

## Pengembangan Sistem Aplikasi Manajemen Proyek menggunakan Kanban Framework

Nurul Faizah<sup>1</sup>, Nurudin Santoso<sup>2</sup>, Arief Andy Soebroto<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>nurulfaizah1007@gmail.com, <sup>2</sup>nurudin.santoso@ub.ac.id, <sup>3</sup>ariefas@ub.ac.id

### Abstrak

Proyek merupakan suatu kegiatan yang bersifat sementara yang memiliki batas waktu, ruang lingkup, dan biaya. Proyek memiliki banyak jenis salah satunya yaitu proyek perangkat lunak. *Software house* mengerjakan bermacam-macam jenis proyek perangkat lunak. Setiap jenis proyek mempunyai tahap-tahap/alur untuk mengembangkannya secara berbeda-beda sesuai dengan ketentuan pada *software house* atau ketentuan dari manajer proyeknya. Dalam satu waktu yang bersamaan *software house* setidaknya mengerjakan 1-3 proyek yang dibagi menjadi beberapa tim. Setiap proyek dipimpin oleh manajer proyek yang berbeda dimana, seorang manajer proyek pada proyek tertentu dapat menjadi anggota tim dari proyek lainnya. Manajer proyek kebingungan untuk memantau progres dari proyek yang dipimpinnya dan pengerjaan tugas pada proyek yang lainnya. Membangun sistem yang dapat manajemen suatu proyek dengan menggunakan kanban dalam manajemennya yang dapat membantu manajer proyek dan anggota tim merupakan solusi yang diberikan. Dengan menggunakan kanban seorang manajer proyek dapat memvisualisasikan setiap tahap-tahap/alur untuk menyelesaikan suatu proyek. Dalam penelitian ini dalam pengembangan sistem menggunakan metode *Rapid Application Development* (RAD) dengan pendekatan *Object-Oriented* (OO) dan dibangun pada *platform website*. Pengujian pada sistem ini telah berhasil diuji diantaranya pada pengujian unit dan pengujian integrasi menggunakan *basis path* dalam teknik *white-box testing* dan pengujian validasi menggunakan *scenario based testing* dalam teknik *black-box testing* dan pengujian *compatibility* berjalan baik dalam 6 *web browsers* yang berbeda.

**Kata kunci:** manajemen proyek perangkat lunak, kanban, *website*, *rapid application development*

### Abstract

A project is a temporary activity that has a limit of time, scope and cost. Projects have many types, one of which is software project. Software house work on various types of software project. Each type of a project has different stages to develop according to the provisions of the software house or a project manager. In the same time, software house usually works at least 1-3 projects divided into several team. Each project is led by different project manager in which, a project manager on particular project can be a team member of another project. A project manager is confused to monitor the progress of the project he leads and work on other projects. The solution provided is to build a system that can manage a project using kanban for management that can help the project manager and team member. By using kanban, a project manager can visualize each stages to complete a project. In this research the system development using *Rapid Application Development* (RAD) method with *Object-Oriented* (OO) approach and build on *website platform*. Testing on this system has been successfully tested including unit testing and integration testing using *basis path* in *white-box testing* and validation testing in *black-box testing* and *compatibility testing* run well in 6 different browsers.

**Keywords:** software project management, kanban, *website*, *rapid application development*

## 1. PENDAHULUAN

Proyek merupakan suatu kegiatan yang bersifat sementara yang memiliki batas waktu, ruang lingkup dan biaya. Proyek yang dikatakan

sukses adalah proyek tersebut dikerjakan *on time*, *on delivery*, dan *on budget*. Proyek perangkat lunak merupakan salah satu jenis dari proyek. *Software House* mengerjakan bermacam-macam jenis proyek perangkat lunak.

