

## Rekayasa Ulang Iteratif pada Aplikasi I-Lab Universitas Muhammadiyah Malang

Arif Rahmadhani<sup>1</sup>, Wildan Suharso<sup>2</sup>, Gita Indah Marthasari<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika/Universitas Muhammadiyah Malang

arif\_437165@webmail.umm.ac.id<sup>1</sup>, wsuharso@umm.ac.id<sup>2</sup>, gita@umm.ac.id

### Abstrak

Sistem informasi adalah hal yang penting bagi suatu perusahaan atau instansi seperti I-Lab yang merupakan sistem informasi pada Laboratorium Teknik Informatika Universitas Muhammadiyah Malang. Kemudian pihak Laboratorium Teknik Informatika ingin melakukan perubahan sistem untuk meningkatkan layanan kepada mahasiswa maupun dosen serta untuk menanggapi perubahan kurikulum baru yang diterapkan pada jurusan Teknik Informatika. Pada penelitian ini, dilakukan rekayasa ulang dengan menggunakan metode rekayasa ulang iteratif yaitu melakukan rekayasa ulang yang dilakukan dengan cara bertahap. Adapun tahapan yang dilalui dalam penelitian ini adalah analisis sistem warisan, klasifikasi data, desain ulang basisdata, adaptasi program warisan, migrasi data, desain ulang fungsi, tes ekivalensi, dan membersihkan data residual. Pada penelitian ini berhasil melakukan rekayasa ulang sebagian pada komponen penilaian akhir praktikum, penjadwalan praktikum, dan menampilkan seluruh user serta dilakukan penambahan komponen penilaian ujian praktikum.

**Kata Kunci:** Sistem Informasi, Rekayasa Ulang, Kebutuhan, Perubahan Sistem, Iteratif

### Abstract

Information systems are important for a company or agency such as I-Lab which is an information system at the Informatics Engineering Laboratory of the University of Muhammadiyah Malang. Then the Informatics Engineering Laboratory wants to make a system change to improve services to students and lecturers as well as to respond to changes in the new curriculum applied to the Department of Informatics. In this study, a re-engineering was carried out using an iterative reengineering method, namely reengineering carried out in a phased manner. The stages passed in this research are analyze legacy system, data classification, redesign database, adapt legacy component, migrate data, redesign function, equivalence tests, and empty residual data. In this study, it succeeded in re-engineering a portion of the components of the final practicum assessment, practicum scheduling, and displaying all users as well as adding a component of the practical exam assessment.

**Keywords:** Information System, Reengineering, Requirements, System Change, Iterative

### 1. Pendahuluan

Sistem informasi adalah hal yang penting bagi suatu perusahaan atau instansi bisa di ibaratkan sebagai tulang punggung dari aliran informasi yang ada di perusahaan atau instansi [1]. Sistem dan data yang ada pada sistem informasi merupakan asset yang berharga bagi perusahaan atau instansi tersebut [2]. Perkembangan perusahaan harus diikuti juga dengan berkembangnya sistem informasi dalam perusahaan. Dalam bukunya Ian Sommerville mengemukakan perubahan perangkat lunak dimana perangkat lunak lama perlu dilakukan rekayasa ulang supaya dapat memenuhi kebutuhan baru yang dibutuhkan saat ini dan kedepannya[3].

Pada saat ini Laboratorium Informatika Universitas Muhammadiyah Malang telah memiliki sebuah sistem informasi dan manajemen yang disebut dengan i-lab. Seiring berjalannya waktu pihak Lab Informatika Universitas Muhammadiyah Malang ingin melakukan perubahan sistem karena kebutuhan guna meningkatkan layanan untuk mahasiswa maupun dosen dan menanggapi perubahan kurikulum yang digunakan pada jurusan Teknik Informatika.

