena dapat digunakan pada desktop maupun mobile dan kompatibel diberbagai versi perangkat. Selain itu, aplikasi berbasis web lebih cocok untuk menampilkan diagram dan tabel, mengingat pada perangkat lunak MI-FIK terdapat fitur analisis statistik.

Perancangan dan pembangunan web akan menggunakan framework yang dapat menerapkan konsep design pattern MVC. Konsep desain ini dipilih karena banyak digunakan saat ini terutama pada aplikasi berbasis web. Arsitektur ini terdiri dari 3 komponen penyusun. Model sebagai komponen untuk melakukan interaksi dengan database, Controller sebagai komponen untuk menerima permintaan dari pengguna, dan View sebagai komponen desain antarmuka. Dengan adanya arsitektur tersebut, akan mempermudah dilakukannya pemetaan dan pengelolaan data dari backend system ke dalam frontend system. Dan tentunya resource dari perangkat lunak dapat terstruktur baik dan mudah dikelola [2].

Setelah setiap fitur atau fungsi telah selesai dikembangkan maka akan dilakukan Black Box Testing untuk menguji apakah setiap fitur atau fungsi bekerja sesuai kebutuhan dan menghasilkan hasil yang sesuai [3].

## B. Topik dan Batasannya

Permasalahan yang dapat di angkat berdasarkan latar belakang adalah sebagai berikut:

1. Bagaimana implementasi arsitektur MVC pada web dashboard MI-FIK yang telah didesain.
2. Bagaimana implementasi arsitektur MVC pada aplikasi mobile MI-FIK yang telah didesain.
3. Apakah desain dan implementasi dari web dashboard dan aplikasi mobile MI-FIK sudah valid.

## C. Tujuan

Tujuan yang ingin dicapai dari perancangan dashboard dan perangkat lunak berbasis mobile adalah sebagaiberikut:

1. Membangun web dashboard untuk manajemen dan visualisasi data agar dapat dikelola oleh admin, dosen,dan staf.
2. Membangun aplikasi mobile sebagai media informasi bagi Fakultas Industri Kreatif, Telkom University.
3. Membangun web dashboard dan aplikasi mobile MI-FIK yang telah memenuhi kebutuhan pengguna.

## II. STUDI TERKAIT

### A. Framework

Framework adalah kumpulan library dan tools yang dapat meningkatkan desain arsitektur dari sebuah web. Dengan adanya framework, proyek perangkat lunak akan menjadi lebih ringan, konsisten, mudah dikembangkan, dan di-maintenance [4].

### B. MVC (Model, View, Controller)

MVC adalah sebuah design pattern yang terdiri dari 3 komponen penyusun. Model sebagai komponen untuk melakukan interaksi dengan database, View sebagai komponen desain antarmuka, dan Controller sebagai komponen untuk menerima permintaan dari pengguna dan menampilkan hasilnya pada View [2].

### C. PHP

PHP atau PHP: Hypertext Preprocessor adalah bahasa pemrograman scripting yang biasa dikombinasikan dengan HTML untuk membangun sebuah web. PHP berbasis server-side sehingga terdapat permintaan pada sebuah halaman web, PHP akan mengeksekusi script tersebut dan mengembalikan hasilnya kembali ke halaman web. PHP memiliki berbagai library yang biasanya digunakan untuk membuat logika perangkat lunak, maupun mengakses database [5].

### D. Dart

Dart adalah sebuah bahasa pemrograman berbasis C-style syntax yang dikembangkan oleh Google dan digunakan untuk membuat berbagai jenis aplikasi mobile, desktop, backend, dan web. Dart memiliki kesamaan dengan bahasa pemrograman C, C#, Python, Java, dan JavaScript [6].