Pada jenis proyek mempunyai tahap/alur yang berbeda-beda sesuai dengan ketentuan pada *software house* atau ketentuan dari manajer proyeknya. Dalam satu waktu yang bersamaan biasanya *software house* mengerjakan setidaknya 1-3 proyek yang dibagi menjadi tim. Setiap proyek dipimpin oleh proyek manajer yang berbeda-beda, namun seorang proyek manajer pada proyek tertentu dapat menjadi anggota tim dari proyek lainnya. Sehingga dapat kebingungan untuk memantau progres dari proyek tersebut dan pengerjaan tugas pada proyek yang lainnya. Pelaporan harian dan pelaporan mingguan merupakan pelaporan progres yang digunakan pada *software house*. Dalam pelaporan harian anggota tim mengirim progres di grup *WhatsApp* pada 15 menit sebelum pulang. Sedangkan pada pelaporan mingguan terdapat rapat setiap hari sabtu yang bertujuan untuk melaporkan progres seminggu itu. Dengan pelaporan progres seperti sebelumnya dapat menyebabkan kesusahan bagi proyek manajer untuk memantau progres setiap tugas yang telah dikerjakan maupun belum dikerjakan oleh member secara *real-time* dan kesusahan untuk merekap progres yang telah dikirimkan pada grup karena rentan akan kehilangan data pada grup *chat* tersebut.

Berdasarkan permasalahan yang diutarakan diatas maka salah satu upaya untuk mengurangi permasalahan tersebut dengan menerapkan sistem perangkat lunak berbasis *website* untuk membantu manajer proyek dan anggota tim manajemen suatu proyek perangkat lunak yang sedang dikerjakan. Sistem ini dapat manajemen suatu proyek dengan menggunakan prinsip dari kanban untuk manajemennya. Dalam sistem ini dapat memantau setiap pekerjaan yang telah diberikan, seorang manajer proyek dapat membuat alur/tahap-tahap untuk menyelesaikan proyek tersebut yang nantinya ditampilkan pada kanban *board*. Dengan adanya sistem ini seorang manajer proyek dapat memantau proyek yang dipimpinnya secara *real-time* dan anggota tim dapat melaporkan progres dari tugasnya setiap saat sehingga proyek tersebut sukses.

## 2. LANDASAN KEPUSTAKAAN

Penelitian-penelitian terdahulu, teori dan metode yang digunakan akan di jelaskan pada bagian landasan kepustakaan.

### 2.1. Kajian Pustaka

Penelitian terdahulu yang berkaitan dengan

manajemen proyek pernah dikerjakan oleh Fahat (2018) dengan judul “Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Scrum Studi Kasus CV. Nusantara Media Mandiri (CVNMM)”. Penelitian ini dilakukan karena seorang proyek manajer dapat memimpin banyak proyek. Oleh karena itu penulis membangun sistem manajemen proyek dengan menggunakan scrum pada manajemen proyeknya. Metode pengembangan perangkat lunaknya menggunakan *waterfall model* dan menggunakan pendekatan *object oriented*. Penelitian selanjutnya yaitu mengenai kanban pernah dilakukan oleh Nakazawa dan Tanaka (2015) dengan judul “*Prototype of Kanban Tool and Preliminary Evaluation of Visualizing Method for Task Assignment*”. Penelitian mengenai pengembangan prototipe dari kanban *tool*, dimana menampilkan tugas dari setiap pengembang dalam beberapa baris horizontal dan setiap *boardnya* terhubung dengan GitHub yang memiliki sinkronisasi *realtime* antara klient dan pengembang. Selain itu penelitian kanban pernah dilakukan pula oleh Ahmad,et.al (2017) dengan judul “*An empirical study of portfolio management and kanban in agile and lead software companies*”. Penelitian mengenai manajemen portofolio pada perusahaan perangkat lunak, dilakukan literatur dan wawancara pada tujuh perusahaan perangkat lunak di Finlandia. Menghasilkan bahwa minat yang meningkat dalam menggunakan kanban untuk memberikan visual lebih jelas.

### 2.2. Kanban

Kanban merupakan cara manajemen dengan memvisualisasikan seluruh alur kerja daripada melihat hasil dari setiap proses yang memiliki tujuan untuk menyeimbangkan permintaan dengan kapasitas yang tersedia dan mengidentifikasi potensi kemacetan dalam proses dan mengatasi kemacetan tersebut. Kanban Menurut Hammarberg dan Sundén(2014), kanban memiliki 3 prinsip dasar yaitu :

#### 1. Prinsip 1 (P1) - Visualize Work (Visualisasikan Pekerjaan)

Kanban *Board* merupakan model visual untuk menggambarkan alur kerjanya yang membantu untuk mengamati dan memeriksa alur dari awal hingga akhir. Kanban *board* dibuat sesuai dengan tahap-tahap untuk melakukan pengembangan perangkat lunak.