*Reengineering* atau rekayasa ulang merupakan proses membentuk ulang sebuah sistem yang mana nantinya sistem yang telah di rekayasa ulang diharapkan lebih banyak fungsionalitasnya, meningkatkan *performance & realibility*, juga kemampuan untuk

*maintainability*-nya bertambah [4]. Pada penelitian sebelumnya menerangkan bahwa manfaat dari *reengineering* sebuah *software* adalah meningkatkan *maintainability*, maningkatkan performa, menigkatkan *interoperability*, menurunkan ketergantungan terhadap personal, dan meningkatkan *testability* [5].

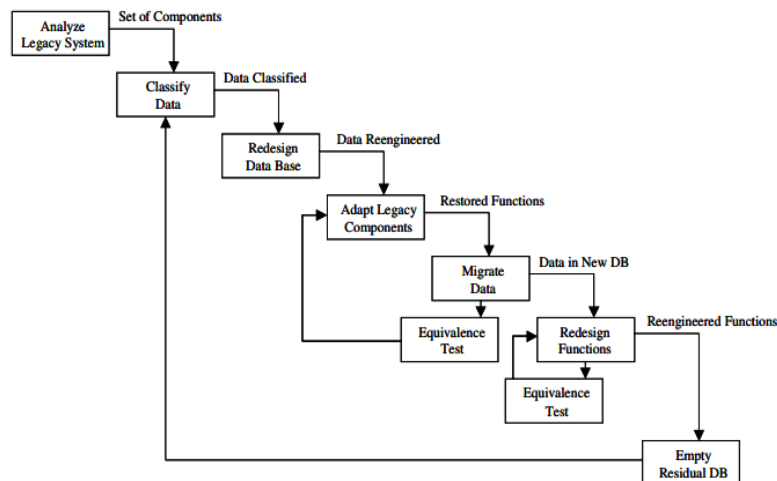
Proses Rekayasa Ulang Iteratif adalah rekayasa ulang sistem yang dilakukan dengan cara bertahap. Adapun dalam proses dilakukannya Rekayasa Ulang Iteratif terdapat berbagai macam tahapan komponen. Komponen terdiri dari beberapa fase evolusi sistem yaitu yang pertama *Legacy*, merupakan bagian(komponen) dari sistem warisan atau bagian(komponen) yang belum dilakukan rekayasa ulang. Kedua adalah *Restored*, merupakan bagian(komponen) yang digunakan kembali dari sistem warisan dengan tujuan agar dapat beradaptasi dengan data yang baru telah direkayasa ulang. Ketiga adalah *Reengineered*, merupakan bagian(komponen) dari sistem warisan yang telah dimodifikasi dan menjadi bentuk yang baru dan mencapai tingkatan tertentu. Dan yang keempat adalah *New*, merupakan bagian(komponen) yang tidak terdapat pada sistem warisan, tetapi ditambahkan untuk memenuhi kebutuhan baru pada sistem dengan domain yang masih sama sebagai pelengkap sistem yang baru [6].

Penelitian sebelumnya melakukan *reengineering* pada aplikasi desktop BPPKI Surabaya yang masih menggunakan sistem *offline* sehingga tidak dapat diakses secara *online*. Hasil dari penelitian berupa aplikasi sistem informasi manajemen kepegawaian dimana dalam sistem tersebut terdapat beberapa bagian yaitu sistem informasi proses transaksi kenaikan pangkat, gaji pokok, pengajuan mutasi, pengajuan pensiun hingga pengajuan cuti dimana aplikasi ini berbasis website dan menggunakan framework Laravel [7]. Pada penelitian sebelumnya melakukan rekayasa ulang karena sistem warisan tidak mendapatkan perkembangan dengan menyesuaikan keadaan lingkungan perusahaan sehingga perlu di rekayasa ulang agar sistem warisan dapat digunakan kembali. [8] [9] [10]. Perbedaan dalam penelitian yang akan dilakukan adalah dengan melakukan studi kasus baru pada aplikasi I-Lab UMM.