### E. Laravel

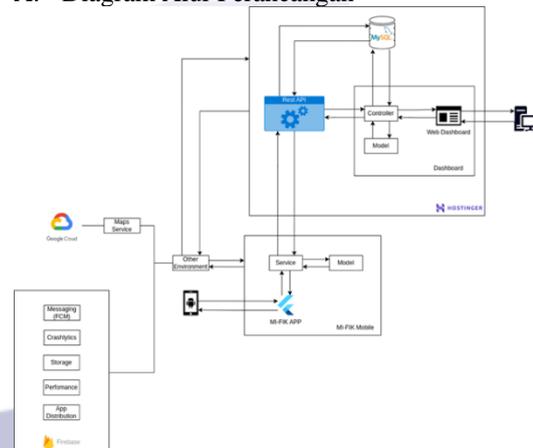
Laravel adalah framework backend yang menyediakan berbagai fitur untuk merancang sebuah aplikasi berbasis web yang modern, dinamis, dan real-time [7]. Framework ini menggunakan bahasa pemrograman PHP dan mendukung penerapan routing, validasi, caching, queues, dan penyimpanan file. Laravel terdiri dari 3 lapisan penyusun, yaitu core component extension, label extension, dan batch query [8]. Terdapat juga template Blade yang menggunakan format Bahasa PHP, template ini digunakan pada komponen View. Cara kerjanya dengan mengkompilasi code PHP dan di-cache hingga siap untuk dimodifikasi.

### F. Flutter

Flutter adalah framework open source yang menggunakan bahasa pemrograman Dart dan memungkinkan pembangunan sebuah aplikasi multi-platform dengan hanya menggunakan satu codebase [9]. Flutter memiliki performa lebih baik dibanding framework mobile lainnya, hal ini dikarenakan framework ini mengkompilasi code secara native sesuai platform yang digunakan [10].

## III. PERANCANGAN SISTEM

### A. Diagram Alur Perancangan



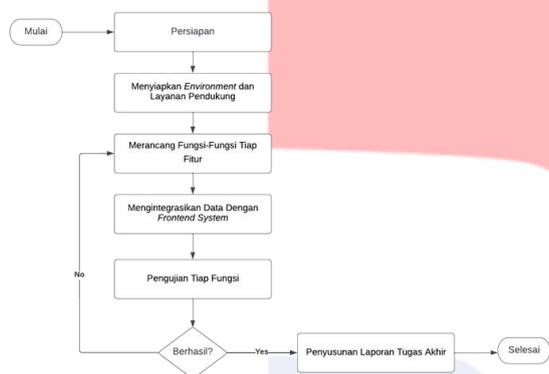
GAMBAR 1.  
Flowchart Alur Perancangan

Gambar diatas menggambarkan alur proses perancangan arsitektur MVC pada web *dashboard* dan aplikasi *mobile*. Proses perancangan dimulai dengan menganalisis dokumen yang berkaitan dengan penerapan arsitektur MVC, dokumen yang dianalisis berupa jurnal, buku, maupun dokumentasi. Selanjutnya menyiapkan Github sebagai media repository untuk menunjang version control. Dilakukan juga instalasi framework Laravel dan Flutter, instalasi layanan pendukung, dan menghubungkan aplikasi dengan database phpMyAdmin. Instalasi layanan pendukung dilakukan dalam bentuk CDN (Content Delivery Network) untuk perangkat web. Dan instalasi dalam bentuk package pada perangkat mobile.

Setelah instalasi kebutuhan perancangan berhasil, maka dilanjutkan dengan perancangan tiap

fungsi yang menyusun suatu fitur, perancangan fungsi dilakukan pada komponen Controller dan pemetaan data menggunakan Model. Kemudian akan dilakukan pengujian dengan menggunakan Black Box Testing pada setiap fungsi yang telah dirancang. Jika terdapat kesalahan pada fungsi, maka akan dilakukan perancangan ulang pada fungsi tersebut, dan jika semua fungsi berhasil lolos dari tahap pengujian, maka perancangan dokumen laporan tugas akhir dapat dimulai.

### B. Sistem Arsitektur



GAMBAR 2.  
Diagram Sistem Arsitektur

Sistem MI-FIK terdiri dari server, database MySQL, Laravel sebagai framework untuk web dashboard, Flutter untuk aplikasi mobile, dan Firebase Storage untuk penyimpanan file. Aplikasi berkomunikasi dengan server saat pertama kali dibuka. Data diakses melalui Rest API (mobile) dan Controller (web). Penggunaan Model untuk pemetaan data. Server-side (PHP) menyimpan data sesi seperti username, token, role, email, dan foto profil untuk web. Client-side (Javascript) menyimpan data hasil API dan pencarian di browser. Aplikasi mobile (Dart) menggunakan memory storage untuk data sementara yang diperbarui saat aplikasi ditutup. Skema ini mengarahkan lalu lintas data dan menyimpan data cadangan pengguna.