#### 2. Prinsip 2 (P2) - Limit Work In Progress

### (Batasi Pekerjaan yang Sedang Berlangsung)

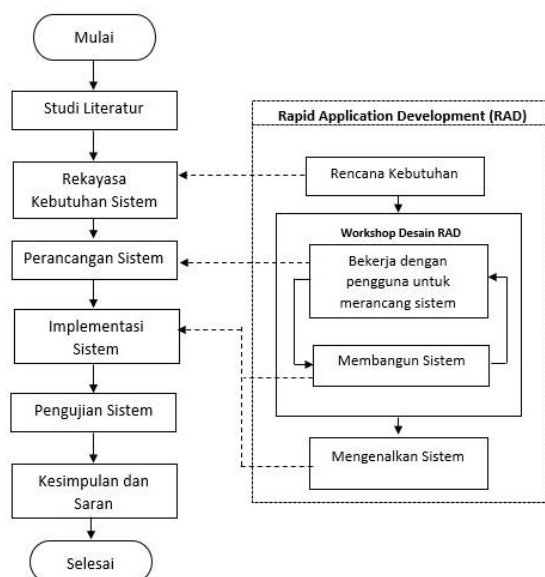
Tim pada tahap awal menentukan batas pekerjaan pada setiap alur yang ada pada kanban board yang disebut “*Work In Progress (WIP)*”. Tujuan dari WIP ini untuk mengurangi pemborosan dan membantu tim untuk fokus menyelesaikan pekerjaan yang sedang ada pada progres lalu memulai pekerjaan baru setelahnya.

### 3. Prinsip 3 (P3) - Focus on Flow of Work (Fokus pada alur kerja)

Hasil yang efektif yang didapatkan dari kanban adalah tim harus fokus pada aliran pengerjaan proyek dari awal hingga proyek selesai. Dengan mengikuti dua prinsip yang telah di jelaskan diatas untuk membantu dalam pencapaian fokus ini. Fokus pada aliran pengerjaan proyek ini mengarahkan tim untuk menggambarkan kemacetan yang akan terjadi dan menindaklanjutinya, sehingga aliran pengerjaan proyek ini dapat berjalan dengan tetap.

## 3. METODOLOGI PENELITIAN

Tahap-tahapan dalam menyelesaikan dan menyusun penelitian ini akan dijelaskan pada metodologi penelitian. Penelitian diawali dengan studi literatur sampai dengan kesimpulan dan saran. Gambar 1 merupakan diagram alir dari metodologi yang digunakan pada penelitian.



Gambar 1 Diagram Alir Metodologi Penelitian

Studi literatur merupakan awal dari penelitian ini untuk mendapatkan kerangka berpikir dengan cara mencari referensi dasar teori, konsep dasar yang akan membantu

penelitian ini yaitu pengembangan sistem aplikasi manajemen proyek menggunakan kanban *framework*.

Setelah tahap studi literatur yaitu tahap rekayasa kebutuhan yang terintegrasi dengan proses rencana kebutuhan dalam metode *Rapid Application Development (RAD)*. Mendapatkan kebutuhan apa saja yang diperlukan untuk membangun sistem dan memastikan sistem yang akan dibangun seperti kebutuhan dari pengguna merupakan tujuan dari tahap rekayasa kebutuhan.

Tahap perancangan yang terintegrasi dengan proses *workshop desain* RAD bagian bekerja dengan pengguna untuk merancang sistem dalam metode RAD yang terjadi proses iterasi dengan membangun sistem pada implementasi merupakan tahap selanjutnya. Memberikan gambaran dari sistem yang akan berjalan secara lebih detail kepada pemangku kepentingan merupakan tujuan dari tahap perancangan

Setelah dilakukan perancangan sistem dilanjutkan pada tahap implementasi sistem yang terintegrasi dengan proses *workshop desain* RAD bagian membangun sistem dan mengenalkan sistem pada metode *Rapid Application Development (RAD)* yang melakukan iterasi dengan bagian bekerja untuk merancang sistem di bagian perancangan. Pembuatan sistem berdasarkan rekayasa kebutuhan dan pula perancangan sistem kedalam tahap koding merupakan tujuan dari tahap implementasi. Selanjutnya merupakan tahap pengujian untuk menguji sistem yang telah diimplementasikan apakah sistem telah dibangun sesuai dengan sistem yang diharapkan merupakan. Tahap selanjutnya kesimpulan untuk pengambilan kesimpulan yang didasarkan dari proses rekayasa kebutuhan hingga pengujian sistem. Setelah mengambil kesimpulan, dituliskan saran untuk menjadi bahan pertimbangan penelitian yang akan datang.