Dalam penelitian ini permasalahan yang dibahas adalah Bagaimana melakukan proses rekayasa ulang dengan metode iteratif pada aplikasi I-Lab Laboratorium Informatika Universitas Muhammadiyah Malang dan bagaimana pengaruh dari rekayasa ulang terhadap kinerja sistem dilihat dari atribut respon time sistem. Dan tujuan dari penelitian ini adalah bagaimana melakukan untuk menerapkan model rekayasa ulang iteratif pada aplikasi I-Lab Laboratorium Informatika Universitas Muhammadiyah Malang.

## 2. Metode Penelitian

Adapun metodologi penelitian yang digunakan dalam penelitian ini yang ditunjukkan pada Gambar 1.



Gambar 1. Metode penelitian

### 2.1 Analisa sistem warisan

Fase awal dari rekayasa ulang membutuhkan identifikasi, analisis, dan interpretasi dari komponen dan data di sistem warisan. Proses adalah analisis komponen dari sistem warisan untuk memutuskan komponen-komponen yang bisa dipecah. Komponen ini dipartisi dan

bertujuan sehingga tidak ada komponen yang saling ketergantungan antar komponennya. Komponen yang berdiri sendiri ini nantinya bisa mulai dilakukan rekayasa ulang pada komponen tersebut. Setelah dilakukan proses Rekayasa Ulang kemudian menghasilkan komponen baru yang terbentuk dan berada pada sistem tujuan. Dalam hal ini Rekayasa Ulang komponen antara iterasi yang pertama tidak tergantung oleh Rekayasa Ulang komponen pada tahap yang lainnya [2].

## 2.2 Klasifikasi data

Fase kedua dari rekayasa ulang iteratif terdiri dari aktifitas identifikasi dan interpretasi dari data beserta relasinya yang berada pada basis data warisan. Fase ini juga dinamakan dengan Rekayasa Balik (*reverse engineering*). Pada fase klasifikasi data ini diutamakan pada pengelompokan data menjadi data primer dan data residual [2].

## 2.3 Desain ulang basis data

Fase ketiga yaitu desain ulang data dimana ketika melakukan desain ulang basis data, setelah berbagai macam data yang telah diklasifikasikan harus dilakukan restrukturisasi pada basis data tujuan agar menjadi lebih efektif. Restrukturisasi yang dimaksud yaitu melakukan desain ulang organisasi data, mengidentifikasi relasi, ketergantungan, *foreign key* dan *primary key* diantara entitas pada *database* tujuan [2].

## 2.4 Mengadaptasi komponen sistem warisan

Pada fase ke empat ini setiap program pada sistem warisan yang mengakses data baru yang telah direkayasa ulang pada awalnya harus diadaptasikan agar dapat mengakses basis data tujuan. Dalam hal ini menunjukkan bahwa setiap program harus dilakukan analisis terlebih dahulu guna mengidentifikasi instruksi pengaksesan data.

## 2.5 Memindahkan data

Pada fase ini perpindahan data dirasa sangat penting dilakukan yaitu pada basis data warisan secara bertahap dipindahkan ke dalam basis data tujuan, sedangkan data yang tidak terkait dalam rancangan sistem tujuan dipindahkan ke dalam residual basis data.

## 2.6 Mendesain ulang fungsi

Dalam tahapan ini akan dilakukan proses desain ulang fungsi dan komponen, sehingga fungsi secara struktur menjadi lebih baik. Adapun yang akan dilakukan di tahapan ini yaitu melakukan restrukturisasi komponen yang direkayasa ulang, menyatukan proses yang berbeda tetapi mempunyai fungsi yang sama, melakukan update data dari komponen yang baru, melakukan perbaikan algoritma yang sudah lama digunakan, melakukan perbaikan antar muka pengguna dan modul antar muka, meng-*update* bahasa pemrograman menjadi bahasa yang baru yang lebih baik dari Bahasa pemrograman sebelumnya, dan melakukan eksekusi dari hasil komponen yang dimunculkan kembali.

## 2.7 Tes ekivalensi

Setelah dilakukan desain ulang fungsi pada tahap sebelumnya maka program atau komponen siap untuk dijalankan dan mempunyai fungsi yang sama dengan komponen pada sistem warisan maka dilakukan pengujian untuk memastikan bahwa komponen pada sistem yang direkayasa ulang mempunyai persamaan dengan yang ada pada sistem warisan.

