

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Era globalisasi yang kini terus berkembang berdampak pada pemanfaatan teknologi informasi secara ekstrem pada hampir semua lapisan masyarakat yang sudah tidak dapat dihindarkan lagi. Kemudahan dan kecepatan yang ditawarkan dalam mencari, menemukan dan mengolah informasi selalu menjadi kunci yang selalu diutamakan. Para vendor pengembang teknologi informasi terus berlomba-lomba dan bersaing dalam meningkatkan kualitas dan kuantitas produk serta layanan mereka.

Dalam proses pengembangan tersebut tentunya melibatkan berbagai macam alat bantu untuk kemudahan dalam proses development hingga proses finalisasi sampai akhirnya suatu produk bisa dinikmati oleh masyarakat. Salah satu aspek terpenting dalam sebuah proses development adalah time frame atau rincian waktu kapan sebuah proses development dimulai dan diakhiri.

Secara umum, sebuah time frame menyajikan informasi mengenai berapa lama sebuah proses development berjalan. Sering juga disertai adanya rincian task-task apa saja yang dikerjakan, berapa banyak jumlah developer, total man-days dan lain-lain. Kebanyakan para vendor membuat sebuah time frame dengan menggunakan bantuan software office seperti Microsoft Excel atau Open Office Calc. Tentu saja cara penyajian seperti itu dirasa cukup mumpuni, akan tetapi masih memiliki banyak kelemahan-kelemahan yang ditimbulkan. Oleh sebab itu, penulis mencoba membuat sebuah aplikasi yang terintegrasi untuk memenuhi kebutuhan secara spesifik dalam pembuatan dan penyajian time frame.

Aplikasi ini menggunakan arsitektur Model-View-Controller atau disingkat MVC. MVC adalah sebuah design pattern yang memisahkan tiga layer penting dari sebuah aplikasi web dimana Model mengacu pada data, View pada user interface atau tampilan dan Controller pada navigasi dan input, yaitu bagaimana sebuah View bereaksi terhadap interaksi yang dilakukan user. Arsitektur ini dibuat oleh Trygve Reenskaug, seorang computer engineer berkebangsaan Norwegia yang bekerja pada Smalltalk-80 pada tahun 1979. Keuntungan menggunakan arsitektur ini adalah pemisahan terhadap basis kode, fokus dalam proses development karena terbagi menjadi tiga layer dan pengembangan dapat dilakukan secara paralel oleh tim yang berbeda. Dari ketiga keuntungan tersebut bisa dilihat bahwa dengan pendekatan model ini, development dapat dilakukan lebih cepat dan terstruktur. Selain itu, arsitektur ini adalah yang paling banyak digunakan oleh vendor dalam industri software khususnya yang berbasis aplikasi web.

Karena perkembangan teknologi, maka muncul beberapa arsitektur yang berkaitan dengan masalah separasi atau pemisahan terhadap presentation layer selain arsitektur MVC. Ada dua design pattern yang mengadopsi MVC sebagai generasi selanjutnya yaitu Model-View-Presenter atau MVP dan Model-View-ViewModel atau MVVM.

MVP pertama kali diperkenalkan oleh Mike Potel dari Taligent (IBM) pada tahun 1996. MVP lebih condong ke sebuah Presenter yang merupakan sebuah class yang meng-encapsulate Selection, Command dan Interactor yang pada model MVC, fungsi dari Controller dianggap redundansi karena sebuah sistem operasi yang baru telah menyediakan hampir semua fungsi-fungsi dari Controller di dalam View itu sendiri.

MVVM pertama kali diperkenalkan oleh John Gossman, seorang arsitek WPF (Windows Presentation Foundation) pada tahun 2005. Ia menjelaskan bahwa MVVM dibuat berdasarkan perkembangan arsitektur User Interface dimana sebuah UI diposisikan oleh seorang designer daripada seorang developer. Sama

halnya dengan MVC, View di dalam konsep ini dapat berperan sebagai data binding dan data display, namun tanpa adanya code tapi berupa ekstensi XAML.

Dari ketiga model arsitektur tersebut baik MVC, MVP dan MVVM menawarkan kelebihan masing-masing. Namun penulis disini lebih memilih menggunakan konsep MVC karena fleksibilitas, generalitas dan fungsionalitas yang ditawarkan oleh *C (Controller)* itu sendiri telah memberikan sebuah gambaran konsep yang sangat jelas terhadap sebuah separasi atau pemisahan antara *MV (Model dan View)*.

1.2 Batasan Masalah

Dalam pembuatan dan penyajian time frame menggunakan software office, kesulitan pertama kali yang kita alami adalah dalam membuat rancangan desain. Meskipun software tersebut sudah menyediakan fitur-fitur lengkap seperti table, cell, column dan sebagainya tetapi tetap tidak tersedia fitur otomatisasi terhadap elemen-elemen dalam sebuah time frame. Karena memang pada dasarnya software tersebut tidak ditujukan untuk hal ini.

Selain itu, data yang kita sajikan menjadi tidak relevan lagi ketika adanya perubahan dalam sebuah time frame, misalkan adanya perubahan jumlah hari, perubahan jumlah developer dan lain sebagainya. Kita terpaksa untuk mendesain ulang time frame yang sudah kita buat karena tidak adanya otomatisasi data.

Dalam hal ini, penulis mencoba menggambarkan secara spesifik terhadap pembuatan dan manajemen sebuah time frame. Dimana aspek kemudahan, kecepatan dan cara penyajian informasi yang penulis utamakan.

1.3 Tujuan Penelitian

Tujuan dibuatnya aplikasi ini adalah memberikan kemudahan dalam mengolah dan menyajikan sebuah time frame, sehingga dapat membantu para pengembang software untuk dapat mempercepat sebuah proses development produk mereka.

1.4 Metode Penelitian

Metode-metode yang dilakukan adalah :

✓ Analisis Masalah

- Pengumpulan informasi dan masalah berdasarkan data di lapangan
- Membuat ide dan solusi

✓ Perancangan

- Membuat rancangan aplikasi yang terdiri dari modul, fitur dan flow
- Membuat EER, struktur navigasi dan desain mockup

✓ Implementasi

- Metode pengembangan menggunakan Agile
- Sistem yang digunakan berbasis terdistribusi atau Distributed-System yang berbentuk Aplikasi Web
- Teknologi yang diadopsi berbasis Multiplatform menggunakan bahasa pemrograman Java
- Pengolahan data menggunakan DBMS MySQL

- Tools development yang digunakan adalah Eclipse IDE, Apache Maven, Balsamiq Mockup dan MySQL Workbench
- Framework dan Library yang digunakan adalah Spring Injection, Spring MVC, Spring Security, JPA, Hibernate, Thymeleaf, Bootstrap, JQuery dan Highcharts JS
- Server Container menggunakan Apache Tomcat

✓ **Uji Coba**

- Uji coba yang dilakukan dimulai dari proses deployment, pengujian aplikasi, positif test – negative test hingga proses upload

1.5 Sistematika Penulisan

Pengulisan ilmiah ini terdiri atas 4 bab dimana setiap bab terdiri dari beberapa sub bab dengan sistematika sebagai berikut :

BAB I : PENDAHULUAN

Bab ini membahas tentang latar belakang masalah, batasan masalah, tujuan penulisan ilmiah dan metode penelitian serta sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini menjelaskan mengenai teori yang berhubungan dengan penulisan ilmiah ini seperti Java dan MVC Pattern.

BAB III : PEMBAHASAN

Bab ini menjelaskan tahapan-tahapan pembuatan aplikasi. Dimulai dengan rancangan EER, modul dan fitur, struktur navigasi, mockup dan struktur project.

BAB IV : PENUTUP

Bab ini berisi tentang kesimpulan pembahasan serta saran-saran.

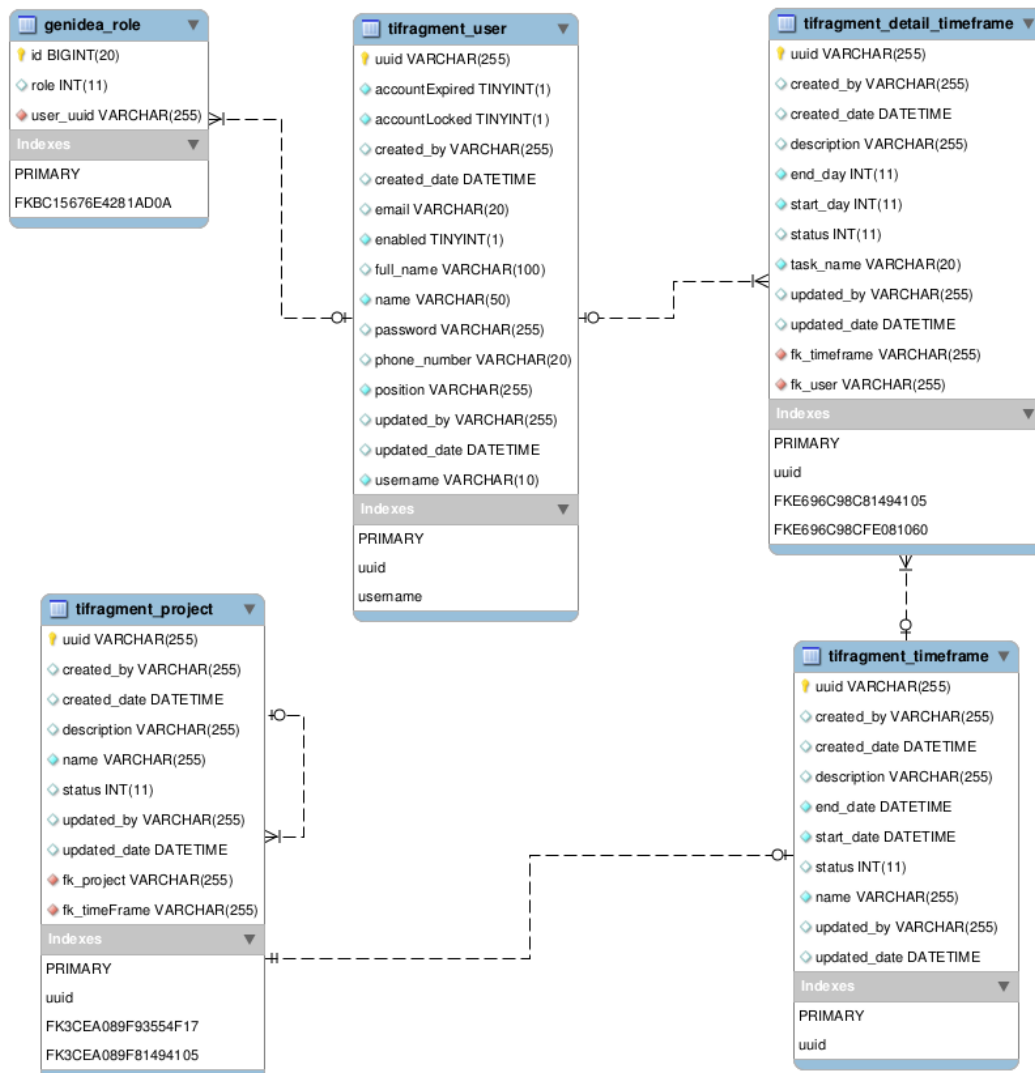
BAB III

PEMBAHASAN

Dalam pembuatan aplikasi ini, ada beberapa tahapan yang dilakukan penulis. Yang pertama yaitu membuat rancangan Extended Entity-Relationship Model (EER), membuat table schema pada MySQL, lalu selanjutnya adalah rancangan struktur navigasi, rancangan user interface menggunakan model mockup, setelah itu proses development menggunakan bantuan Eclipse IDE sebagai development environment. Tahapan terakhir adalah proses deployment dan testing.

3.1 Rancangan Extended Entity-Relationship Model (EER Model)

EER adalah sebuah data konseptual model yang menggabungkan ekstensi kedalam Entity-Relationship (ER) model yang digunakan dalam desain database. Model EER mencakup semua konsep yang terdapat dalam ER. Selain itu termasuk konsep dari subclass dan superclass yang menggambarkan hubungan-hubungan yang dibentuk oleh masing-masing entitas. Di aplikasi ini terdapat lima entitas yang digunakan yaitu Role, User, Project, Time Frame dan Detail Time Frame.



Gambar 1 - EER Aplikasi

Pada EER diatas terdapat empat tabel master yaitu role, user, project dan time frame serta satu tabel turunan atau normalisasi yaitu detail time frame. Pada tabel project terdapat hubungan bi-directional pada tabel itu sendiri yang digunakan untuk mengisi data subproject. Tabel seperti itu disebut juga dengan Tabel Recursive. Pada tabel project dan time frame terdapat hubungan One-To-One karena setiap satu project memiliki satu time frame.