## 4. REKAYASA KEBUTUHAN

Rekayasa kebutuhan dalam penelitian ini berisi proses elisitasi kebutuhan, lalu dibuat fungsionalitas sistem. *Use case diagram* dan *use case scenario* merupakan pemodelan kebutuhan yang merupakan bagian dari rekayasa kebutuhan.

### 4.1. Elisitasi Kebutuhan

Pada elisitasi kebutuhan ini dilakukan

elisitasi dengan metode wawancara. Tahap wawancara dilakukan pada tiga *software house* yang berada di Malang yaitu CV. Profile Image Studio, Komal Development Indonesia dan CV. Sarana Utama Solusindo. Pada wawancara tersebut penulis menanyakan mengenai gambaran umum seperti profil *software house* tersebut, mekanisme dari pengerjaan proyek yang bertujuan untuk mengetahui bagaimana *software house* mengerjakan suatu proyek dan memanajemennya, dan kendala mekanisme pengerjaan proyeknya untuk mengetahui permasalahan selama pengerjaan dari proyek.

## 4.2. Deskripsi Sistem

Sistem aplikasi manajemen proyek menggunakan *kanban framework* merupakan sistem yang digunakan untuk manajemen proyek perangkat lunak. Berbasis *website* dan menggunakan *kanban* untuk manajemen proyeknya. Dalam sistem terdapat fitur utama yaitu proyek merupakan fitur untuk mengelola suatu proyek, *kanban board* yang digunakan untuk visualisasi tahapan/alur dalam proyek, dan tim yang digunakan untuk mengerjakan suatu proyek.

## 4.3. Identifikasi Aktor

Sistem aplikasi manajemen proyek menggunakan *kanban framework* memiliki tiga aktor. Pengunjung, klien dan tim pengembang merupakan aktor yang ada dalam sistem ini.

## 4.4. Kebutuhan Fungsional

Kebutuhan yang dimiliki oleh sistem untuk memenuhi kebutuhan dari pengguna yang telah didefinisikan sebelumnya disebut dengan kebutuhan fungsional. Kebutuhan fungsional yang dihasilkan dalam penelitian ini adalah 34 kebutuhan fungsional yang dihasilkan melalui dua iterasi.

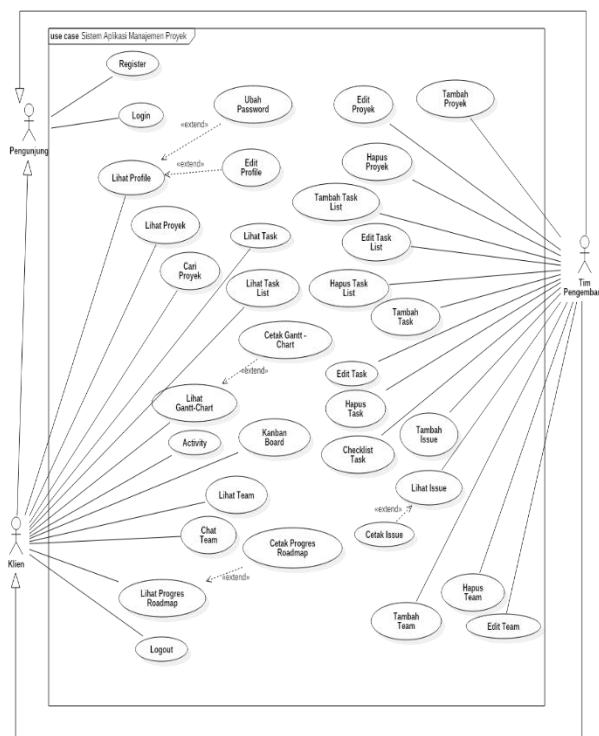
## 4.5. Kebutuhan Non Fungsional

Kebutuhan fungsional yang dihasilkan dari sistem aplikasi manajemen proyek menggunakan *kanban framework* yaitu *compatibility* bahwa sistem harus dapat berjalan dalam berbagai macam peramban (*browser*).