## 2.8 Mengosongkan data residual

Pada tahap ini dilakukan pembersihan data pada basis data residual karena data pada basis data tersebut tidak penting atau tidak dibutuhkan pada sistem tujuan. Pada penelitian sebelumnya data residual ditempatkan pada basis data tujuan namun dengan catatan data tersebut pada akhir proses rekayasa ulang akan dihapus karena tidak ada komponen warisan yang mengakses data tersebut.

## 2.9 Iterasi

Tahap ini ciri khas dari rekayasa ulang iteratif. Setiap komponen yang direkayasa ulang dilakukan dengan tahapannya masing-masing. Jadi ketika satu komponen direkayasa ulang

maka komponen itu akan melalui tahapan dari analisis sistem warisan hingga tes ekivalensi baru melanjutkan ke komponen berikutnya.

### 3. Hasil Penelitian dan Pembahasan

Hasil dari penelitian ini adalah berupa aplikasi baru yang telah direkayasa ulang menggunakan metode rekayasa ulang iteratif. Berikut merupakan proses dari rekayasa ulang iteratif:

#### 3.1 Analisa sistem warisan

Tahap ini menganalisa sistem warisan yaitu I-Lab. Dimana sudah disebutkan pada penjelasan sebelumnya bahwa sistem warisan memiliki lebih dari 70 modul. Pada basis data sistem I-Lab juga terdapat kurang lebih 49 table. Dalam kasus ini penelitian hanya dilakukan pada komponen penilaian ujian praktikum (m1) merupakan tambahan baru yang dikembangkan dari modul sebelumnya yaitu *visitor\_teacher\_my\_class\_task*, kemudian komponen penilaian akhir praktikum pada modul *visitor\_teacher\_my\_class\_final\_grade(m2)*, komponen jadwal praktikum ada pada modul *admin\_schedule (m3)* dan juga modul *admin\_dashboard (m4)*, dan komponen manajemen semua *user* pada modul *admin\_user (m5)*. Dari Analisa sistem warisan didapatkan hasil pada Tabel 1 dibawah ini.

Tabel 1. Hasil Analisa Sistem Warisan

No.	Table	m1	m2	m3	m4	m5
1	user	R	R		R	CRUD
2	user_student	R	R		R	CRUD
3	user_asisten	R	R		R	CRUD
4	user_instructor		R		R	CRUD
5	user_student_class	R	R			
6	class	R	RU	RUD	R	
7	Presence		R		R	
8	presence_student		R			
9	Task	CRD	R		R	
10	task_detail	CRUD	R			
11	User_admin				R	CRUD

Dari hasil Analisa sistem warisan didapat bahwa ada 5 modul yang terkait dengan komponen yang direkayasa ulang dan melibatkan 11 *table* pada database. Pada tabel 1 juga terdapat notasi CRUD (*Create, Read, Update, Delete*) yang merupakan akses dari 5 modul terhadap data pada 11 *table* tersebut.

#### 3.2 Klasifikasi data

Tahap ini adalah proses pengklasifikasian basis data warisan. Proses klasifikasi data ini akan dibatasi hanya pada data yang berhubungan dengan proses penilaian ujian praktikum, penilaian akhir praktikum, jadwal praktikum dan manajemen semua user. Data secara umum bersifat primer atau esensial, namun ada beberapa data yang bersifat residual. Data residual dalam penelitian ini merupakan data yang tingkat record berisikan null yang tinggi dan tidak ada data yang terisi yang terlihat pada. Sedangkan data primer disini yang termasuk digunakan dalam aplikasi tujuan. Adapun data yang termasuk dalam data residual pada Tabel 2.

Tabel 2. Data Residual

Table	Kolom	Fungsi yang berhubungan	Jenis data
		Visitor_teacher_my_class_task.php:	
task	Percentage	- Detail(\$class_id) - Add () - Delete()	Data residual
user	skype	Admin_user.php: - Index()	Data residual

Selain ada data residual yang akan tidak digunakan dalam sistem tujuan ada juga data tambahan yang akan digunakan dalam merekayasa komponen manajemen seluruh user. Data tambahan tersebut berbentuk tabel view pada database tujuan nantinya.