3.2 Membuat Database dan Table pada MySQL

Berikut script-script yang digunakan untuk membuat database dan table-table yang digunakan pada aplikasi :

Membuat Database

```
CREATE DATABASE `tifragment`
```

Tabel Role

```
CREATE TABLE IF NOT EXISTS `genidea_role` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `role` int(11) DEFAULT NULL,  
  `user_uuid` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FKBC15676E4281AD0A` (`user_uuid`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Tabel User

```
CREATE TABLE IF NOT EXISTS `tifragment_user` (  
  `uuid` varchar(255) NOT NULL,  
  `accountExpired` tinyint(1) NOT NULL,  
  `accountLocked` tinyint(1) NOT NULL,  
  `created_by` varchar(255) DEFAULT NULL,  
  `created_date` datetime DEFAULT NULL,  
  `email` varchar(20) DEFAULT NULL,  
  `enabled` tinyint(1) NOT NULL,  
  `full_name` varchar(100) DEFAULT NULL,  
  `name` varchar(50) NOT NULL,  
  `password` varchar(255) DEFAULT NULL,  
  `phone_number` varchar(20) DEFAULT NULL,  
  `position` varchar(255) NOT NULL,
```

```

`updated_by` varchar(255) DEFAULT NULL,
`updated_date` datetime DEFAULT NULL,
`username` varchar(10) NOT NULL,
PRIMARY KEY (`uuid`),
UNIQUE KEY `uuid` (`uuid`),
UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel Project

```

CREATE TABLE IF NOT EXISTS `tifragment_project` (
  `uuid` varchar(255) NOT NULL,
  `created_by` varchar(255) DEFAULT NULL,
  `created_date` datetime DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `name` varchar(255) NOT NULL,
  `status` int(11) DEFAULT NULL,
  `updated_by` varchar(255) DEFAULT NULL,
  `updated_date` datetime DEFAULT NULL,
  `fk_project` varchar(255) DEFAULT NULL,
  `fk_timeFrame` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`uuid`),
  UNIQUE KEY `uuid` (`uuid`),
  KEY `FK3CEA089F93554F17` (`fk_project`),
  KEY `FK3CEA089F81494105` (`fk_timeFrame`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel Time Frame

```

CREATE TABLE IF NOT EXISTS `tifragment_timeframe` (
  `uuid` varchar(255) NOT NULL,
  `created_by` varchar(255) DEFAULT NULL,
  `created_date` datetime DEFAULT NULL,

```

```

`description` varchar(255) DEFAULT NULL,
`end_date` datetime NOT NULL,
`start_date` datetime NOT NULL,
`status` int(11) DEFAULT NULL,
`name` varchar(255) NOT NULL,
`updated_by` varchar(255) DEFAULT NULL,
`updated_date` datetime DEFAULT NULL,
PRIMARY KEY (`uuid`),
UNIQUE KEY `uuid` (`uuid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel Detail Time Frame

```

CREATE TABLE IF NOT EXISTS `tifragment_detail_timeframe` (
  `uuid` varchar(255) NOT NULL,
  `created_by` varchar(255) DEFAULT NULL,
  `created_date` datetime DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `end_day` int(11) NOT NULL,
  `start_day` int(11) NOT NULL,
  `status` int(11) DEFAULT NULL,
  `task_name` varchar(20) NOT NULL,
  `updated_by` varchar(255) DEFAULT NULL,
  `updated_date` datetime DEFAULT NULL,
  `fk_timeframe` varchar(255) DEFAULT NULL,
  `fk_user` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`uuid`),
  UNIQUE KEY `uuid` (`uuid`),
  KEY `FKE696C98C81494105` (`fk_timeframe`),
  KEY `FKE696C98CFE081060` (`fk_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Tabel Constraints

```
ALTER TABLE `tifragment_detail_timeframe`  
ADD CONSTRAINT `FKE696C98C81494105` FOREIGN KEY  
(`fk_timeframe`) REFERENCES `tifragment_timeframe` (`uuid`),  
ADD CONSTRAINT `FKE696C98CFE081060` FOREIGN KEY (`fk_user`)  
REFERENCES `tifragment_user` (`uuid`);
```