### C. Functional Requirement

ID	Functional Requirement	Functionality	Priority
FR-M-001	Dosen, Staff, dan Mahasiswa dapat melakukan registrasi	Autentikasi	High
FR-M-002	Dosen, Staff, dan Mahasiswa dapat melakukan login	Autentikasi	High
FR-M-	Dosen, Staff,	Autentikasi	Medium

003	dan Mahasiswa dapat melakukan logout		
FR-M-004	Dosen, Staff, dan Mahasiswa dapat melihat dan request tag baru	Tag	High
FR-M-005	Dosen, Staff, dan Mahasiswa dapat mengelola data profile	User	High

ID	Functional Requirement	Functionality	Priority
FR-W-001	Dosen dan Staf dapat melakukan registrasi	Autentikasi	High
FR-W-002	Admin, Dosen, dan Staf dapat melakukan login untuk mengakses keseluruhan web dashboard	Autentikasi	High
FR-W-003	Admin, Dosen, dan Staf dapat melakukan logout	Autentikasi	Medium
FR-W-004	Admin, Dosen, dan Staf dapat melihat daftar event yang akan berlangsung, sedang berlangsung, dan sudah selesai	Event	High
FR-W-005	Admin, Dosen, dan Staf dapat melihat detail event	Event	High

### C. Non-Functional Requirement

ID	Non Functional Requirements	Quality
NFR-G-SR-002	Autentikasi pengguna dengan mengenkripsi password menggunakan hash pada database	Security Requirement
NFR-G-SR-003	Password harus terdiri minimal 6 karakter, yang terdiri dari kombinasi huruf dan angka	Security Requirement
NFR-G-SR-004	Pengguna dan admin dapat menggunakan semua fitur pada aplikasi jika telah terdaftar pada database dan telah melewati authentication (login)	Security Requirement
NFR-G-SR-005	Audit trail : server akan mencatat aktivitas login pengguna (token, hak akses, dan waktu server)	Security Requirement
NFR-M-SF-006	Data pribadi tidak dapat dilihat oleh pengguna lain, data yang hanya bisa dilihat oleh pengguna lain yaitu nama pengguna dan foto profil.	Safety Requirement
NFR-W-UR-007	Pengguna akan diberikan dokumentasi dan menu "Help" untuk membantu mempelajari cara mengoperasikannya.	Usability

### D. Perancangan MVC Web Dashboard

Framework MVC yang digunakan pada saat perancangan sistem web dashboard yaitu Laravel versi 9.0 dengan menggunakan bahasa pemrograman

PHP sebagai bahasa dasar. penerapan arsitektur pemrograman MVC (Model, View, Controller) yang memisahkan logika pemrograman dengan tampilan antarmuka pada suatu perangkat lunak. Penerapan arsitektur ini dapat dilakukan diberbagai macam framework PHP. Namun pada perangkat lunak web dashboard MI-FIK, penulis memilih Laravel yang merupakan framework berbasis Symfony 2. Laravel dipilih karena Laravel lebih fleksibel, terukur, mudah digunakan dibanding framework serupa, dan cocok digunakan pada perancangan proyek skala besar. Selain itu, Laravel juga mendukung penerapan Restful API yang mana nantinya akan digunakan untuk menunjang perancangan aplikasi mobile [11].

#### E. Perancangan MVC Aplikasi Mobile

Framework yang digunakan pada saat perancangan sistem aplikasi mobile yaitu Flutter versi 3.0 dengan menggunakan bahasa pemrograman Dart sebagai bahasa dasar. Pada framework ini sebagian besar waktu dihabiskan untuk mengembangkan dari sisi tampilan antarmuka. Hal ini dikarenakan Flutter tidak menggunakan markup language seperti XML, sehingga tidak mendukung user-defined widgets. Penulis memilih untuk menggunakan Dart dikarenakan bahasa pemrograman ini memiliki performa yang tinggi, object-oriented, produktivitas yang tinggi, dan multi-platform [10].