### 3.3 Desain ulang basis data

Pada tahapan ini dilakukan desain ulang data dimana data yang tadi tidak terpakai lagi akan dihilangkan pada basis data tujuan. Data yang dihilangkan yaitu data percentage yang berada pada *table task*. Pada basis data tujuan juga menghilangkan beberapa table yang tidak digunakan lagi pada sistem tujuan nantinya yaitu *table class\_teor* dan *semester*. Tidak hanya menghilangkan beberapa table pada basis data tujuan juga menambahkan *table view* yang diberi nama *v\_user* yang telah dijelaskan pada fase sebelumnya yaitu klasifikasi data. Desain ulang data dilakukan juga pada beberapa table yang berkaitan dengan modul yang akan direkayasa ulang. Pada *table task* menambahkan kolom *final\_exam*. Dimana kolom ini akan menandakan bahwa penilaian ini termasuk dalam penilaian ujian praktikum atau penilaian tugas praktikum biasa.

### 3.4 Mengadaptasi program warisan

Dalam proses adaptasi program warisan ke basis data tujuan, terdapat analisis fungsi pengaksesan data pada sistem warisan. Proses ini mencari manakah program warisan yang berkaitan dengan data yang direkayasa ulang dan mengadaptasikan fungsi baru untuk mengakses basis data tujuan.

Pada aplikasi I-Lab sendiri fungsi baru akan mengadaptasikan dari program warisan. Fungsi-fungsi ini akan diletakkan kedalam modul lama maupun modul baru. Adapun beberapa fungsi baru seperti pada Tabel 4.

Tabel 3. Fungsi baru

No.	Nama Fungsi	Fungsi	Tabel yang berhubungan
1	Add_new()	<ul style="list-style-type: none"> <li>Membuat penugasan</li> <li>Membuat penilaian ujian praktikum</li> <li>Membuat nilai tambahan</li> </ul>	Task Task_detail
2	Change_percentage_new (\$class_id)	Mengubah persentase penilaian akhir praktikum	Class Task Task_detail Presence Presence_detail
3	ajax_schedule()	Menampilkan jadwal praktikum pada hari ini pada dashboard admin	Class

Berikut implementasi dari adaptasi sistem warisan tiap komponen yang direkayasa ulang:

#### a) Komponen penilaian ujian praktikum

Mengadaptasi fungsi *add()* pada modul *visitor\_teacher\_my\_class\_task* kedalam fungsi baru yang diberikan nama *add\_new()* seperti pada tabel 2. Pada fungsi baru menambahkan opsi penilaian ujian praktikum, penilaian tambahan atau ekstra, dan penilaian tugas praktikum. Pada fungsi baru juga menambahkan persyaratan data not null pada form isian kolom *title*.

#### b) Komponen penilaian akhir praktikum

Dalam penilaian akhir praktikum mengadaptasi program warisan pada modul *visitor\_teacher\_my\_class\_final\_grade* dimana fungsi *change\_percentage()* digunakan dalam mengubah persentase penilaian akhir praktikum. Program dari fungsi *change\_percentage()* diadaptasi untuk digunakan kembali pada sistem tujuan. Pada modul yang sama yaitu fungsi *get\_count\_task\_each\_student()* juga diperbarui dalam mengambil data nilai ujian praktikum, nilai ekstra dan nilai modul praktikum.

#### c) Komponen penjadwalan praktikum

Pada komponen penjadwalan terdapat dua modul yang dirubah yaitu pada modul *admin\_dashboard* dan *admin\_schedule*. Pada *admin\_dashboard* pada fungsi *ajax\_schedule()* di

adaptasi kedalam sistem yang baru. Kemudian pada modul `admin_schedule` pada tampilan penjadwalan admin juga diadaptasi pada sistem tujuan.

d) **Komponen manajemen user**

Pada komponen ini berkaitan dengan modul `admin_user` dimana permasalahan dari modul ini terlalu lama dalam menampilkan data seluruh user. Oleh karena itu memutuskan untuk membuat modul baru dengan nama `admin_user_fix` tapi masih mengambil beberapa fungsi dari sistem warisan. pada modul `admin_user` pada fungsi `index()` untuk menampilkan user diadaptasi pada sistem tujuan.