```
ALTER TABLE `tifragment_project`  
ADD CONSTRAINT `FK3CEA089F81494105` FOREIGN KEY  
(`fk_timeFrame`) REFERENCES `tifragment_timeframe` (`uuid`),  
ADD CONSTRAINT `FK3CEA089F93554F17` FOREIGN KEY (`fk_project`)  
REFERENCES `tifragment_project` (`uuid`);
```

3.3 Fitur Aplikasi

Berikut adalah fitur-fitur yang terdapat dalam aplikasi ini :

1. User Management
2. Project dan Time Frame Management
3. Penyajian hasil akhir secara grafikal menggunakan chart berbentuk Stacked-Bar
4. Responsive Theme

3.4 Modul Aplikasi

1. Registrasi User
2. User Login
3. CRUD Project
4. CRUD Time Frame
5. Dashboard

3.5 User Role

Aplikasi ini mempunyai dua user role yaitu :

1. Supervisor → dapat membuat user dengan role project manager dan developer, membuat, menghapus, mengubah project atau timeframe, serta dapat menambahkan anggota dari project tersebut.
2. Project Manager → dapat membuat, menghapus, mengubah project atau timeframe, serta dapat menambahkan anggota dari project tersebut
3. Developer → dapat melihat project dan time frame yang sudah dibuat untuk user tersebut

3.6 Struktur Navigasi

Struktur navigasi adalah struktur atau alur dari suatu program yang merupakan rancangan hubungan (rantai kerja) dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan Website. Menentukan struktur navigasi merupakan hal yang sebaiknya dilakukan sebelum membuat suatu Website. Ada empat macam bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan Website, yaitu : ***Struktur Navigasi linier, Struktur Navigasi Non-linier, Struktur Navigasi Hirarki, Struktur Navigasi Campuran.***

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

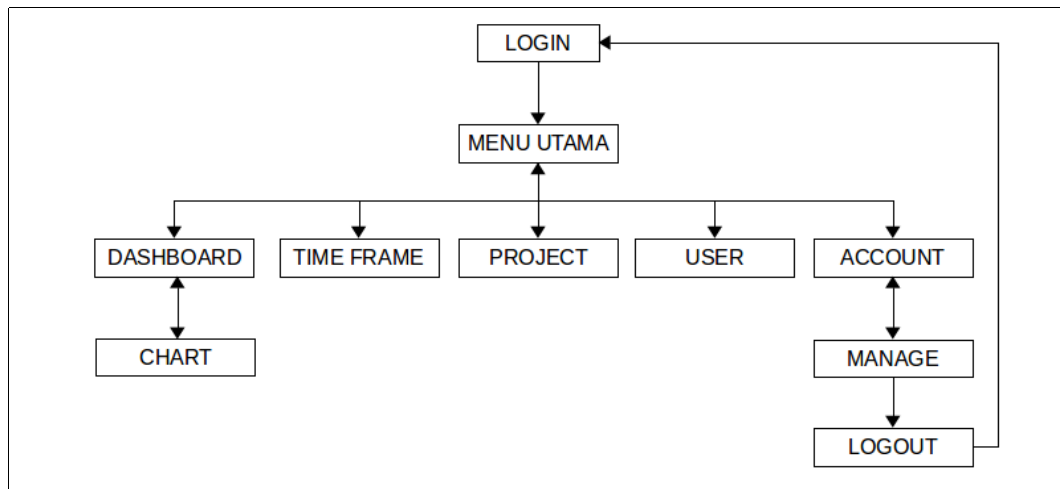
Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan

membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai Master Page (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut Slave Page (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan secara linier.

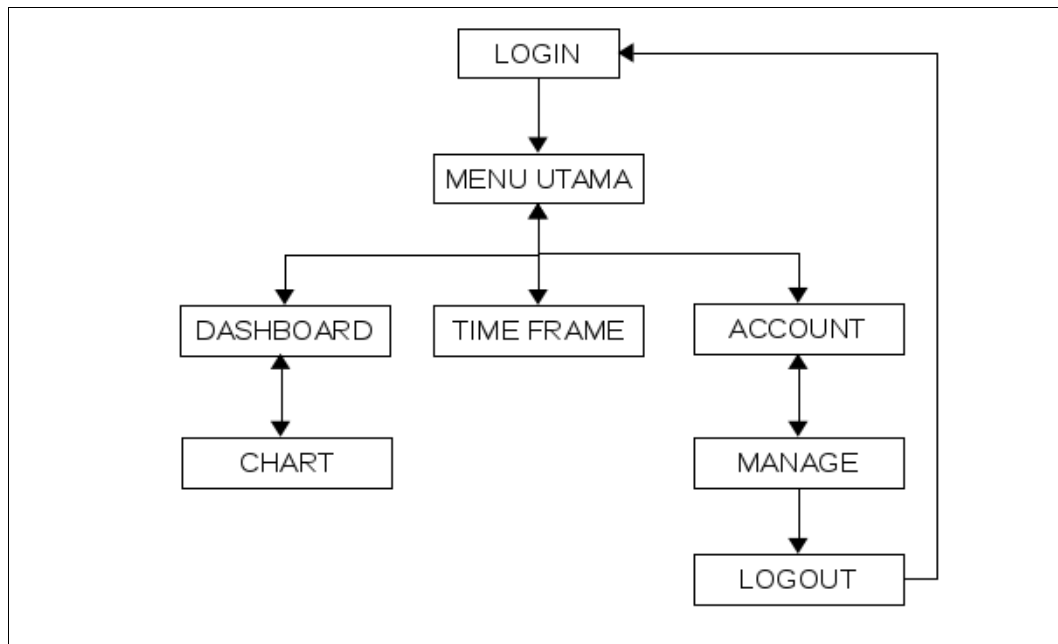
Dalam pembuatan aplikasi ini penulis menggunakan struktur navigasi campuran. Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat digunakan dalam pembuatan website sehingga dapat memberikan ke-interaksian yang lebih tinggi.

Aplikasi ini memiliki dua struktur navigasi yaitu untuk user project manager dan user developer. Perbedaan dari keduanya adalah pada hak akses yang diberikan.



Gambar 2 - Struktur navigasi project manager

Struktur navigasi diatas adalah untuk user yang memiliki role sebagai Project Manager. User tersebut dapat melakukan operasi CRUD (Create, Read, Update, Delete) pada menu Time Frame, Project dan User. Pada menu Account, user dapat melakukan konfigurasi dari accountnya atau dapat melakukan proses log out dari aplikasi. Pada menu dashboard terdapat fitur chart dimana user dapat melihat progress chart dari setiap time frame yang dibuat. Untuk stuktur navigasi pada user yang memiliki role sebagai Developer tampak pada gambar di bawah ini.



Gambar 3 - Struktur navigasi Developer

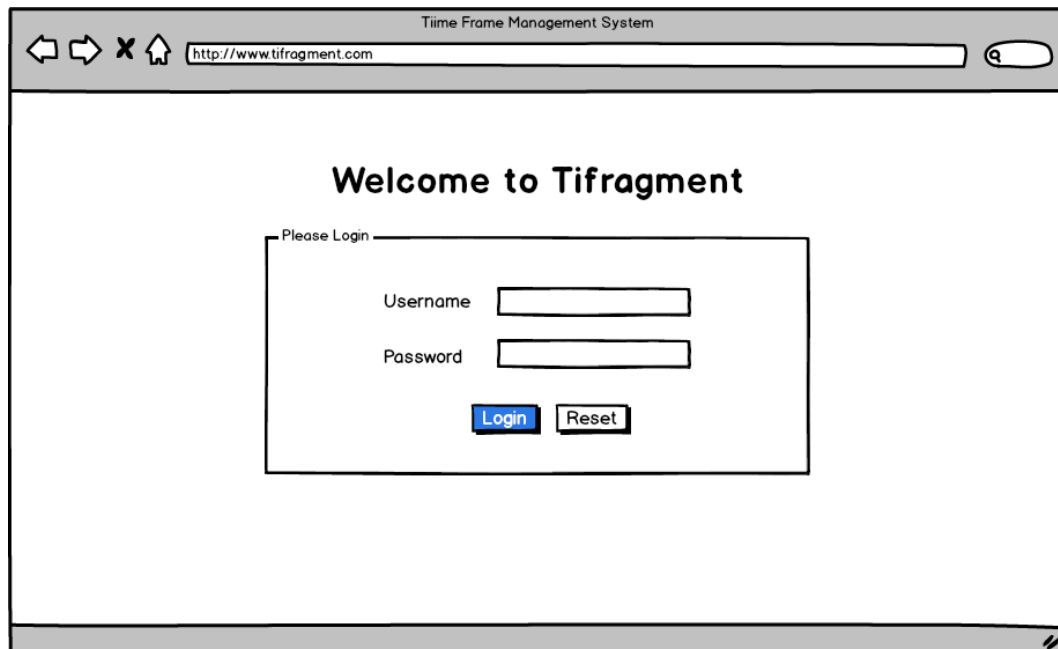
Pada user developer hanya dapat melihat pada menu dashboard yang berisi informasi mengenai time frame pada project yang ia kerjakan serta time frame apa saja yang diikutsertakan oleh user tersebut. Dan seperti pada navigasi user project manager, terdapat juga menu Account untuk melakukan konfigurasi account atau untuk logout dari aplikasi.

3.7 Rancangan User Interface

Salah satu hal terpenting dalam sebuah aplikasi khususnya aplikasi berbasis web adalah User Interface (UI). Dalam hal ini, UI bertindak sebagai sebuah penghubung antara end-user dengan aplikasi. Berikut adalah beberapa rancangan user interface di dalam aplikasi ini.

3.7.1 Tampilan Halaman Login

Menu login digunakan oleh setiap user untuk mengakses halaman utama pada aplikasi. Untuk dapat mengakses, user diharuskan memasukkan username dan password. Untuk user project manager, login dapat menggunakan user default dari aplikasi yang sudah dibuat, sedangkan untuk user developer dapat menggunakan user dan password yang sudah dibuat oleh user project manager.

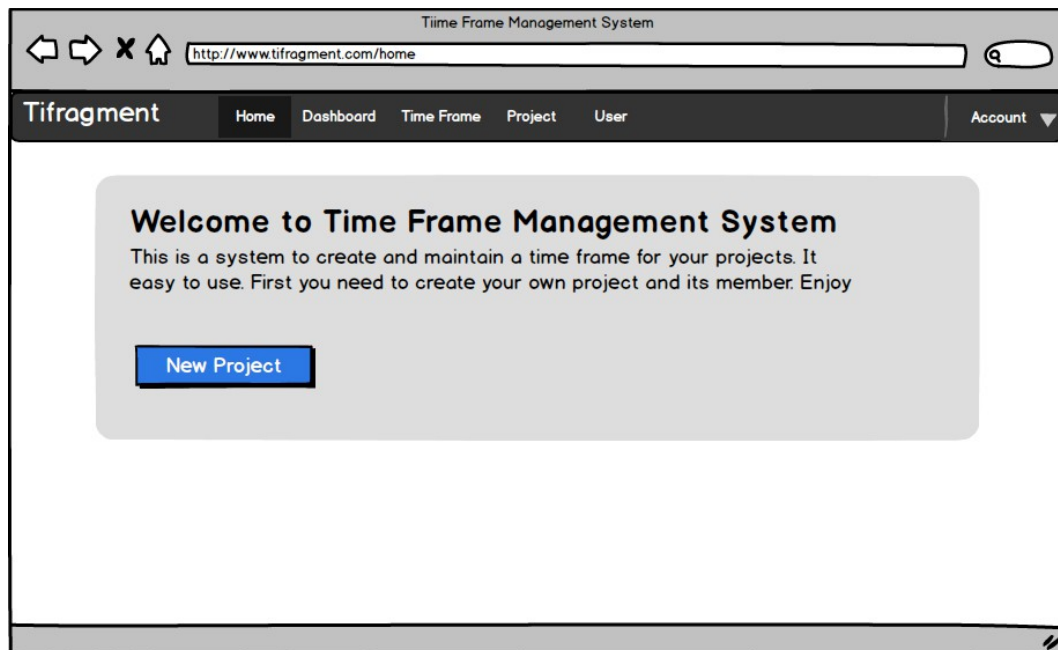


The image shows a web browser window for the 'Time Frame Management System'. The address bar contains the URL 'http://www.tifragment.com'. The main content area displays a 'Welcome to Tifragment' message. Below this is a 'Please Login' section with a form containing two input fields: 'Username' and 'Password'. At the bottom of the form are two buttons: 'Login' and 'Reset'.

Gambar 4 - Tampilan halaman login

3.7.2 Tampilan Halaman Utama User Project Manager

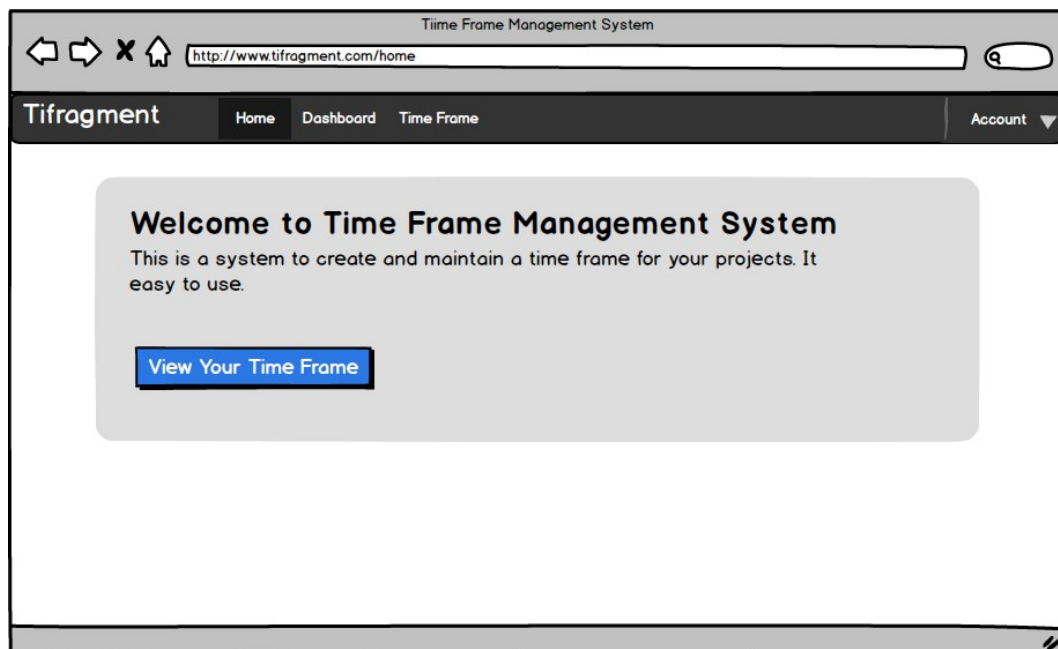
Setelah login, maka user akan dihadapkan pada menu utama. Pada tampilan di bawah ini adalah halaman utama dari user yang memiliki role sebagai project manager. Seperti pada struktur navigasi pada pembahasan sebelumnya, halaman ini mempunyai enam menu yaitu Dashboard, Time Frame, Project, User dan Account.



Gambar 5 - Tampilan halaman utama user Project Manager

3.7.3 Tampilan Halaman Utama User Developer

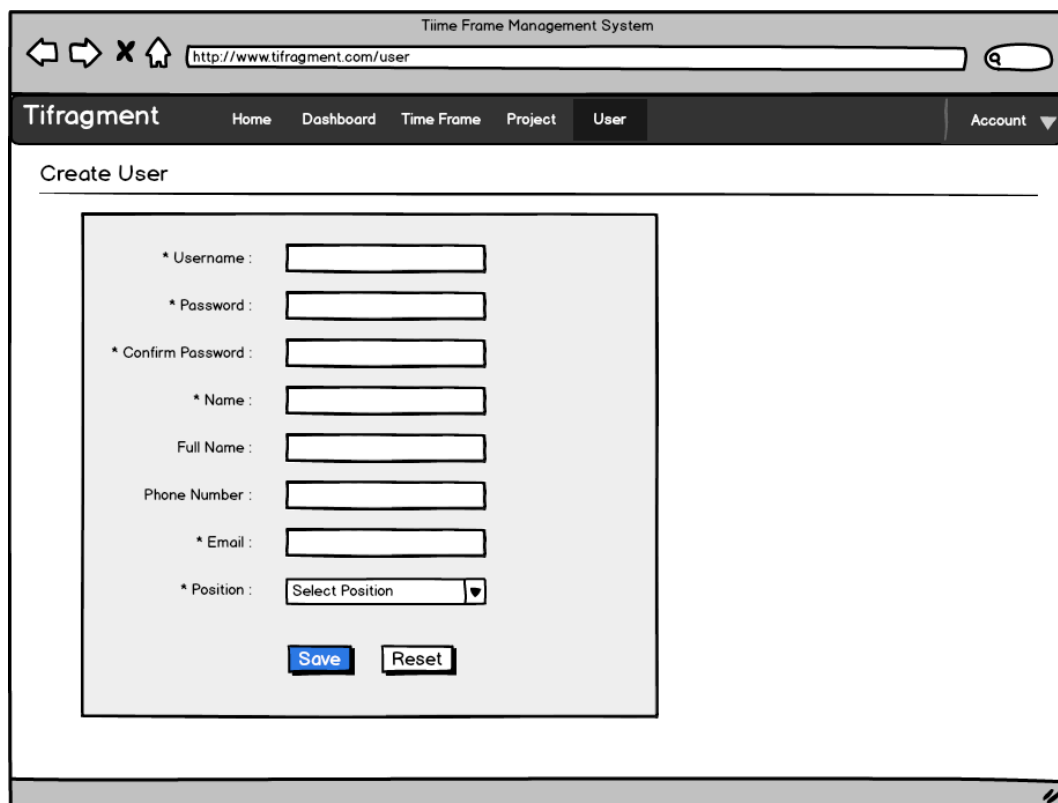
Berbeda dengan halaman utama milik project manager, untuk user yang memiliki role sebagai developer memiliki halaman utama yang lebih sederhana. Seperti pada struktur navigasi, ia hanya dapat melihat project atau time frame yang di tugaskan untuk dirinya sendiri. User dapat melihat melalui menu Dashboard yang terdapat pada panel navigasi.



Gambar 6 - Tampilan halaman utama user Developer

3.7.4 Form Create User

Form ini digunakan untuk membuat user oleh user yang memiliki role project manager. Ketika membuat user, terdapat dua pilihan position yaitu sebagai role **Project Manager** atau sebagai **Developer**.