## 4.6. Pemodelan Kebutuhan

Pemodelan kebutuhan yang dilakukan dalam penelitian kali ini dibagi menjadi dua pemodelan yaitu *use case diagram* dan *use case*

*scenario*. Gambar 2 merupakan *use case diagram* dari sistem aplikasi manajemen proyek menggunakan *kanban framework* yang memiliki 34 *use case*. Penjelasan alur dari masing-masing *use case* dijelaskan lebih detail dalam *use case scenario*.



Gambar 2. Use Case Diagram

## 5. PERANCANGAN

Perancangan dalam penelitian ini akan menjelaskan mengenai gambaran pada sistem yang akan dibangun. Perancangan antarmuka, perancangan basis data, perancangan komponen dan perancangan arsitektur adalah bagian dari tahap perancangan.

### 5.1. Perancangan Arsitektur

Permodelan *class diagram* dan permodelan *sequence diagram* merupakan perancangan arsitektur yang dilakukan dari sistem aplikasi manajemen proyek menggunakan *kanban framework*. Permodelan *sequence diagram* dibuat menjadi 5 sampel *sequence diagram* yang dibagi menjadi *sequence diagram* iterasi pertama yaitu 3 sampel dan *sequence diagram* iterasi kedua yaitu 2 sampel. Pendefinisian struktur kelas yang ada dalam sistem dijelaskan pada tahap permodelan *class diagram*. Pada permodelan *class diagram* dihasilkan 6 kelas *controller* yaitu kelas *Board*, *Task*, *Task\_List*, *Team*, *User* dan *Project*, 6 kelas *model* yaitu



*M\_task\_list*, *M\_user*, *M\_board*, *M\_team*, *M\_task* dan *M\_project* dan 24 *view*.

## 5.2. Perancangan Basis Data

Perancangan basis data dilakukan pembuatan *Conceptual Data Model* (CDM) yang akan dijelaskan lebih detail pada *Entity Relationship Diagram* (ERD), pada iterasi pertama dan kedua pembuatan ERD dihasilkan 13 entitas yang ditransformasikan menjadi *Physical Data Model* (PDM) yang memiliki hasil 13 tabel antara lain tabel *activity*, *assigned\_to*, *board*, *chat*, *member*, *project*, *project\_member*, *status*, *task*, *task\_list*, *team*, *user* dan *issue*.

## 5.3. Perancangan Komponen

Perancangan komponen dalam sistem ini menggunakan tiga sampel dari algoritme yaitu fungsi tambah proyek, tambah tim dan edit tim. Pada jurnal ini hanya ditampilkan algoritme untuk fungsi tambah proyek yang dijabarkan pada Tabel 1.

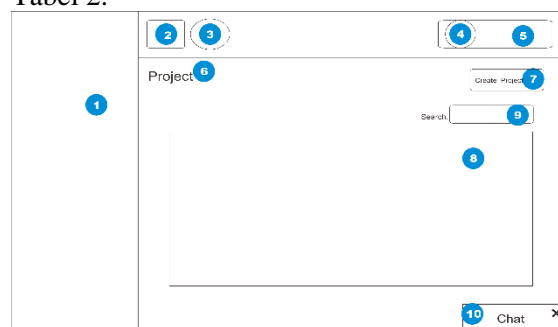
Tabel 1. Algoritme Tambah Proyek

Algoritme	
1	MULAI
2	Menetapkan peraturan yang berisi peraturan validasi form
3	IF (Validasi bernilai true)
4	Memasukkan data proyek pada database
5	Inisialisasi variabel <i>project_id</i> berisi id data proyek yang baru disimpan
6	Memasukkan data kanban board pada database sesuai dengan <i>project_id</i>
7	Inisialisasi variabel <i>member</i> berisi data <i>member</i> sesuai dengan <i>team id</i> yang dipilih
8	Melakukan perulangan (sebanyak variabel <i>member</i> )
9	Memasukkan data <i>member</i> yang mengikuti <i>project</i> tersebut pada database
10	Akhir Perulangan
11	Memanggil library bernama <i>All_library</i> untuk memasukkan aktivitas tambah proyek
12	Inisialisasi variabel <i>status_email</i> berisi nilai kembalian dari pemanggilan <i>method</i> untuk mengirim email
13	IF ( <i>status_email</i> = true)
14	Mengirimkan pesan melalui session flash data bernama <i>success</i> berisi "Proyek Baru Bernama nama proyek Telah Berhasil dibuat & Pemberitahuan pada email telah terkirim!"
15	redirect ke halaman proyek
16	ELSE
17	Mengirimkan pesan melalui session flash data

	bernama <i>success</i> berisi "Proyek Baru bernama nama proyek Telah berhasil dibuat. Namun, pemberitahuan ke email gagal dikirim!"
18	redirect ke halaman proyek
19	ENDIF
20	ELSE
21	Mengirimkan pesan melalui session flash data bernama <i>failed</i> berisi "Input tidak boleh ada yang kosong!"
22	redirect ke halaman proyek
23	END IF
24	SELESAI