#### Komponen Yang Digunakan

Nama Komponen	Justifikasi
Model	Representasi table dari database yang akan digunakan untuk pembangunan logika pada controller
Controller / Service	Logika atau interaksi dari aplikasi ke database maupun <i>cloud service</i>
View	Struktur tampilan antarmuka yang akan menampilkan data dari API, ataulangsung dari Controller.
Helpers	Fungsi-fungsi yang akan membantu perancangan logika bisa berpakonversi data, validasi, maupun template dari SQL Query
Mailers	Konfigurasi pengiriman email kepada pengguna ataupun admin
Routes	Pemanggilan fungsi dari controller atau deklarasi sebuah halaman
Rules	Definisi kategori atau tipe dari data yang akan dikelola sebelum disimpan pada database
Queue	Pemisahan proses pertukaran data pada logika yang mana akan menciptakan sebuah proses yang berjalan dilatar belakang untuk menghemat waktu
Multi Language	Dukungan multi penggunaan multi bahasa pada aplikasi
Schedule	Logika seperti controller namun dijalankan pada kurun waktu tertentu
Migrations	Komponen untuk membuat isi database berdasarkan template yang sudah didefinisikan
Factories	Komponen untuk membuat kumpulan data <i>dummy</i> berdasarkan model yang ada

#### F. Skenario Alur Pengujian

##### 1. Identifikasi Requirement

Langkah pertama adalah mengidentifikasi 50 Functional Requirement dan 9 Non-Functional Requirement yang telah ditetapkan oleh System Analyst. Functional Requirement diuji berdasarkan peran pengguna seperti Mahasiswa, Dosen/Staff, dan Admin, sedangkan Non-Functional Requirement diuji sekali karena sebagian besar berlaku secara keseluruhan pada sistem.

##### 2. Menentukan Test Case

Setelah mengidentifikasi semua requirement, langkah berikutnya adalah membuat Test Case untuk setiap Functional dan Non-Functional Requirement. Setiap Test Case berisi ID requirement, hasil yang diharapkan, hasil pengujian, dan kesimpulan. Hanya happy flow yang akan diuji, sementara pengujian dengan failed flow akan ditangani oleh Quality Assurance dengan lebih rinci.

##### 3. Melaksanakan Pengujian dan Dokumentasi

Setelah Test Case terkumpul, dilakukan pengujian pada aplikasi yang telah dirilis. Pengujian ini dilakukan di akhir iterasi setelah semua requirement terpenuhi. Setiap pengujian, baik Functional Requirement maupun Non-Functional Requirement, akan didokumentasikan dengan tangkapan layar. Setelah semua Test Case sesuai dengan requirement, penulis membuat kesimpulan dari hasil pengujian.

### IV. IMPLEMENTASI & PENGUJIAN

#### A. Persiapan

Nama Komponen	Justifikasi
Model	Model sebagai representasi tentunya akan terdapat juga defenisi dari kolom <i>primary key</i> , seluruh kolom yang dapat di isi, kolom <i>increment</i> , kolom bertipe <i>array</i> , dan kolom bertipe tanggal. Sementara pada aplikasi berbasis mobile, model digunakan untuk representasi dari <i>body</i> (dalam bentuk JSON) yang akan dikirimkan atau diambil dari Rest API nantinya.
Controller / Service	Pada aplikasi berbasis web, Controller yang berisi logika dan interaksi aplikasi ke dalam database maupun <i>cloud service</i> (Firebase), penggunaan queue, schedule, dan menentukan navigasi serta halaman yang akan ditampilkan. Sementara pada aplikasi mobile terdapat <i>service</i> , yang menjembatani Rest API dan model yang telah dibuat.
View	Pada struktur folder <i>resources/views</i> penulis bertugas untuk merancang logika perkondisian, perulangan, maupun konversi data sebelum akan dikonsumsi oleh <i>frontend developer</i> .
Helpers	Fungsi-fungsi yang akan membantu