The screenshot shows a web browser window titled "Time Frame Management System" with the URL "http://www.tifragment.com/user". The navigation bar includes "Tifragment", "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The main heading is "Create User". The form contains the following fields:

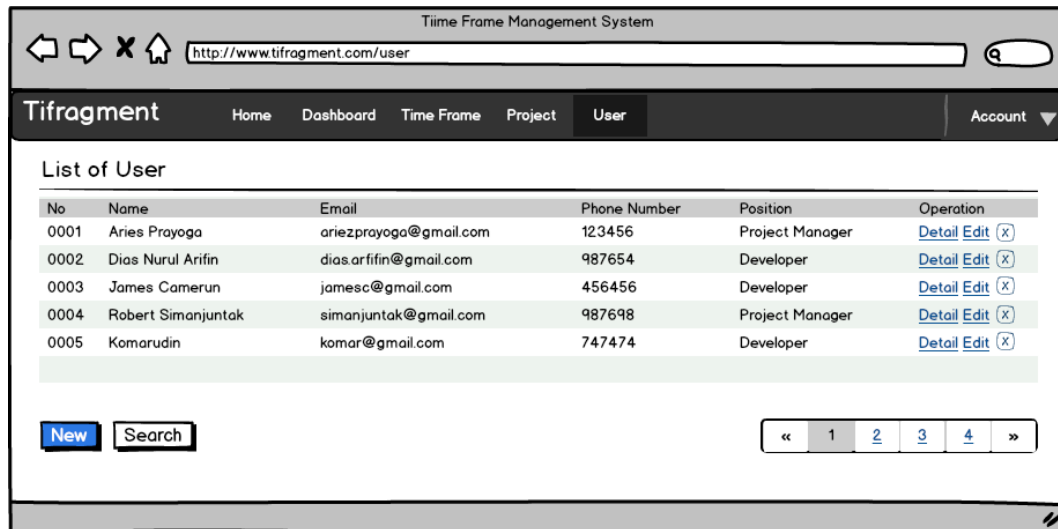
- * Username :
- * Password :
- * Confirm Password :
- * Name :
- Full Name :
- Phone Number :
- * Email :
- * Position :

At the bottom of the form are two buttons: "Save" and "Reset".

Gambar 7 - Form Create User

3.7.5 List User

Halaman ini menampilkan list atau data-data setiap user yang telah dibuat. Pada list tersebut terdapat tombol untuk melakukan operasi Edit dan Delete. Tombol pagination digunakan untuk menampilkan data berikutnya.



Gambar 8 - Halaman List User

3.7.6 Detail User

Halaman ini digunakan untuk melihat rincian dari data user yang sudah di inputkan. Semua informasi user disajikan dalam bentuk sebuah form.

The screenshot shows a web browser window titled "Tiime Frame Management System". The address bar contains "http://www.tifragment.com/user". The navigation menu includes "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The "User" menu item is selected. The page title is "Detail User". The user details are displayed in a form with the following fields:

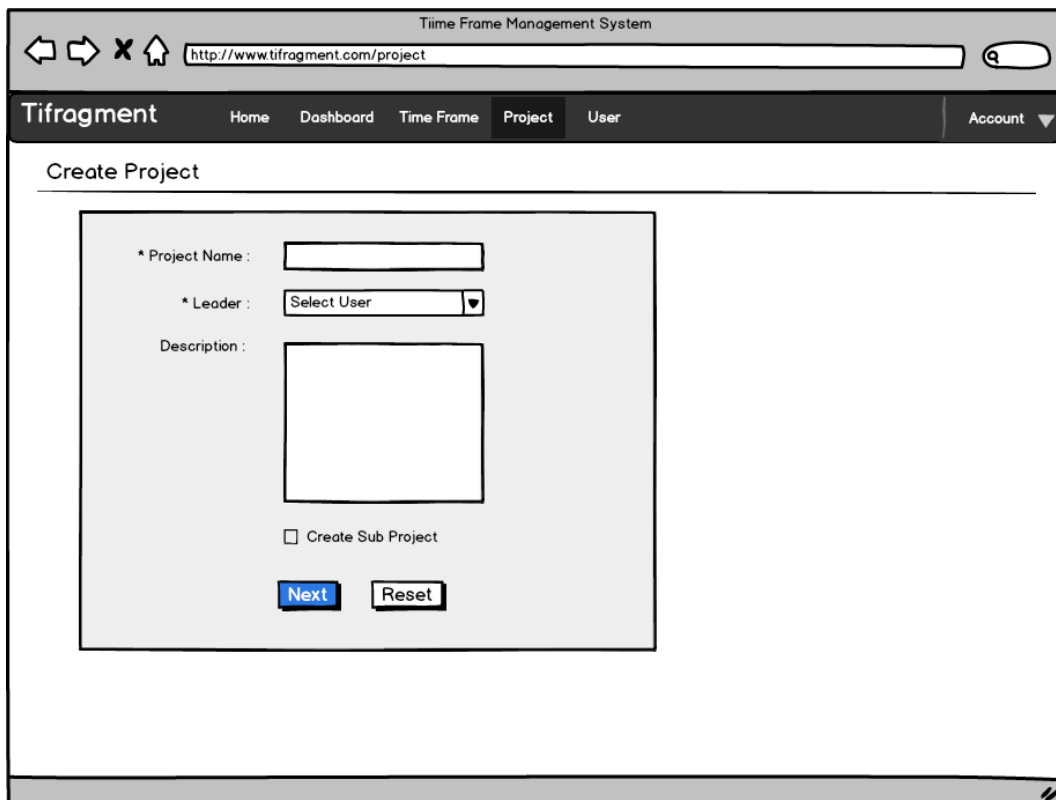
Username :	ariesp
Name :	Aries
Full Name :	Aries Prayoga
Phone Number :	123456
Email :	ariesprayoga@gmail.com
Position :	Project Manager

A "Back" button is located at the bottom right of the form.

Gambar 9 - Detail User

3.7.7 Create Project

Form ini digunakan untuk membuat sebuah project. Jika kolom checkbox *create subproject* dicentang, maka akan muncul form baru untuk membuat Sub Project. Field yang harus dimasukkan adalah project name dan leader dari list user yang memiliki role sebagai project manager.



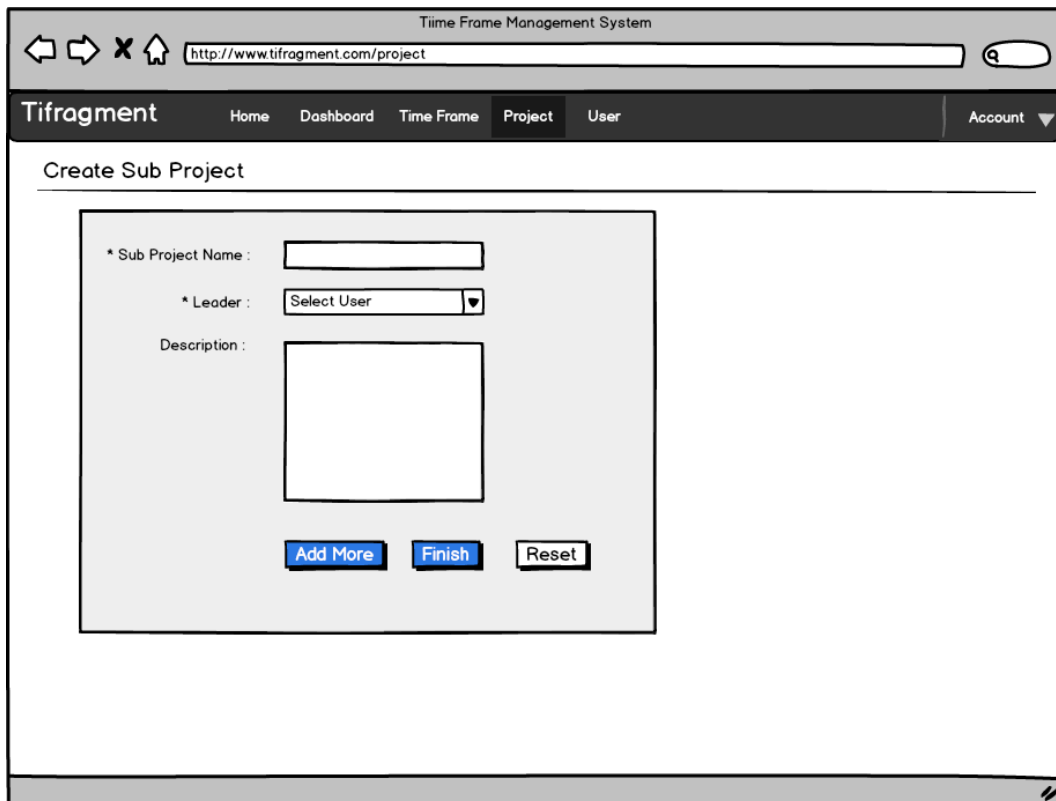
The screenshot shows a web browser window titled "Time Frame Management System" with the URL "http://www.tifragment.com/project". The navigation bar includes "Tifragment", "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The main heading is "Create Project". The form contains the following elements:

- * Project Name :
- * Leader :
- Description :
- ☐ Create Sub Project
-

Gambar 10 - Form Create Project

3.7.8 Create Sub Project

Form ini digunakan untuk membuat subproject dari parent project pada sebelumnya. Form ini hanya muncul jika kolom checkbox pada form sebelumnya dicentang.



The screenshot shows a web browser window titled "Time Frame Management System" with the URL "http://www.tifragment.com/project". The navigation bar includes "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The main heading is "Create Sub Project". The form contains the following fields and buttons:

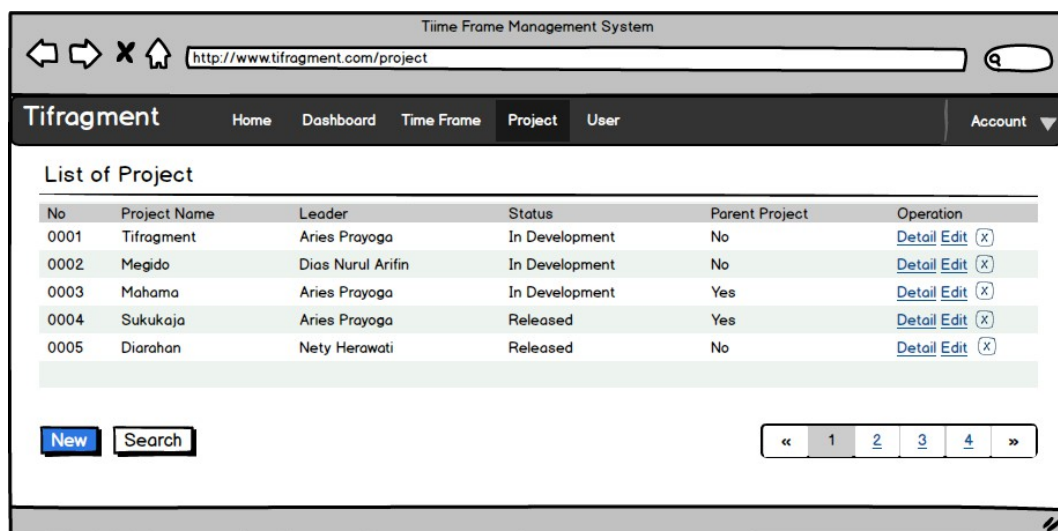
- * Sub Project Name :
- * Leader :
- Description :
- Buttons: **Add More**, **Finish**, and **Reset**

Gambar 11 - Form Create Sub Project

Terdapat dua tombol submit dan satu tombol reset pada form tersebut. Jika tombol **Add More** ditekan maka akan muncul form sama yang berfungsi untuk menginputkan data subproject baru. Jika tombol **Finish** ditekan maka akan diarahkan pada halaman baru list project.

3.7.9 List Project

Halaman ini menampilkan list dari project-project yang telah dibuat. Pada halaman ini ditampilkan juga status dari project yaitu **In Development** yang berarti project tersebut sedang dalam tahap pengembangan, dan status **Released** yang berarti project sudah selesai dirilis dan **Not Released** jika project tersebut belum pernah dibuat time framenya.



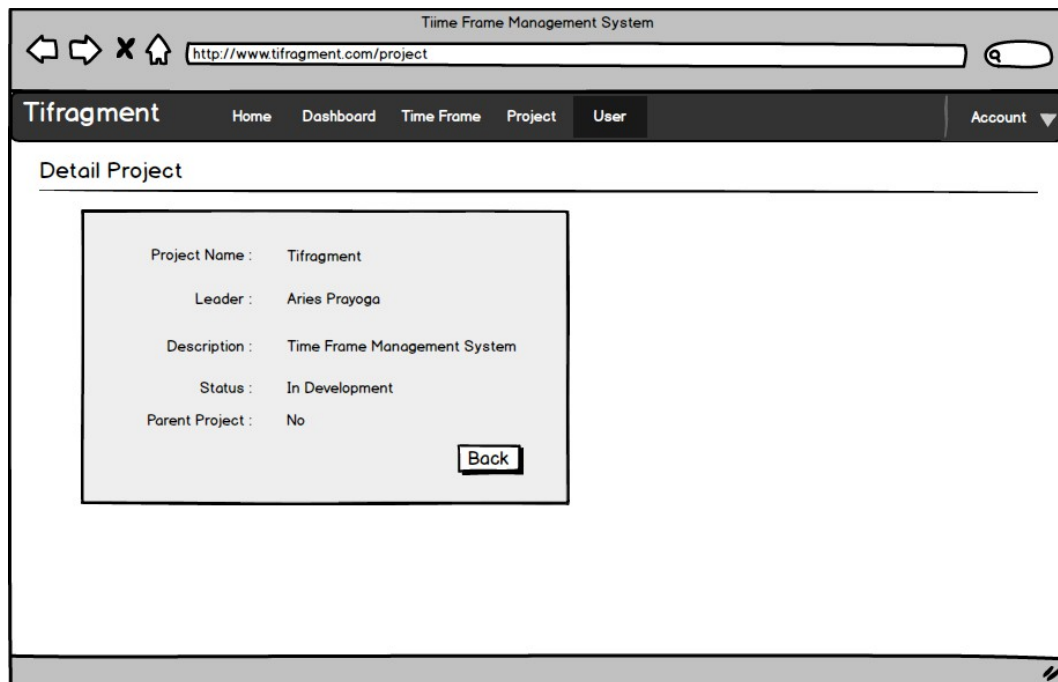
No	Project Name	Leader	Status	Parent Project	Operation
0001	Tifragment	Aries Prayoga	In Development	No	Detail Edit X
0002	Megido	Dias Nurul Arifin	In Development	No	Detail Edit X
0003	Mahama	Aries Prayoga	In Development	Yes	Detail Edit X
0004	Sukukaja	Aries Prayoga	Released	Yes	Detail Edit X
0005	Diarahan	Nety Herawati	Released	No	Detail Edit X

Gambar 12 - List Project

Pada kolom Parent Project, menandakan bahwa apakah suatu project tersebut merupakan sebuah project yang memiliki subproject atau tidak. Untuk dapat melihat detail suatu project beserta detail dari subproject, kita dapat menekan tombol **Detail** pada kolom **Operation**.

3.7.10 Detail Project

Halaman ini menampilkan form detail dari data Project yang tidak memiliki Sub Project.



The screenshot shows a web browser window with the address bar displaying `http://www.tifragment.com/project`. The page title is "Time Frame Management System". The navigation menu includes "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The main content area is titled "Detail Project" and contains a form with the following details:

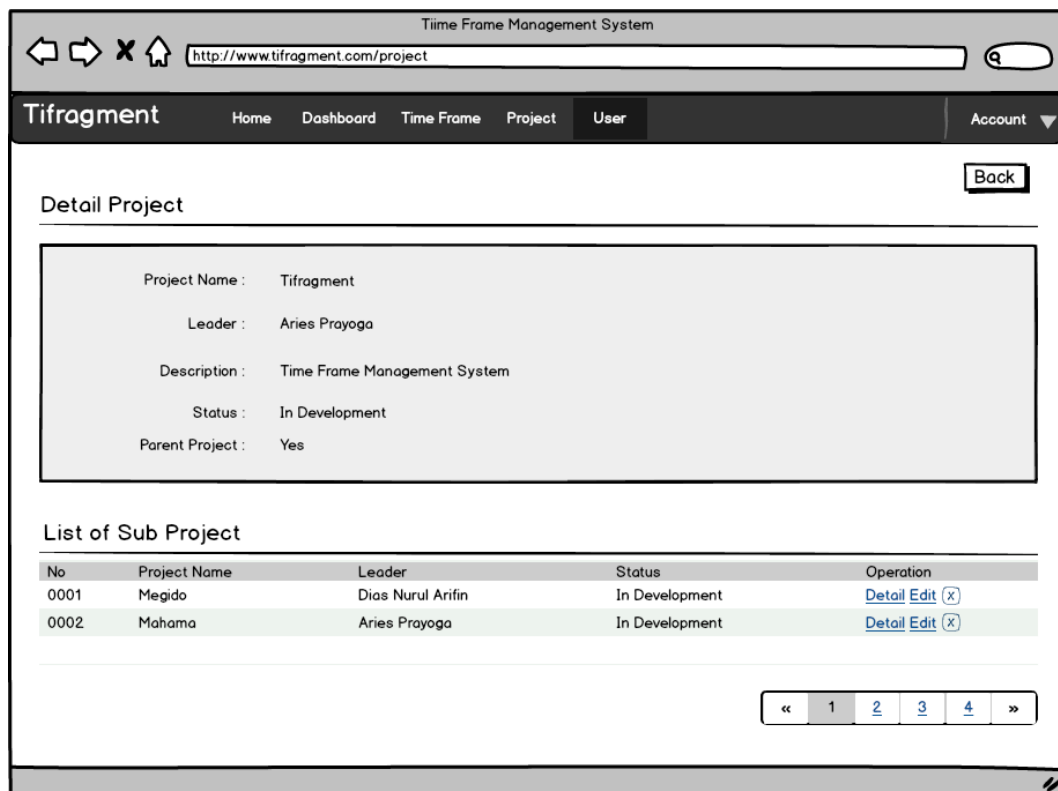
Project Name :	Tifragment
Leader :	Aries Prayoga
Description :	Time Frame Management System
Status :	In Development
Parent Project :	No

A "Back" button is located at the bottom right of the form.

Gambar 13 - Detail Project

3.7.11 Detail Project dan Sub Project

Halaman ini menampilkan form detail dari data Project yang memiliki Sub Project. Data List dari Sub Project yang dimiliki ditampilkan di baris **List of Sub Project**.



The screenshot shows a web browser window with the URL <http://www.tifragment.com/project>. The page title is "Time Frame Management System". The navigation bar includes links for Home, Dashboard, Time Frame, Project, User, and Account. The main content area is titled "Detail Project" and contains a "Back" button. Below the title, there is a form displaying project details for "Tifragment".

Project Name :	Tifragment
Leader :	Aries Prayoga
Description :	Time Frame Management System
Status :	In Development
Parent Project :	Yes