### 3.5 Perpindahan data

Proses perpindahan data hanya mengekspor data per-table dari semua tabel yang digunakan kembali pada sistem tujuan. Kemudian setelah di ekspor data di import kedalam database tujuan dengan nama `ilab_new`. Dalam penelitian ini menggunakan jenis basis data yang sama yaitu `mysql`.

### 3.6 Desain ulang fungsi

Adapun pada studi kasus aplikasi I-Lab terdapat beberapa proses desain ulang fungsi yang dilakukan yaitu sebagai berikut:

a) **Desain ulang fungsi pembuatan task**

Pembuatan kategori penilaian ujian praktikum dan menambahkan atribut `final_exam`. Melakukan rekayasa ulang terhadap prosedur penambahan penilaian tugas praktikum dengan memberi opsional penilaian ujian praktikum, nilai tambahan dan penilaian tugas praktikum biasa.

b) **Desain ulang fungsi proporsi penilaian akhir praktikum**

Menambahkan persentase tambahan yaitu ujian praktikum dan nilai tambahan. Dalam perhitungan nilai akhir nilai dari akhir dari kegiatan praktikum yang meliputi nilai tugas modul dan presensi kehadiran ditambahkan dengan nilai ujian praktikum dan nilai tambahan jika ada.

c) **Desain ulang fungsi jadwal praktikum**

Penambahan fungsi baru pada jadwal praktikum di dashboard admin. Dimana fungsi tersebut dipergunakan untuk menentukan jadwal praktikum minggu pertama atau kedua.

d) **Desain ulang fungsi manajemen user**

Perubahan algoritma datatable dalam menampilkan data seluruh user. Dengan menggunakan metode `serverside datatable`. Fungsi yang telah didesain ulang menggunakan `serverside datatable` akan diletakkan pada modul baru dengan nama `admin_user_fix`. Adapun fungsi-fungsi baru dalam modul ini diantaranya `index()`, `get_totalrecord()`, `get_total()`, `get_data()`, `buil_query()`, `get_query()`, `search_or()`, dan `order_by()`.

### 3.7 Tes Ekuivalensi

Dalam hal ini komponen-komponen yang direkayasa ulang diletakkan pada file yang sama dengan komponen-komponen dari aplikasi lama. Kemudian apabila hasil dari komponen yang telah direkayasa ulang memiliki kesamaan dengan komponen lama dari aplikasi lama maka tes ekuivalensi dinyatakan berhasil. Dan apabila hasil dari komponen sama maka komponen berhasil direkayasa ulang. Berikut merupakan hasil dari tes ekuivalensi tiap komponen.

**a. Komponen penilaian ujian praktikum**

Hasil dari tes ekuivalensi dilakukan pada fungsi menambahkan penilaian tugas dengan opsional memilih *final exam*, *atau extra* *atau daily task* dengan kondisi kolom isian hanya judul saja yang diisi dan juga kondisi isian kolom judul dan deskripsi hasilnya fungsi berhasil menambahkan penilaian tugas dan muncul notifikasi berhasil dan dapat menambahkan nilai pada mahasiswa yang mengikuti praktikum tersebut serta dapat menghapus penilaian tersebut. Dengan hasil tersebut komponen baru dinyatakan sesuai dengan komponen sistem warisan.

**b. Komponen penilaian akhir praktikum**

Dari hasil uji coba mengganti persentase dengan menginputkan nilai presentase hasil yang keluar sesuai dengan yang diharapkan. Dan hasil keluaran nilai akhir sesuai dengan rumus penilaian yang baru. Dengan pegujian yang telah dilakukan hasilnya sesuai dengan komponen sistem warisan dan dinyatakan valid.