## 5.4. Perancangan Antarmuka

Perancangan antarmuka yang dihasilkan dari penelitian ini adalah perancangan antarmuka dari halaman *dashboard* yang berada pada Gambar 3 dan detail penjelasan dari pada Tabel 2.



Gambar 3. Perancangan Antarmuka Halaman *Dashboard*

Tabel 2. Penjelasan Antarmuka Halaman *Dashboard*

No	Deskripsi
1	<i>Side bar</i> berisi menu-menu
2	Tombol untuk menampilkan atau menyembunyikan <i>side bar</i>
3	Tombol untuk menampilkan <i>input text</i> untuk mencari judul proyek
4	Gambar profil aktor
5	Menampilkan nama aktor dan <i>dropdown</i> yaitu <i>menu profile dan logout</i>
6	Label dari judul halaman
7	Tombol untuk menampilkan modal form tambah proyek
8	Tabel berisi daftar proyek yang dikerjakan
9	<i>Input text</i> untuk mencari proyek yang ada pada tabel semua proyek
10	Label bertuliskan chat untuk menampilkan daftar tim yang diikuti aktor

## 6. IMPLEMENTASI

Implementasi sistem dari sistem aplikasi manajemen proyek menggunakan *framework*

*CodeIgneter*. Implementasi basis data, spesifikasi sistem, implementasi antarmuka dan implementasi komponen merupakan bagian dari tahap implementasi. Hasil dari implementasi sistem dapat diakses pada tautan [kanpro.4pps.web.id](http://kanpro.4pps.web.id).

### 6.1. Spesifikasi Sistem

Spesifikasi dari sistem yang digunakan dalam implementasi sistem aplikasi manajemen proyek menggunakan kanban *framework* yaitu spesifikasi dari perangkat lunak dan spesifikasi dari perangkat keras.

### 6.2. Implementasi Basis Data

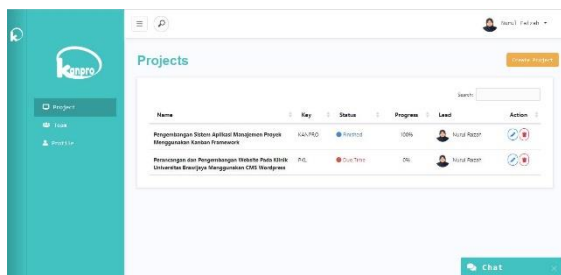
Implementasi basis data yang digunakan pada sistem aplikasi manajemen proyek menggunakan kanban *framework* yaitu phpMyAdmin. Hasil dari Implementasi basis data menghasilkan 13 tabel *user*, *project*, *team*, *chat*, *task\_list*, *task*, *activity*, *board*, *member*, *issue*, *assigned\_to*, *status* dan *project\_member*.

### 6.3. Implementasi Komponen

Pada implementasi komponen sistem aplikasi manajemen proyek menggunakan kanban *framework* ini menggunakan kode program yaitu bahasa pemrograman PHP untuk mengimplementasikannya.

### 6.4. Implementasi Antarmuka

Hasil implementasi antarmuka salah satunya yaitu implementasi antarmuka halaman *dashboard* yang terdapat pada Gambar 4.



Gambar 4. Implementasi Antarmuka Halaman Dashboard

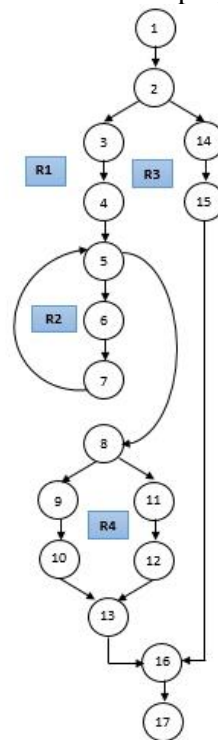
## 7. PENGUJIAN

Pengujian yang dipilih untuk menguji sistem aplikasi manajemen proyek menggunakan kanban *framework* yaitu pengujian integrasi, pengujian validasi,

pengujian unit dan pengujian unit.