Below the project details, there is a section titled "List of Sub Project" containing a table with two rows of sub-projects.

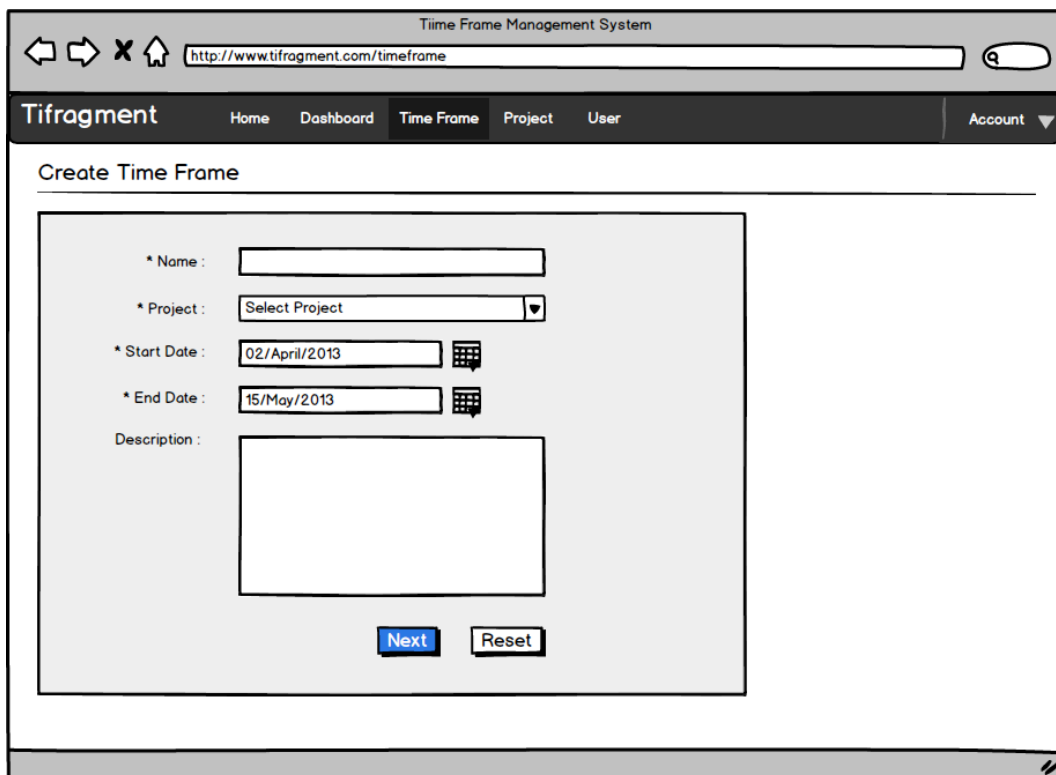
No	Project Name	Leader	Status	Operation
0001	Megido	Dias Nurul Arifin	In Development	Detail Edit (X)
0002	Mahama	Aries Prayoga	In Development	Detail Edit (X)

At the bottom right of the table, there is a pagination control showing "« 1 2 3 4 »".

Gambar 14 - List Sub Project

3.7.12 Create Time Frame

Form ini digunakan untuk membuat sebuah time frame. Untuk dapat membuat time frame, user perlu memasukkan field Nama dari time frame tersebut, memilih list project mana dari data Project, Start Date dan End Date yang digunakan untuk menandakan kapan mulai dan berakhirnya tahap pengembangan dari suatu project tersebut.



The screenshot shows a web browser window titled "Time Frame Management System" with the URL "http://www.tifragment.com/timeframe". The browser's address bar and navigation buttons are visible. Below the browser window is a navigation bar for "Tifragment" with links to "Home", "Dashboard", "Time Frame", "Project", "User", and an "Account" dropdown menu. The main content area is titled "Create Time Frame" and contains a form with the following fields:

- * Name :
- * Project : (dropdown menu)
- * Start Date : (calendar icon)
- * End Date : (calendar icon)
- Description :

At the bottom of the form are two buttons: "Next" (blue) and "Reset" (white).

Gambar 15 - Create Time Frame

Jika form telah diisi semua, maka untuk melanjutkan user harus menekan tombol **Next** yang berfungsi untuk melanjutkan pengisian pada Detail Time Frame seperti pada gambar di bawah ini.

The screenshot shows a web browser window titled "Time Frame Management System" with the URL "http://www.tifragment.com/timeframe". The navigation bar includes "Tifragment", "Home", "Dashboard", "Time Frame", "Project", "User", and "Account". The main heading is "Create Time Frame". The form contains the following fields and buttons:

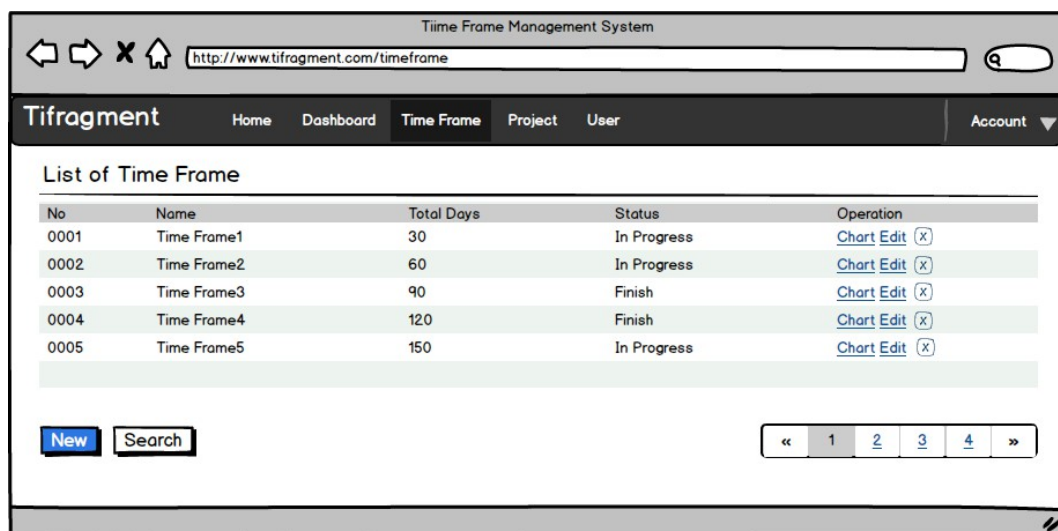
- * Task Name :
- * Developer :
- * Start Day :
- * End Day :
- Description :
-

Gambar 16 - Create Task

Form diatas adalah untuk membuat sebuah task pada time frame. Sebuah time frame terdiri dari satu atau banyak task. Di form tersebut user harus memasukkan Task Name, Developer yaitu list user yang memiliki role sebagai developer, Start Day yaitu waktu mulai pengerjaan dan End Day yaitu waktu akhir pengerjaan dari task tersebut. Tombol **Add More** digunakan untuk menambah Task Baru, dan tombol **Finish** untuk mengakhiri pembuatan time frame.

3.7.13 List Time Frame

Halaman ini memuat semua list data time frame yang telah dibuat. Seorang project manager dapat melihat nama time frame, total days yang diperlukan, dan status dari time frame tersebut apakah **In Progress** atau **Finish**. Kita dapat melakukan operasi **Edit** dan **Delete** pada kolom **Operation**, serta kita dapat melihat detail dan grafik chart dari time frame pada link **Chart**.



The screenshot shows a web application titled "Time Frame Management System". The browser address bar displays "http://www.tifragment.com/timeframe". The application has a navigation menu with "Home", "Dashboard", "Time Frame", "Project", and "User". The "Time Frame" section is active, showing a "List of Time Frame" table. The table has five columns: "No", "Name", "Total Days", "Status", and "Operation". It lists five time frames with their respective IDs, names, durations, and statuses. Each row includes links for "Chart" and "Edit", along with a delete icon (X). Below the table, there are "New" and "Search" buttons, and a pagination control showing page 1 of 4.

No	Name	Total Days	Status	Operation
0001	Time Frame1	30	In Progress	Chart Edit (X)
0002	Time Frame2	60	In Progress	Chart Edit (X)
0003	Time Frame3	90	Finish	Chart Edit (X)
0004	Time Frame4	120	Finish	Chart Edit (X)
0005	Time Frame5	150	In Progress	Chart Edit (X)

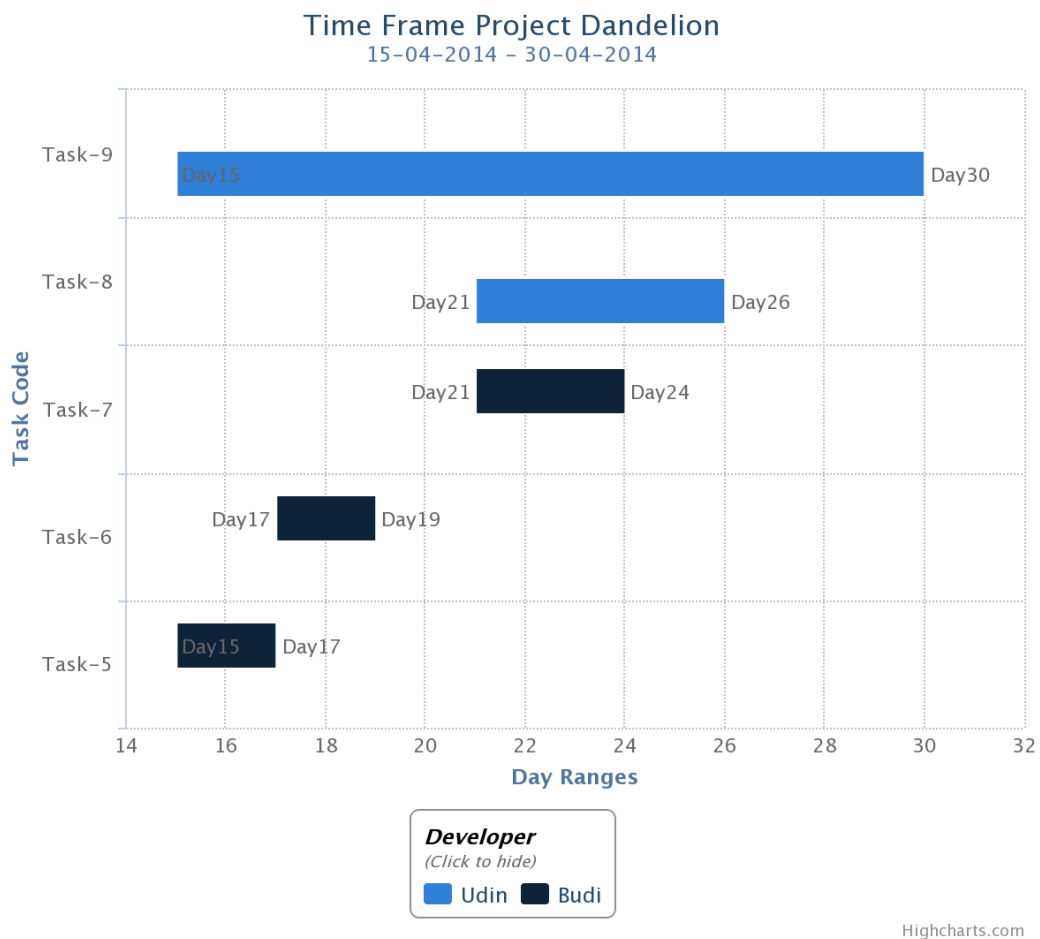
New Search

« 1 2 3 4 »

Gambar 17 - List Time Frame

3.7.14 Chart

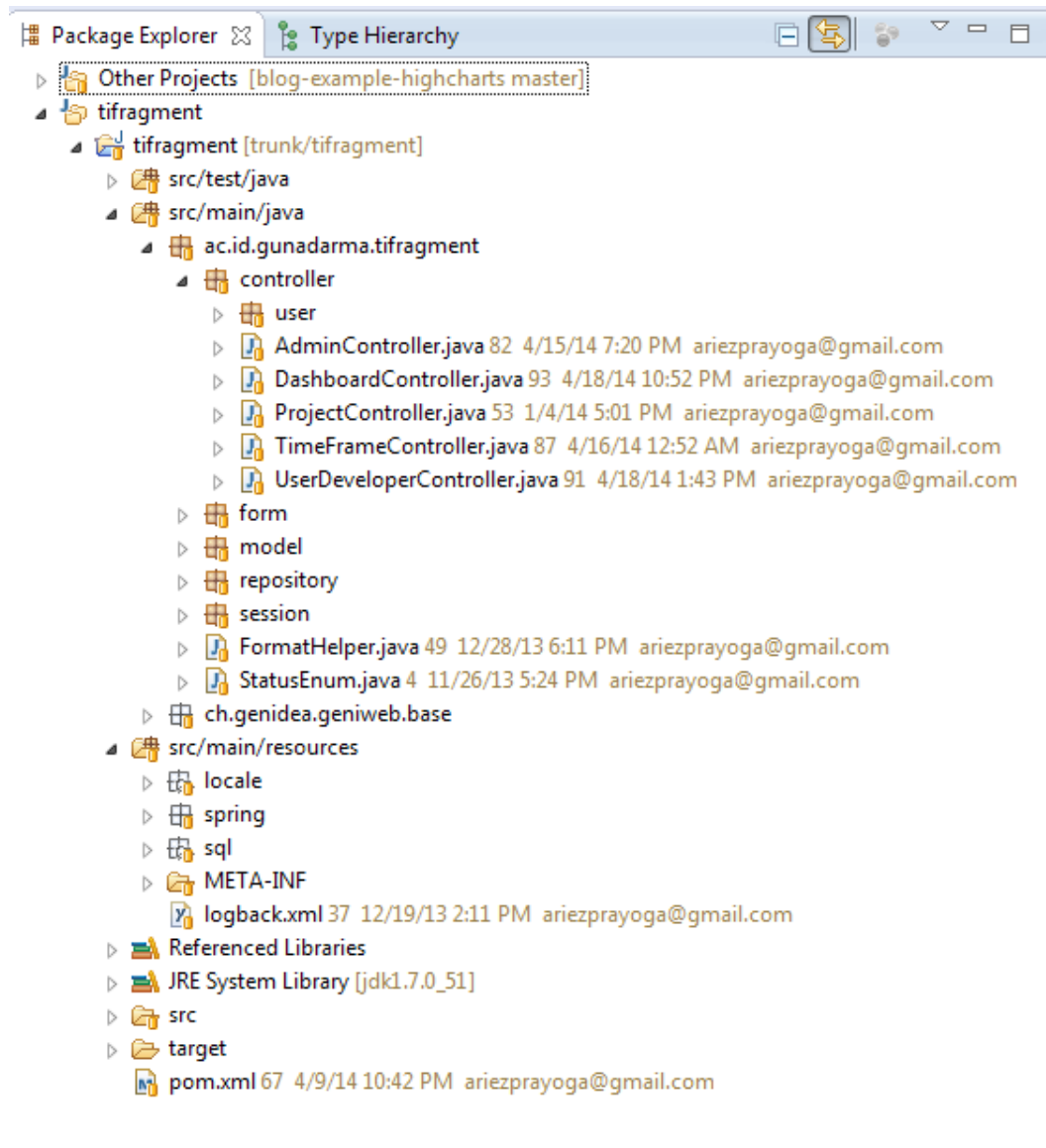
Setiap time frame yang dibuat akan menghasilkan sebuah chart yang merupakan summary dari task-task yang telah dibuat. Di dalam chart ini terdapat user-user yang terkait dengan pengerjaan project tersebut dan day range (rentang hari) yang diperlukan. Chart menggunakan tipe stacked bar dimana aksis Y menunjukkan task-task yang dikerjakan oleh user dan aksis X menunjukkan berapa total man-days yang diperlukan.



Gambar 18 - Tampilan Chart

3.8 Proses Development

Langkah pertama yang dilakukan adalah membuat project dengan menggunakan Eclipse IDE dengan tipe Maven Project dimana susunannya seperti pada gambar berikut.



Gambar 19 - Struktur Project

Karena menggunakan basis MVC, maka pembuatan kelas-kelas dan viewer juga mengikuti struktur tersebut. Berikut adalah beberapa pattern dari masing-masing object kelas dan view yang digunakan :

- Entity Class → Kelas entity object yang digunakan untuk mapping dari table DBMS ke dalam sebuah object atau sebaliknya. Merupakan bentuk implementasi dari ORM
- Service → Kelas yang terdiri dari interface dan implementasinya, digunakan sebagai repository service yang menangani logical request seperti CREATE, SELECT, UPDATE, DELETE terhadap database
- Controller Class → Kelas controller yang menangani routing dan request response dari user ke server dan sebaliknya. Semua URL path di mapping di kelas ini
- View → File dengan ekstensi html yang didalamnya terdapat bahasa markup HTML yang mengolah tampilan dan bahasa scripting parser yang berguna untuk mengatur response data dari server.

Berikut adalah satu contoh kode-kode dari pembatan module Project dimulai dari entity hingga view.

3.8.1 Entity Class