### c. Komponen penjadwalan praktikum

Dari hasil uji coba dengan menampilkan jadwal praktikum pada dashboard admin menghasilkan keluaran yang sama berupa jadwal yang sedang aktif pada hari yang sesuai. Pada menu admin juga sama-sama menampilkan jadwal praktikum. dengan hasil tersebut maka komponen baru sesuai dengan sistem warisan.

### d. Komponen manajemen user

Dari hasil uji coba menampilkan data *user* dihasilkan data yang sama dengan keluaran fungsi yang lama berupa data seluruh *user*. Dengan demikian komponen baru sesuai dengan komponen sistem warisan.

## 3.8 Membersihkan data residual

Data yang dibersihkan sesuai dengan tahap klasifikasi data dimana yang tidak digunakan kembali akan dihapus. Pada *table task* setelah dilakukan perpindahan ke database baru kemudian dilakukan penghapusan *attribute percentage*. Hal yang sama juga dilakukan pada *table user* menghilangkan *attribute skype*.

## 4. Kesimpulan

Adapun kesimpulan dari hasil penelitian yang telah dilakukan sebagai berikut:

1. Rekayasa ulang dapat dilakukan sedikit demi sedikit dimana dilakukan pada komponen-komponen kecil dan dilakukan secara bertahap dengan proses yang telah dijelaskan pada komponen penilaian akhir praktikum, penjadwalan praktikum, dan manajemen seluruh user serta dilakukan penambahan komponen penilaian ujian praktikum.
2. Hasil dari pengujian ekivalensi menunjukkan bahwa tingkat kesesuaian antara komponen pada sistem warisan dan sistem tujuan berjalan sesuai 100%.
3. Pada penelitian selanjutnya diharapkan dapat melanjutkan rekayasa ulang iteratif secara menyeluruh pada aplikasi I-Lab. Kemudian diharapkan dapat menerapkan metode rekayasa ulang lain seperti *chicken little strategy* maupun *butterfly methodology*.

## Referensi

- [1] D. Lawless, B. Wu, J. Bisbal, and J. Grimson, "Legacy information systems: Issues and directions," *IEEE*, vol. 16, no. 5, pp. 103–111, 1999.
- [2] S. Rochimah, C. Zain, and D. H. Murti, "Penerapan Rekayasa Ulang Iteratif Pada Sistem Informasi Akademik Ftif Its Surabaya," *JUTI J. Ilm. Teknol. Inf.*, vol. 5, no. 1, p. 19, 2006.
- [3] I. Sommerville, *Software Engineering*. 2011.
- [4] R. S. Pressman, *Software Engineering*. 2010.
- [5] A. S. Abbas, W. Jeberson, and V. V. Klinsega, "The Need of Re-engineering in Software Engineering," vol. 2, no. 2, pp. 292–295, 2012.
- [6] A. Bianchi and I. C. Society, "Iterative Reengineering of Legacy Systems," vol. 29, no. 3, pp. 225–241, 2003.
- [7] W. A. N. Putri and I. M. Suartana, "Rekayasa Ulang Sistem Informasi Manajemen Kepegawaian Menggunakan Framework Laravel Pada Bppki Surabaya," vol. 6, pp. 82–90, 2016.
- [8] M. Model, K. Iso, A. B. Raharjo, U. L. Yuhana, and S. Rochimah, "Rekayasa Ulang SIM Akademik ITS Berdasarkan Karakteristik Pemeliharaan," vol. 2, no. 1, 2013.
- [9] S. B. Utomo and S. D. Ariyanto, "Rekayasa Ulang Sistem Informasi Akademik Politeknik Manufaktur ABC Dalam Jaringan Berbasis Web Internet," 2017.
- [10] Wahyuni, "Rekayasa Ulang (Reengineering) Sistem Informasi Manajemen Pertanahan Nasional Dengan Pendekatan Unified Modelling Language (UML)," *Bhumi*, vol. 3, no. April, 2017.