	perancangan logika bisa berupa konversidata, validasi, maupun template dari SQL Query
Mailers	Komponen ini nantinya akan digunakan pada fitur menambahkan event bagipengguna maupun admin dan pengingat event bagi pengguna. Sementara bagiadmin, akan mendapat mailer pada Task Scheduling. Pada komponen ini, penulis akan menyiapkan konfigurasi email admin yang akan digunakan untuk mengirim email ke pengguna beserta logika nya.
Routes	Penulis akan merancang <i>routes</i> web yang akan memuat URL yang berisi pemanggilan fungsi dari controller yang telah dirancang dan nantinya hanya digunakan pada aplikasi berbasis web.
Rules	Komponen ini berisi definisi dari kategori data-data apa saja yang akandiperbolehkan untuk dikelola di dalam database.
Queue	Pada saat proses pertukaran data di dalam sistem, terdapat salah satu proses didalam logika yang mengonsumsi waktu yang signifikan. Proses tersebut terjadi saat sistem mengirimkan email ke pengguna menggunakan protokol SMTP (Simple Mail Transfer Protocol). Proses pengiriman email yang lama,akan menghambat waktu <i>loading</i> yang dialami oleh pengguna, oleh karena itu diperlukan sebuah skenario dimana sistem akan memisahkan proses pertukaran data menjadi 2 cabang.
Multi Language	Penulis mengembangkan sebuah fitur <i>multi-language</i> yang memungkinkan pengguna untuk memilih antara Bahasa Indonesia dan Bahasa Inggris. Fitur ini nantinya akan diterapkan pada sisi tampilan saja (tidak terdapat pada <i>controller</i> ). Untuk pilihan Bahasa yang digunakan akan disimpan pada <i>session</i> .
Schedule	Schedule merupakan komponen logika sama seperti controller tetapi nantinyafungsi ini akan dijalankan secara otomatis dalam periode waktu tertentu
	sesuai yang didefinisikan oleh Admin untuk mengelola data yang terdapat didalam database. Fungsi utama dari penggunaan komponen ini pada aplikasiMI-FIK ialah untuk menghapus data yang sudah tidak digunakan lagi setelahbatas waktu yang ditentukan, sebagai report admin, dan pengingat event atau task.
Migrations	Migrations hanya akan digunakan untuk membuat tabel beserta kolom sesuaidengan yang sudah didefinisikan. Dimana setiap kolom dan relasi akan menyesuaikan diagram ERD yang sudah dibuat oleh System Analyst.
Factories	Komponen ini digunakan untuk membuat data palsu ( <i>dummy</i> ) sesuai dengantable dan kolom pada model yang telah dibuat. Data palsu ini digunakan untukmembantu dalam proses <i>development</i> maupun <i>staging</i> .

## B. Menyiapkan *Environment* dan Layanan Pendukung

Github sebagai media penyimpanan source code untuk kedua aplikasi. Semua tim anggota terlibat dalam repositori dan membuat branch pribadi. Penulis sebagai lead developer memantau dan mengelola progress fitur di kedua aplikasi. Setiap developer bekerja pada branch masing-masing, dan ketika selesai, mereka melakukan commit. Setelah beberapa commit disetujui oleh penulis, developer tersebut melakukan pull request ke branch utama. Penulis, sebagai lead developer dan backend developer, bertanggung jawab atas instalasi dan pembaruan packages sesuai kebutuhan aplikasi.

## C. Merancang Fungsi Tiap-Tiap Fitur & Integrasi Data dengan Front-End

TABLE 1.  
Contoh Perancangan Fitur MI-FIK Web

ID	PHP & SQL Syntax
FR-W-007	<p>- Validasi variable</p> <pre>\$validator = Validation::getValidateEvent(\$request); if (\$validator-&gt;fails()) {     \$errors = \$validator-&gt;messages();     return redirect()-&gt;back()-&gt;with('failed_message', \$errors); } else {- }</pre> <p>- Inisialisasi variable</p> <pre>\$tag = Converter::getTag(\$request-&gt;content_tag); \$fulldate_start = Converter::getFullDate(\$request-&gt;content_date); \$fulldate_end = Converter::getFullDate(\$request-&gt;content_date); \$user_id = Generator::getUserId2(session()-&gt;get('role_key')); \$slug = Generator::getSlugName(\$request-&gt;content_title, "content"); \$uuid = Generator::getUUID();  if(\$request-&gt;content_image    \$request-&gt;content_image != ""){     \$imageURL = \$request-&gt;content_image; } else {     \$imageURL = null; }  if(!isset(\$status)){     \$draft = 1;     \$failed_attach = true; }</pre> <p>- Pertukaran data</p> <pre>\$header = ContentHeader::create([     'id' =&gt; \$uuid,     'slug_name' =&gt; \$slug,     'content_title' =&gt; \$request-&gt;content_title,     'content_desc' =&gt; \$request-&gt;content_desc,     'content_data_start' =&gt; \$fulldate_start,     'content_data_end' =&gt; \$fulldate_end,     'content_reminder' =&gt; \$request-&gt;content_reminder,     'content_image' =&gt; \$imageURL,     'is_draft' =&gt; \$draft,     'created_at' =&gt; date("Y-m-d H:i"),     'updated_at' =&gt; null,     'deleted_at' =&gt; null, ]);  if(\$tag    \$request-&gt;has('content_attach')){     function getFailedAttach(\$failed, \$att_content){         if(\$failed){             return null;         } else {             return \$att_content;         }     }      \$detail = ContentDetail::create([         'id' =&gt; Generator::getUUID(),         'content_id' =&gt; \$uuid, //for now         'content_attach' =&gt; getFailedAttach(\$failed_attach, \$request-&gt;content_attach),         'content_tag' =&gt; \$tag,         'content_loc' =&gt; \$request-&gt;content_loc,         'created_at' =&gt; date("Y-m-d H:i"),         'updated_at' =&gt; null,     ]); }</pre> <p>- Mengirim hasil event yang telah dibuat ke email pengguna</p> <pre>Mail::to(session()-&gt;get('email_key'))-&gt;send(new OrganizerEmail(\$header, \$detail));</pre>