```
@Entity
@Table(name = "tifragment_project")
public class Project implements Serializable {

    private static final long serialVersionUID = 8894985001552485023L;

    @Id
    @GeneratedValue(generator = "system-uuid")
    @GenericGenerator(name = "system-uuid", strategy = "uuid")
    @Column(name = "uuid", unique = true)
    private String uuid;

    @Column(name = "name", nullable = false)
    private String projectName;

    @Column(name = "description", length = 1000)
    private String description;

    @Column(name = "status")
    private StatusEnum status;

    @OneToMany(mappedBy = "project", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private List<Project> subprojects;

    @ManyToOne
    @JoinColumn(name = "fk_project")
    private Project project;

    @ManyToOne
    @JoinColumn(name = "fk_user")
    private User user;

    @OneToOne
    @JoinColumn(name = "fk_timeFrame")
    private TimeFrame timeFrame;
```

Kelas tersebut adalah sebuah entity untuk module project dimana akan diparse oleh Hibernate menjadi sebuah table. Terdapat annotations object yang menandakan sebagai property table dalam database yaitu :

- @Entity → menandakan bahwa kelas entity object
- @Table → kelas akan di parse sebagai table baru dengan nama “tifragment_project”
- @Id, @GeneratedValue, @GenericGenerator → menandakan bahwa property kelas digunakan sebagai primary key dengan nilai berbentuk uuid

- `@Column` → menandakan property kelas sebagai sebuah kolom dalam table database dengan beberapa parameter seperti not null, name, length
- `@ManyToOne` → menandakan bahwa property pada kelas memiliki hubungan relasi Many-To-One terhadap kelas entity lain
- `@JoinColumn` → menandakan bahwa property kelas melakukan join terhadap kelas entity lain dengan foreign key
- `@OneToOne` → menandakan bahwa property pada kelas memiliki hubungan relasi One-To-One terhadap kelas entity lain

3.8.2 Service

Ada dua kelas service yaitu sebuah interface dan kelas implementasinya

```
package ac.id.gunadarma.tifragment.repository;

import java.util.List;

import ch.genidea.geniweb.base.domain.User;
import ch.genidea.geniweb.base.utility.ListWrapper;
import ac.id.gunadarma.tifragment.model.Project;

public interface ProjectRepository {

    void save(Project project);

    void update(Project project);

    Project findProjectById(String projectId);

    ListWrapper<Project> findAll(int max, int page);

    List<User> findProjectManagers();

    long totalRow();
}
```

Terdapat beberapa method dalam interface tersebut yang berfungsi untuk layer database seperti save, update select by, count.

```

@Repository
public class ProjectRepositoryImpl implements ProjectRepository {

    @PersistenceContext
    private EntityManager em;

    @Override
    @Transactional(rollbackFor = Exception.class)
    public void save(Project project) {
        em.persist(project);
        em.flush();
    }

    @Override
    @Transactional(rollbackFor = Exception.class)
    public void update(Project project) {
        em.merge(project);
        em.flush();
    }

    @Override
    @Transactional(readOnly = true)
    public Project findProjectById(String projectId) {
        Project project = em.find(Project.class, projectId);
        return project;
    }
}

```

Kelas di atas adalah kelas implementasi dari interface sebelumnya dimana kelas tersebut meng-override semua method-method yang ada. Berikut adalah penjelasan dari annotations yang digunakan :

- `@Repository` → annotation dari Spring, menandakan bahwa kelas menangani persistence layer
- `@PersistenceContext` → annotation dari JPA, menandakan bahwa kelas EntityManager adalah kelas yang di inject dan berupa kelas persistence yang menangani operasi pada layer database
- `@Transactional` → annotation dari Spring, menandakan bahwa method yang ditandai adalah method yang melakukan transaksi ke dalam layer database seperti SELECT, CREATE, UPDATE atau DELETE

3.8.3 Controller Class

```
package ac.id.gunadarma.tifragment.controller;

import static ac.id.gunadarma.tifragment.session.SessionAuthentication.getAuthenticationContext;

@Controller
public class ProjectController {

    @Autowired
    private ProjectRepository projectRepository;

    @Autowired
    private UserRepository userRepository;

    @RequestMapping(value = "/base/page/admin/project/create", method = RequestMethod.GET)
    public String create(Model model) {
        model.addAttribute(MENU_ACTIVE, PROJECT);
        model.addAttribute("project", new ProjectForm());
        model.addAttribute("users", getProjectLeaders());
        return "base/page/admin/project/create";
    }

    @RequestMapping(value = "/base/page/admin/project/createsub{id}", method = RequestMethod.GET)
    public String createSub(Model model, HttpServletRequest request) {
        ProjectForm form = new ProjectForm();
        form.setParentUuid(request.getParameter("projectId"));
        model.addAttribute(MENU_ACTIVE, PROJECT);
        model.addAttribute("project", form);
        model.addAttribute("users", getProjectLeaders());
        return "base/page/admin/project/createsub";
    }

    @RequestMapping(value = "/base/page/admin/project/detail{id}", method = RequestMethod.GET)
    public ModelAndView view(Model model, HttpServletRequest request) {
        Project project = projectRepository.findById(request.getParameter("projectId"));
        model.addAttribute(MENU_ACTIVE, PROJECT);
        return new ModelAndView("base/page/admin/project/detail", "project", project);
    }
}
```

Potongan kode di atas adalah contoh kelas Controller yang berfungsi sebagai routing terhadap request dan response. Spring framework menandai sebuah kelas controller dengan annotation `@Controller`. Berikut penjelasannya

- `@Controller` → kelas ditandai sebagai sebuah komponen yaitu controller yang berfungsi sebagai handler request dan response yang datang dari user sebagai router
- `@Autowired` → menandakan bahwa interface `ProjectRepository` adalah Injected Class, artinya kita bisa menggunakan attribute-attribute kelas yang diinject tanpa harus meng-instance object kelas tersebut.
- `@RequestMapping` → menandakan bahwa method yang ditandai annotation ini merupakan sebuah method yang menangani request dan

memappingkannya ke dalam path URL. Contoh pada request mapping dengan value `“/base/page/admin/project/create”` dan dengan HTTP method GET. Artinya, method tersebut menangani request dengan HTTP method GET dan pada URL `“/base/page/admin/project/create”` dengan return value berupa sebuah template yang terletak pada directory `“base/page/admin/project/create”`.

3.8.4 View

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head th:include="base/page/header :: headerTiFragment" />

<body>
<div th:include="base/page/menu-admin :: menuTiFragment"></div>
<div class="container">
<form id="form" th:action="@{/base/page/admin/project/createproject}" th:object="${project}" role="form" class="form-horizontal" method="post">
<fieldset>
<legend class="text-info">Create Project</legend>
<div class="form-group error">
<label for="name" class="col-lg-2 control-label">*Project Name</label>
<div class="col-lg-3">
<input id="name" th:field="*{name}" placeholder="project name" class="form-control" type="text" required="required"/>
<span th:if="${#fields.hasErrors('name')}" th:errors="*{name}" class="label label-danger"></span>
</div>
</div>
<div class="form-group error">
<label for="name" class="col-lg-2 control-label">*Leader</label>
<div class="col-lg-3">
<select class="form-control" th:field="*{userId}" required="required">
<option value="" selected="selected">--Select Leader--</option>
<option th:each="user : ${users}" th:value="${user.userId}" th:text="${user.fullName} == '' ? ${user.name} : ${user.fullName}"/>
</select>
<span th:if="${#fields.hasErrors('userId')}" th:errors="*{userId}" class="label label-danger"></span>
</div>
</div>
<div class="form-group error">
<label for="name" class="col-lg-2 control-label">Description</label>
<div class="col-lg-3">
<textarea id="description" rows="10" cols="100" th:field="*{description}" placeholder="description" class="form-control"></textarea>
</div>
</div>
</div>