### 7.1. Pengujian Unit

Pengujian unit ini akan menguji tiga sampel yaitu fungsi tambah proyek, tambah team dan mengedit team. Kelas *TestDriver* pada pengujian ini dibutuhkan untuk menjalankan komponen yang ingin diuji. Gambar 5 merupakan contoh dari hasil *flow graph* dari pengujian yang dilakukan pada *method addNewProject()* yang berfungsi untuk menambah proyek.



Gambar 5. Flow Graph Method *addNewProject()*

#### 1. Cyclomatic Complexity

- 1)  $V(G) = \text{Jumlah Region} = 4$
- 2)  $V(G) = \text{edge} - \text{node} + 2 = 2 + 2 = 4$
- 3)  $V(G) = \text{predicate node} + 1 = 3 + 1 = 4$

#### 2. Independent Path

Jalur 1 = 1 - 2 - 14 - 15 - 16 - 17

Jalur 2 = 1 - 2 - 3 - 4 - 5 - 8 - 11 - 12 - 13 - 16 - 17

Jalur 3 = 1 - 2 - 3 - 4 - 5 - 8 - 9 - 10 - 13 - 16 - 17

Jalur 4 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 5 - 8 - 9 - 10 - 13 - 16 - 17

Berdasarkan hasil uji dari jalur *independent* sebelumnya yang digunakan sebagai dasar dari kasus uji dari algoritme dengan fungsi

*addNewProject()*.

#### 1. Jalur 1

- Kasus Uji: Klas *TestDriver* menjalankan *method addNewProject()* dengan *validation = false*, nama\_proyek – ‘Proyek Kanban’, *num\_row\_member = 4* dan *status\_email = true*.
- *Expected Result* : Menunjukkan pesan berisi “input tidak boleh ada yang kosong”.
- *Result* : Menunjukkan pesan berisi “input tidak boleh ada yang kosong”.
- Status : *Pass*

#### 2. Jalur 2

- Kasus Uji: Klas *TestDriver* menjalankan *method addNewProject()* dengan *validation = true*, nama\_proyek – ‘Proyek Kanban’, *num\_row\_member = 0* dan *status\_email = false*.
- *Expected Result* : Perulangan tidak dijalankan dan menunjukkan pesan “Tidak menjalankan perulangan. Proyek baru bernama Proyek Kanban telah berhasil dibuat. Namun, pemberitahuan email gagal dikirim”
- *Result* : Perulangan tidak dijalankan dan menunjukkan pesan “Tidak menjalankan perulangan. Proyek baru bernama Proyek Kanban telah berhasil dibuat. Namun, pemberitahuan email gagal dikirim”
- Status : *Pass*

#### 3. Jalur 3

- Kasus Uji: Klas *TestDriver* menjalankan *method addNewProject()* dengan *validation = true*, nama\_proyek – ‘Proyek Kanban’, *num\_row\_member = 0* dan *status\_email = true*.
- *Expected Result* : Perulangan tidak dijalankan dan menampilkan pesan “Tidak menjalankan perulangan. Proyek baru bernama Proyek Kanban telah berhasil dibuat & pemberitahuan email telah terkirim”.
- *Result* : Perulangan tidak dijalankan dan menampilkan pesan “Tidak menjalankan perulangan. Proyek baru bernama Proyek Kanban telah berhasil dibuat & pemberitahuan email telah terkirim”.

- Status : *Pass*

#### 4. Jalur 4

- Prosedur Uji : Klas *TestDriver* menjalankan *method addNewProject()* dengan *validation = true*, nama\_proyek – ‘Proyek Kanban’, *num\_row\_member = 4* dan *status\_email = true*.
- *Expected Result* : Menjalankan perulangan dan menampilkan pesan “Menjalankan perulangan sebanyak 4 kali. Proyek baru bernama Proyek Kanban telah berhasil dibuat & pemberitahuan email telah terkirim”.
- *Result*: Menjalankan perulangan dan menampilkan pesan “Menjalankan perulangan sebanyak 4 kali. Proyek baru bernama Proyek Kanban telah berhasil dibuat & pemberitahuan email telah terkirim”.
- Status : *Pass*

### 7.2. Pengujian Integrasi

Dalam pengujian integrasi ini digunakan *basis path* dalam teknik *white-box testing*. Klas *TestDriver* pada pengujian ini dibutuhkan untuk menjalankan klas yang ingin diuji. Terdapat satu sampel *method* yaitu *method addMemberProject(id\_project)* pada klas *Project* yang memanggil *method addMemberProject(id\_project, member)* dan *method getNamaProject(id\_project)* pada klas *M\_project* menghasilkan *pass* pada semua jalur uji.