## D. Merancang Non-Functional Requirement

TABLE 2.  
Contoh Perancangan Fitur MI-FIK Web

<b>ID</b>	<b>PHP &amp; SQL Syntax</b>
NFR-G-SR-003	Password harus terdiri minimal 6 karakter, yang terdiri dari kombinasi huruf dan angka <b>Implementasi</b> <pre> public static function getValidateUserRegister(\$request){     return Validator::make(\$request-&gt;all(), [         'username' =&gt; 'required min:6 max:30 string',         'first_name' =&gt; 'required min:2 max:35 string',         'last_name' =&gt; 'nullable min:2 max:35 string',         'password' =&gt; 'required min:6 max:50 string',         'batch_year' =&gt; 'nullable numeric',         'email' =&gt; 'required min:11 max:75 string email'     ]); }  public static function hasNumber(\$val) {     \$pattern = '/\d/';     return preg_match(\$pattern, \$val); } </pre>

Memiliki akses notifikasi
Keterangan : Saat aplikasi pertama kali dijalankan

TABLE 3.  
Contoh hasil pengujian BlackBox pada Functional Requirement

Requirement Code	Hasil		Kesimpulan
	Yang diharapkan	Pengujian	
R-W-001	Dosen / Staf berhasil mendaftar akun dan mengirim permintaan request	Sistem menampilkan pesan "Your registration process has completed. You can back to landing page, login using your new account, and wait for admin to approve your account"	Sesuai
R-W-002	Admin dapat melakukan logindan diarahkan ke halaman utama (Home) pada aplikasi	Sistem mengarahkan Admin ke halaman Home dan menampilkan popup report request (opsional)	Sesuai
R-W-003	Admin dapat melakukan sign-out yang terdapat pada navbar	Sistem mengarahkan Admin ke halaman Landing	Sesuai
R-W-004	Admin dapat melihat seluruh event yang akan berlangsung, sedang berlangsung, dan baru saja usai	Sistem menampilkan seluruh event yang ada sesuai periode	Sesuai
R-W-005	Admin dapat melihat detail event	Sistem menampilkan detail event yang terdiri dari gambar header, judul, foto profil pengunggah event, deskripsi, attachment (url, pdf, video, ataupun gambar), lokasi, dan periode event	Sesuai

### E. Pengujian Tiap-Tiap Fungsi

Test Environment Aplikasi MI-FIK Berbasis Web	
Penguji	Leonardho R Sitanggang (Backend Developer)
Perangkat	Laptop
Merk	Macbook Pro M1
Sistem Operasi	MacOs Monterey 12.5
Koneksi Internet	WiFi
Batasan / Izin	Terkoneksi internet Menggunakan browser Chrome / Mozilla Firefox Memiliki akses penyimpanan Keterangan : digunakan saat akan mengunggah file foto profil pengguna (FR-W-007), foto header event, dan file attachment

Test Environment Aplikasi MI-FIK Berbasis Mobile	
Penguji	Leonardho R Sitanggang (Backend Developer)
Perangkat	Smartphone
Merk	Samsung Galaxy A71
Sistem Operasi	Android 13
Koneksi Internet	Wifi & Data selular
Batasan / Izin Aplikasi :	Terkoneksi internet Memiliki kapasitas penyimpanan 40 mb pada saat instalasi Memiliki akses lokasi / GPS Keterangan : Digunakan saat menambahkan event (FR-M-014), melihat detail event (FR-M-006), dan menampilkan lokasi pada halaman home Memiliki akses kamera Keterangan : Digunakan saat menambahkan event (FR-M-014), melakukan registrasi (FR-M-001), dan mengedit profil (FR-M-005) Memiliki akses penyimpanan Keterangan : Digunakan saat mengedit profil (FR-M-005), dan menambahkan event (FR-M-014) Memiliki akses mikrofon Keterangan : Digunakan saat menambahkan event (FR-M-014), dan melihat detail event (FR-006),