```

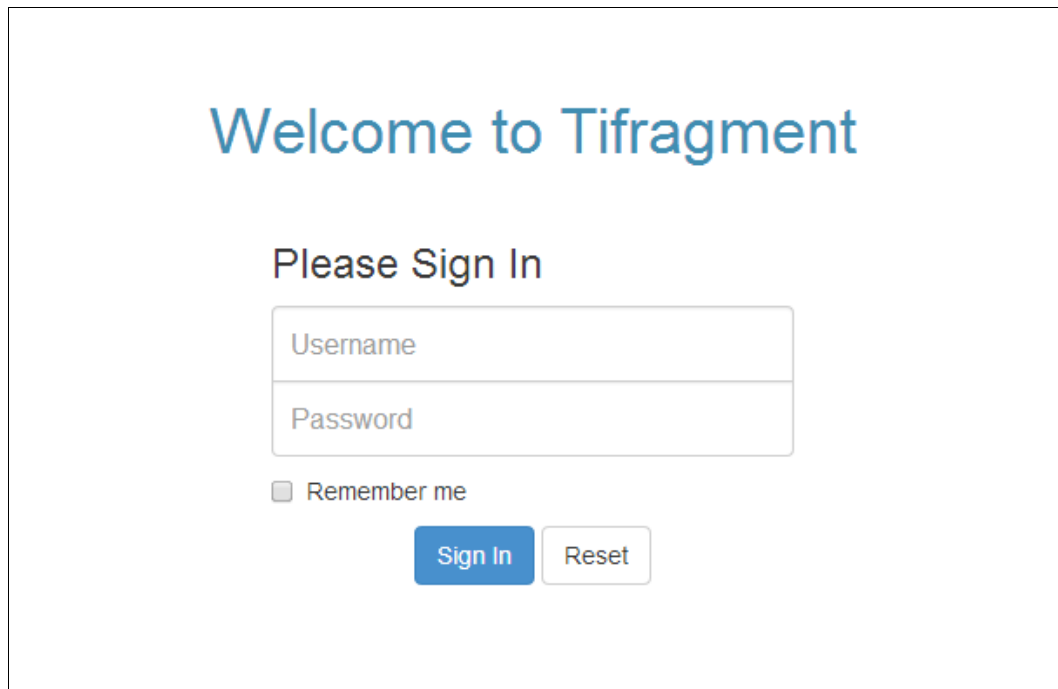
View merupakan file HTML dengan disisipkan Expression Language / EL menggunakan java library Thymeleaf. Pada contoh potongan kode diatas terdapat bentuk tag `xmlns:th="http://www.thymeleaf.org"` yang artinya setiap tag html yang ditandai akan diparse oleh library tersebut berupa EL. Berikut penjelasannya

- `th:include="base/page/header :: headerTiFragment"` → template menyisipkan file dari luar dengan nama header yang berada pada direcotory `“base/page/header”`

- `th:action="@{/base/page/admin/project/createproject}"` → action dari form tersebut akan berada pada host *"hostdomain/base/page/admin/project/createproject"*
- `th:object="${project}"` → object yang dikirim oleh form ditampung oleh object dengan name *project*. Didefinisikan oleh method pada kelas controller yang menghandle request form ini.
- `th:field="*{name}"` → property yang akan dikirimkan ke server dari object *project*

3.9 Hasil Rancangan

3.9.1 Halaman Login



The image shows a login page design for 'Tifragment'. It features a light gray background with a white rectangular area containing the login form. The title 'Welcome to Tifragment' is displayed in a large, blue, sans-serif font. Below it, the text 'Please Sign In' is in a smaller, dark gray font. The login form consists of two stacked input fields: 'Username' and 'Password', both with light gray borders and placeholder text. Below the password field is a checkbox labeled 'Remember me'. At the bottom of the form are two buttons: a blue 'Sign In' button and a white 'Reset' button with a gray border.

Welcome to Tifragment

Please Sign In

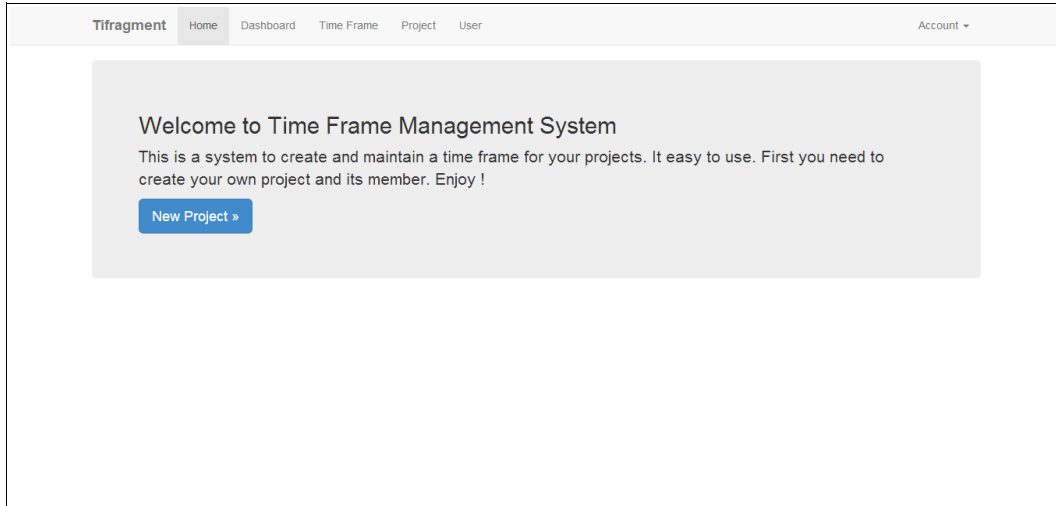
Username

Password

☐ Remember me

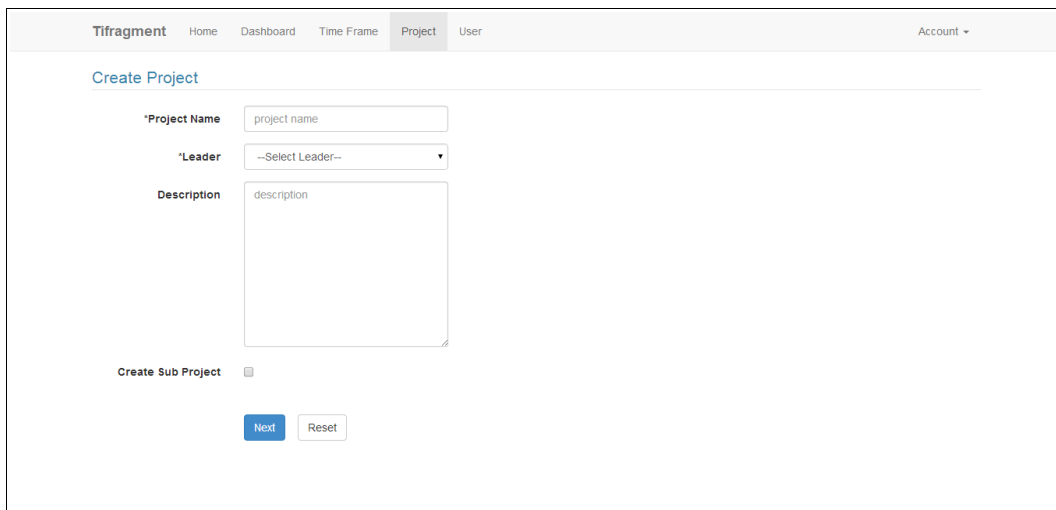
Sign In Reset

3.9.2 Halaman Utama user Project Manager



The screenshot shows the 'Home' page of the Tifragment application for a Project Manager user. The top navigation bar includes 'Tifragment', 'Home', 'Dashboard', 'Time Frame', 'Project', 'User', and an 'Account' dropdown. The main content area features a light gray box with the text: 'Welcome to Time Frame Management System', 'This is a system to create and maintain a time frame for your projects. It easy to use. First you need to create your own project and its member. Enjoy !', and a blue button labeled 'New Project »'.

3.9.3 Create Project



The screenshot displays the 'Create Project' form. The top navigation bar is identical to the previous page, with 'Project' highlighted. Below the navigation bar, the title 'Create Project' is followed by a horizontal line. The form contains three main input fields: '*Project Name' with a text input containing 'project name', '*Leader' with a dropdown menu showing '--Select Leader--', and 'Description' with a large text area containing 'description'. At the bottom left, there is a checkbox labeled 'Create Sub Project' which is currently unchecked. At the bottom center, there are two buttons: a blue 'Next' button and a white 'Reset' button.

3.9.4 List Project

Tifragment

HomeDashboardTime FrameProjectUser

Account

List of Project

No	Project Name	Leader	Status	Parent Project	Operation
1	ProjectIGHI	Aries Prayoga	Not Release	No	✕ ✕
2	ProjectDEF	Aries Prayoga	Not Release	No	✕ ✕
3	Tifragment	Aries Prayoga	In Developement	No	✕ ✕
4	ProjectAB	Aries Prayoga	Not Release	Yes	✕ ✕
5	ProjectIE	Aries Prayoga	Not Release	Yes	✕ ✕

New

12»

Found 9 Row(s)

3.9.5 Detail Project

Tifragment

HomeDashboardTime FrameProjectUser

Account

Detail Project

Project Name

Dandelion

*Leader

Aries Prayoga

Description

Virtual Xchange

Back

3.9.6 Create User

Tifragment

HomeDashboardTime FrameProjectUser

Account ▾

Create User

*Username

user name

*Password

password

*Confirm Password

confirm password

*Name

name

Full Name

full name

Phone Number

ex: 08123456789

*Email

user@yourdomain

*Position

--Select Position--

SaveReset

3.9.7 List User

Tifragment

HomeDashboardTime FrameProjectUser

Account ▾

List of User

No	Name	Full Name	Email	Phone Number	Position	Operation
1	Bejo		bejo@gmail.com	08765432190	Developer	✕ ✕ ✕ ✕
2	Ucok	Ucok Djarahardja	ucok@gmail.com	08123456789	Developer	✕ ✕ ✕ ✕
3	Udin	Udin Mangkunegoro	udin@gmail.com	08123456789	Developer	✕ ✕ ✕ ✕
4	Budi	Budi Sunamo	budi@gmail.com	08123456789	Developer	✕ ✕ ✕ ✕

New

Found 4 Row(s)

3.9.8 Detail User

Tifragment

HomeDashboardTime FrameProjectUser

Account ▾

Detail User

Username

budi

Name

Budi

Full Name

Budi Sunamo

Phone Number

08123456789

Email

budi@gmail.com

Position

Developer

Status

Active

Back

3.9.10 Change Password User

Tifragment

HomeDashboardTime FrameProjectUser

Account ▾

Change Password

Username

budi

*Password

password

*Confirm Password

confirm password

Save

Reset

3.9.11 Create Time Frame

Tifragment

Home

Dashboard

Time Frame

Project

User

Account ▾

Create Time Frame

*Name

time frame name

*Project

--Select Project-- ▾

*Start Date

20-04-2014

*End Date

20-04-2014

Description

description

Next

Reset

3.9.12 Create Task

Tifragment

Home

Dashboard

Time Frame

Project

User

Account ▾

Create Task

*Task Name

time task name

*Developer

--Select Developer-- ▾

*Start Date

01-05-2014

*End Date

31-05-2014

Description

description

Add More Task

▢

Save

Reset

3.9.13 List Time Frame

Tifragment

Home

Dashboard

Time Frame

Project

User

Account

List of Time Frame

No	Name	Total Days	Status	Operation
1	HyperStone	30	In Progress	<div><div></div><div></div><div></div><div></div><div></div></div>
2	Gondangdia	3	In Progress	<div><div></div><div></div><div></div><div></div><div></div></div>
3	Legolas	15	In Progress	<div><div></div><div></div><div></div><div></div><div></div></div>
4	Gondes Putri	37	In Progress	<div><div></div><div></div><div></div><div></div><div></div></div>