### 7.3. Pengujian Validasi

Teknik *black-box testing* merupakan teknik yang digunakan dalam pengujian validasi. Dari hasil pengujian sistem aplikasi manajemen proyek menggunakan *kanban framework* terdapat 52 kasus uji dan dalam semua kasus uji menghasilkan 100% valid yang berarti seluruh spesifikasi kebutuhan telah terpenuhi.

### 7.3. Pengujian Compatibility

Tahap untuk menguji sistem aplikasi manajemen proyek menggunakan *kanban framework* yang dikembangkan dapat dijalankan dalam berbagai *browser* merupakan tahap pengujian *compatibility*. *SortSite* merupakan *tools* yang digunakan dalam pengujian ini dan dihasilkan bahwa sistem dapat berjalan baik

pada enam *browser* berbeda.

## 8. PENUTUP

Sistem Aplikasi Manajemen Menggunakan Kanban *Framework* telah berhasil dikembangkan. Rekayasa kebutuhan menghasilkan 3 aktor, 34 kebutuhan fungsional dan 1 kebutuhan non fungsional. Perancangan arsitektur menghasilkan 5 sampel fungsi untuk *sequence diagram*. *Class diagram* menghasilkan 6 klas *controller*, 6 klas *model* dan 24 *view*. Perancangan basis data menghasilkan 13 tabel pada PMD. Pada perancangan komponen menggunakan lima sampel algoritme. Sedangkan untuk perancangan antarmuka merupakan rancangan antarmuka yang akan diterapkan pada sistem. Implementasi menghasilkan bahwa sistem menggunakan bahasa pemrograman PHP, basis data phpMyAdmin yang menghasilkan 13 tabel dan dibangun berbasis *website* menggunakan CodeIgneter untuk *framework websitenya*,

Pengujian menghasilkan 100% *pass* pada seluruh jalur dalam pengujian integrasi dan pengujian unit dengan menggunakan *basis path testing* dalam teknik *white-box testing*. Pengujian integrasi 100% valid dalam semua 52 kasus uji yang didefinisikan dengan menggunakan teknik *black-box testing*. Berjalan baik pada 6 *browser* yang berbeda ketika dilakukan pengujian *compatibility*.

Saran untuk bahan pertimbangan dalam pengembangan sistem aplikasi manajemen proyek menggunakan kanban *framework* selanjutnya adalah penelitian dapat dikembangkan dengan memberikan pilihan pada pengguna dengan metode kanban atau scrum pada sistem, notifikasi pada *navbar* yang berisi pemberitahuan untuk mengetahui tugas/*task* baru yang ditugaskan pada aktor dan terintegrasi dengan dokumen pada setiap task sehingga pada saat menyelesaikan task terdapat dokumennya dan dapat dimonitoring.

## 9. DAFTAR PUSTAKA

- Ahmad, M.O., Lwakatare, L.E., Kuvaja, P., Oivo, M. & Markkula, J., 2017. An Empirical study of portfolio management and Kanban in agile lean software companies. *Journal of Software : Evaluation and Process*, 29(6).
- Faisal, M.F., 2018. Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Scrum Studi Kasus CV.

Nusantara Media Mandiri (CVNMM).S1. Universitas Brawijaya.

- Hammarberg, M., and Sundén, J., 2014. *Kanban in Action*. New York: Manning Publications.
- Nakazawa, S., and Tanaka, T., 2015. *Prototype of Kanban Tool and Preliminary Evaluation of Visualizing Method for Task Assignment*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/7372311>> [Diakses 20 November 2019].
- Schwalbe, K., 2011. *Information Technology Project Management*. 6rd Edition. Boston : Course Technology.
- Wysocki, R.K., 2006. *Effective Software Project Management*. Indianapolis: Wiley Publishing, Inc.