TABLE 4.  
Contoh hasil pengujian BlackBox pada Non-Functional Requirement

Requirement Code	Hasil		Kesimpulan
	Yang diharapkan	Pengujian	
NFR-G-SR-002	Password akun pengguna berhasil di-hash	Pada database, di table users kolom password. Password pengguna telah di-hash menggunakan Bcrypt	Sesuai
NFR-G-SR-003	Password akun pengguna /admin akan valid jika memiliki panjang karakter lebih dari 6	Pengguna berhasil mendaftarkan akunnya pada proses registrasi	Sesuai
NFR-G-SR-004	Pengguna / admin dapat masuk ke dalam aplikasi secara menyeluruh	Sistem mengarahkan pengguna / admin ke halaman Home	Sesuai

## V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi berdasarkan sequence diagram dan requirement yang telah dibuat oleh System Analyst, dan pengujian yang menggunakan Blackbox Testing pada aplikasi MI-FIK berbasis mobile dan web. Hasil yang didapat ialah, telah berhasil dilaksanakan Blackbox Testing yang merupakan bagian Alpha Testing untuk 21 implementasi Functional Requirement MI-FIK Web dan 29 implementasi Functional Requirement MI-FIK Mobile. Dan juga terdapat 9 implementasi Non-Functional Requirement. Pengujian Blackbox Testing

dilaksanakan sejak akhir Juli 2023 hingga akhir Agustus 2023. Dari seluruh requirement yang telah diuji, didapati bahwa semua requirement yang dirancang oleh System Analyst telah dikembangkan oleh penulis selaku backend developer dan sesuai dengan kebutuhan yang diharapkan.

Adapun saran dari penulis untuk menggunakan metode pengujian White Box Testing pada tiap komponen code yang ada untuk menguji aspek-aspek yang tidak dapat ter-cover sebelumnya oleh Black Box Testing. Selain itu, menerapkan proses pertukaran data cloud service pada sisi backend untuk memastikan bahwa pertukaran data terjadi dengan aman.

#### REFERENSI

- [1] A. Sunardi and S. , "MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based," vol. 157, pp. 134-141, 2019.
- [2] R. Sanjaya, A. Herliana, F. Y. S. Rahayu and T. Suhartini, "Sistem Informasi Manajemen Bisnis dan Keuangan UMKM Menggunakan Model MVC Pada Framework Laravel," vol. 1, pp. 447-485, 2018.
- [3] A. Janes, A. Sillitti and G. Succi, "Effective Dashboard Design," 2013.
- [4] M. Laaziri, K. Benmoussa, S. Khouilji, K. M. Larbi and A. E. Yamami, "A comparative study of laravel and symfony PHP frameworks," vol. 9, pp. 704-712, 2019.
- [5] H. R. Yu, "Design and implementation of web based on Laravel framework," pp. 301-304, 2015.
- [6] L. R. Sitanggang, "Frontend," 29 11 2022. [Online]. Available: <https://laravel.com/docs/9.x/frontend>.
- [7] H. N. Haq, M. F. Hasbi and . H. Maulid, "My TelU : Aplikasi mobile untuk civitas academia Telkom University berbasis Flutter," 2021.
- [8] L. R. Sitanggang, 29 11 2022. [Online]. Available: <https://flutter.dev/>.
- [9] S. Sinha, *Quick Start Guide to Dart Programming : Create High-Performance Applications for the Web and Mobile*, Howrah, West Bengal, 2020, pp. 1-6.
- [10] H. E. Williams and D. Lane, *Web Database Applications with PHP and MySQL: Building Effective Database-Driven Web Sites*, O'Reilly Media, 2004.
- [11] R. Per, "A Survey of Unit Testing Practices," vol. 23, pp. 22-29, 2006.
- [12] Y. Cheon and C. Chavez, "Creating Flutter Apps from Native Android Apps," 2021.
- [13] A. A. Sawant, P. H. Bari and P. M. Chawan, "Software Testing Techniques and Strategies," vol. 2, pp. 980-986, 2012.