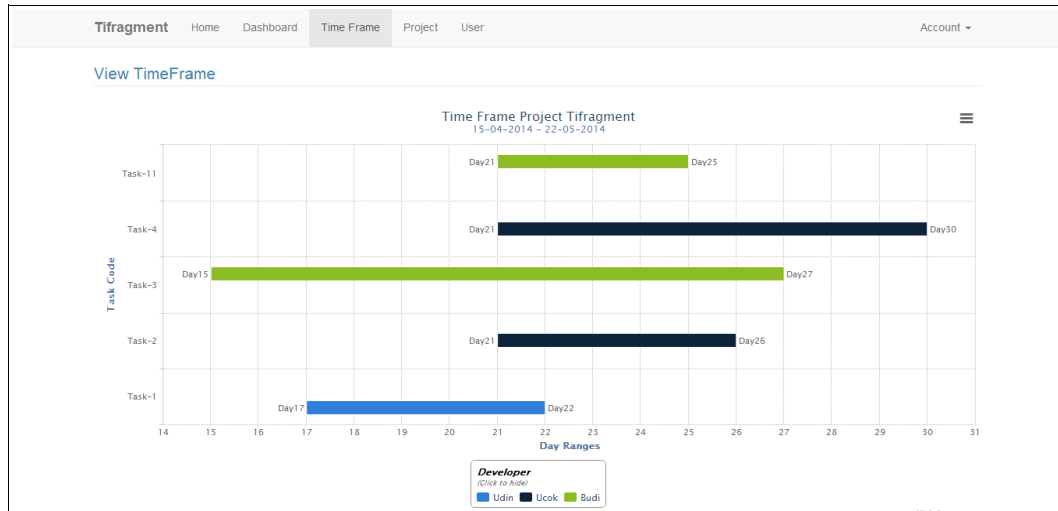
New

Found 4 Row(s)

3.9.14 Detail Time Frame

Tifragment	Home	Dashboard	Time Frame	Project	User	Account ▾	
Detail Time Frame							
Name	<input type="text" value="Legolas"/>						
Project	<input type="text" value="Dandelion"/>						
Project Leader	<input type="text" value="Aries Prayoga"/>						
Start Date	<input type="text" value="15-04-2014"/>						
End Date	<input type="text" value="30-04-2014"/>						
Total Days	<input type="text" value="15"/>						
Description	<div></div>						
List of Tasks							
Task Code	Task Name	Developer	Start Date	End Date	Total Days	Description	Operation
Task-9	create mockup	Udin Mangkunegoro	15-04-2014	30-04-2014	15	create mockup for all requirements	🔍 ✕
Task-8	design database	Udin Mangkunegoro	21-04-2014	26-04-2014	5	create table and it's relation, create eer, define procedure and function	🔍 ✕
Task-7	prepare server	Budi Sunarno	21-04-2014	24-04-2014	3	install and configure server	🔍 ✕
Task-6	prepare library	Budi Sunarno	17-04-2014	19-04-2014	2	add dependency library into project	🔍 ✕
Task-5	create project	Budi Sunarno	15-04-2014	17-04-2014	2	create project using eclipse	🔍 ✕
<div>Back</div> <div>Add Task</div> <div>View Chart</div>							

3.9.15 View Time Frame



3.9.16 Manage Account

Tifragment Home Dashboard Time Frame Project User Account ▾

[Detail User](#)

Username

Name

Full Name

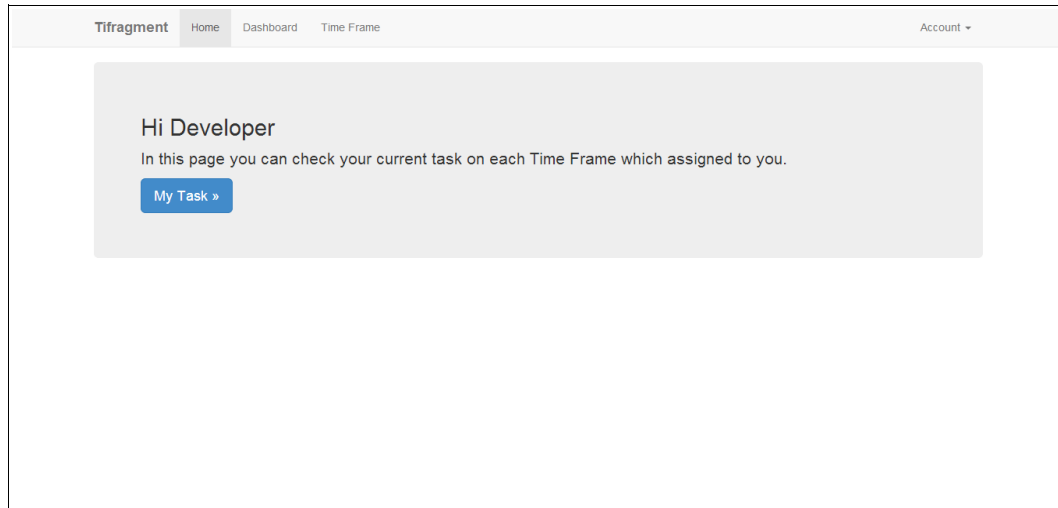
Phone Number

Email

Position

Status

3.9.17 Halaman Utama User Developer



3.9.18 List Time Frame User Developer

Tifragment Home Dashboard Time Frame Account ▾				
List of Time Frame				
No	Name	Total Days	Status	Operation
1	HyperStone	30	In Progress	📄 🗑️
2	Gondangdia	3	In Progress	📄 🗑️
3	Legolas	15	In Progress	📄 🗑️
4	Gondes Putri	37	In Progress	📄 🗑️
Found 4 Row(s)				

3.10 Proses Upload dan Alamat URL

Aplikasi ini di deploy di atas sebuah Cloud PaaS yaitu **OpenShift** dari Redhat. Dengan rincian platform sebagai berikut :

- Tomcat 7 (JBoss EWS 2.0)
- MySQL 5.5
- Jenkins
- phpMyAdmin 4.0

Proses compile aplikasi menggunakan bantuan Continuous Integration yaitu Jenkins. Jenkins mengintegrasikan kode aplikasi kita dengan platform server yaitu Tomcat dan MySQL. Dengan begitu dapat memudahkan kita dalam melakukan proses deployment sebuah aplikasi karena semua terintegrasi dalam satu platform.

3.10.1 Proses Build dengan Jenkins

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete Build

Git Build Data

Previous Build

Jenkins

tifragment-build

#22

Console Output

Started by user Jenkins System Builder

Building remotely on tifragmentbldr in workspace /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo

Checkout: repo / /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo - hudson.remoting.Channel@1fff60:tifragmentbldr

Using strategy: Default

Last Built Revision: Revision 91878428d8b95ff6adfd470e029a4a52b8132c (origin/HEAD, origin/master)

Checkout: repo / /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo - hudson.remoting.LocalChannel@1b3fff02

Wiping out workspace first.

Cloning the remote Git repository

Cloning repository origin

Fetching upstream changes from ssh://532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:/var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo

Seen branch in repository origin/HEAD

Seen branch in repository origin/master

Commencing build of Revision fb39b038e914b4d28918b1c0208e23c6d29af4ef (origin/HEAD, origin/master)

Checking out Revision fb39b038e914b4d28918b1c0208e23c6d29af4ef (origin/HEAD, origin/master)

Warning : There are multiple branch change sets here

[repol] \$ /bin/sh -xe /tmp/hudson2493091001759792469.sh

+ source /usr/lib/openshift/cartridge_sdk/bash/sdk

++ '[' false == true ']'

++ OO_BASH_SOURCE

+ alias 'export' --delete-after -az -e '\''/usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh'\''

+ upstream_ssh=532800664382eca9f5000185@tifragment-ariesp.rhcloud.com

+ rm -f /var/lib/openshift/5352971b50044641f100006c/app-deployments/current/metadata.json

+ marker_presents force_clean_build

+ '[' -f /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/.openshift/markers/force_clean_build ']'

+ set -e

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh '532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:5OPENSHIFT_BUILD_DEPENDENCIES_DIR' /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/build-dependencies/

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh '532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:5OPENSHIFT_DEPENDENCIES_DIR' /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/dependencies/

+ set -e

+ gear build

Building git ref 'origin/HEAD', commit fb39b03

Using Maven mirror /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/.openshift/config/settings.rhcloud.xml

Apache Maven 3.0.3 (r1076437; 2011-06-20 13:22:37-0400)

Maven home: /etc/alternatives/maven-3.0

Java version: 1.7.0_81, vendor: Oracle Corporation

Java home: /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.81/jre

Default locale: en_US, platform encoding: ANSI_X3.4-1968

[INFO] Scanning for projects...

[INFO] -----

[INFO] Building Tifragment 1.0

[INFO] -----

[INFO] --- maven-clean-plugin:2.4.1:clean (default-clean) @ tifragment ---

[INFO] ---

[INFO] --- maven-resources-plugin:2.4.3:resources (default-resources) @ tifragment ---

[INFO] Using 'UTF-8' encoding to copy filtered resources.

[INFO] Copying 15 resources

[INFO] ---

[INFO] --- maven-compiler-plugin:3.0:compile (default-compile) @ tifragment ---

[INFO] Changes detected - recompiling the module!

[INFO] Compiling 54 source files to /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/target/classes

[INFO] ---

[INFO] --- maven-resources-plugin:2.4.3:testResources (default-testResources) @ tifragment ---

[INFO] Using 'UTF-8' encoding to copy filtered resources.

[INFO] skip non existing resourceDirectory /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/src/test/resources

[INFO] ---

[INFO] --- maven-compiler-plugin:3.0:testCompile (default-testCompile) @ tifragment ---

[INFO] Changes detected - recompiling the module!

[INFO] Compiling 5 source files to /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/target/test-classes

[INFO] ---

[INFO] --- maven-surefire-plugin:2.7.2:test (default-test) @ tifragment ---

[INFO] Tests are skipped.

[INFO] ---

[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ tifragment ---

[INFO] Packaging webapp

[INFO] Assembling webapp [tifragment] in /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/target/tifragment

[INFO] Processing war project

[INFO] Copying webapp resources [/var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/src/main/webapp]

[INFO] Webapp assembled in [1365 msec]

[INFO] Building war: /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/webapps/ROOT.war

[INFO] WEB-INF/web.xml already added, skipping

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 28.406s

[INFO] Finished at: Sat Apr 19 11:35:48 EDT 2014

[INFO] Final Memory: 17M/158M

[INFO] -----

+ /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com 'gear stop --conditional --exclude-web-proxy --git-ref fb39b038e914b4d28918b1c0208e23c6d29af4ef'

Stopping near

Stopping PHPMyAdmin cartridge

Waiting for stop to finish

Waiting for stop to finish

Waiting for stop to finish

Waiting for stop to finish

Waiting for stop to finish

Waiting for stop to finish

Stopping jbossesw cartridge

Sending SIGTERM to jbossesw ...

kill -9 413365

kill -9 413364

kill -9 413783

kill -9 413363

++ /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com 'gear create-deployment-dir'

+ deployment_dir=2014-04-19_11-36-19.663

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh /var/lib/openshift/5352971b50044641f100006c/app-deployments/current/metadata.json 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:app-deployments/2014-04-19_11-36-19.663/metadata.json

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh --exclude .git /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/repo/ 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:app-root/runtime/repo/

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/build-dependencies/ 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:app-root/runtime/build-dependencies/

+ rsync --delete-after -az -e /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh /var/lib/openshift/5352971b50044641f100006c/app-root/runtime/dependencies/ 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com:app-root/runtime/dependencies/

+ /usr/libexec/openshift/cartridges/jenkins/bin/git_ssh_wrapper.sh 532800664382eca9f5000185@tifragment-ariesp.rhcloud.com 'gear remotepdeploy --deployment-datetime 2014-04-19_11-36-19.663'

Preparing build for deployment

Deployment id is e8072cab

Activating deployment

Starting MySQL 5.5 cartridge

Starting PHPMyAdmin cartridge

+ rm -f /var/lib/openshift/532800664382eca9f5000185/jbossesw/tmp

+ '[' -d /var/lib/openshift/532800664382eca9f5000185/jbossesw/tmp ']'

+ for d in \$(rm -f /var/lib/openshift/532800664382eca9f5000185/jbossesw/tmp/*)

+ '[' -d /var/lib/openshift/532800664382eca9f5000185/jbossesw/tmp/* ']'

+ '[' -d /var/lib/openshift/532800664382eca9f5000185/jbossesw/work ']'

+ rm -rf /var/lib/openshift/532800664382eca9f5000185/jbossesw/work

+ ls /var/lib/openshift/532800664382eca9f5000185/app-root/runtime/repo/webapps

+ '[' ROOT.war ']'

+ rsync -r --delete /var/lib/openshift/532800664382eca9f5000185/app-root/runtime/repo/webapps/ /var/lib/openshift/532800664382eca9f5000185/jbossesw/webapps/

Starting jbossesw cartridge

Found 127.11.88.1:8080 listening port

Archiving artifacts

Finished: SUCCESS

Help us localize this page

Page generated: Apr 20, 2014 10:10:53 AM

REST API

Jenkins ver. 1.509.1

50

3.10.2 Alamat URL

- Alamat : <http://tifragment-ariesp.rhcloud.com>
- Login
 - User Project Manager
 - Username : tifragment
 - Password : tifragment
 - User Developer
 - Username : developer
 - Password : developer