

# Software Redocumentation untuk Mendukung Pemeliharaan Sistem Informasi Terpadu Universitas Pasundan (SITU)

Ayi Purbasari

Program Studi Teknik Informatika, Universitas Pasundan,  
Bandung, Indonesia  
pbasari@unpas.ac.id

## Abstrak

Penelitian terdahulu memperlihatkan bahwa hampir setengah usaha pemeliharaan dihabiskan untuk memahami aplikasi itu sendiri. Diperlukan pemahaman yang menyeluruh terkait aplikasi, dimana jika dituangkan dalam bentuk teks source code, akan lebih menyulitkan pemahaman. Oleh karena itu, kunci keberhasilan sebuah kegiatan pemeliharaan adalah adanya dokumentasi yang baik terkait aplikasi yang sedang digunakan. Software redocumentation adalah salah satu pendekatan yang digunakan sebagai alat bantu untuk memahami program dalam rangka pemeliharaan dan evolusi perangkat lunak. Penelitian ini akan mendefinisikan bagaimana melakukan tahapan redocumentation Sistem Informasi Terpadu Universitas Pasundan (SITU Unpas) dalam rangka mendukung kegiatan pemeliharaan perangkat lunak. Secara umum, tahapan kegiatan software redocumentation ini terdiri dari System Inventory, System Assessment, Redocumentation Planning dan Redocumentation itu sendiri yang terdiri dari empat kegiatan rinci. Hasil dari penelitian ini adalah redocument SITU, khususnya bagian proses bisnis dan requirement, serta arsitektur sistem dan desain database. Hasil penelitian berupa dokumentasi teknis perangkat lunak sesuai dengan tahapan redocumentation dalam bentuk high level view definition, yaitu view level tinggi dari sistem, berupa fungsionalitas sistem dan bagaimana sistem ini berinteraksi dengan sistem lain. Berdasarkan high level view definition ini, terlihat bahwa secara proses bisnis dan fungsionalitas sistem, sudah terdapat kesesuaian, sistem berpeluang untuk dikembangkan. Sedangkan tahapan cross references extraction, yaitu aktivitas mengidentifikasi referensi silang dari rutin ke rutin (call graph), rutin ke data (tabel CRUD), data ke data (data model), serta fungsionaliti ke rutin, akan menjadi penelitian berikutnya. **Kata kunci:** maintenance, software maintenance, pemeliharaan perangkat lunak, redocumentation, Sistem Informasi Terpadu Universitas Pasundan (SITU).

## 1. Pendahuluan

Dalam sebuah siklus pengembangan aplikasi, terdapat tahapan penting setelah aplikasi dideliver ke pengguna, yaitu tahapan pemeliharaan. SWEBOK, Software Engineering Body of Knowledge, menempatkan pemeliharaan sebagai kajian tersendiri, setara dengan software development (SWEBOK, 2003). Pemeliharaan diperlukan untuk menjamin bahwa perangkat lunak terus menerus memenuhi kebutuhan pengguna. Produk perangkat lunak pada dasarnya akan berubah seiring dengan adanya proses korektif atau non-korektif. Pembiayaan pemeliharaan bergantung dari kategori pemeliharaan itu sendiri, apakah perbaikan kode program ataupun perubahan perancangan aplikasi, atau bahkan perluasan dari aplikasi. Karena itu, pemeliharaan merupakan kegiatan penting dari pengoperasian sebuah sistem aplikasi.

Pemeliharaan itu sendiri melibatkan berbagai issue, yaitu technical issues, management issues, cost estimation, dan measurement. Pada technical issue, persoalan penting adalah terbatasnya pemahaman (limited understanding) dimana adanya keterbatasan seorang software engineer memahami sebelah mana diperlukan perubahan atau perbaikan. Riset memperlihatkan bahwa hampir setengah usaha pemeliharaan dihabiskan untuk memahami aplikasi itu sendiri. Diperlukan pemahaman yang menyeluruh terkait aplikasi, dimana jika dituangkan dalam bentuk teks source code, akan lebih menyulitkan pemahaman. Oleh karena itu, kunci keberhasilan sebuah kegiatan pemeliharaan adalah adanya dokumentasi yang baik terkait aplikasi yang sedang digunakan.

SITU, Sistem Informasi Terpadu Unpas, yang merupakan Sistem Informasi Akademik yang digunakan di lingkungan Universitas Pasundan, terus mengalami perubahan dan perluasan seiring dengan tuntutan kebutuhan pengguna yang semakin meluas (Unpas, 2017). Tim pengembang yang sudah bekerja dengan baik, dirasa lebih baik jika dilengkapi dengan berbagai dokumen internal sistem sebagai pelengkap aplikasi. Di lain sisi, menurut Pressman perangkat lunak itu sendiri adalah program yang dilengkapi database dan dilengkapi dengan dokumentasi teknis. Dokumentasi terdiri dari dokumentasi pengembangan, yaitu dimulai dengan proses bisnis, daftar user requirement (kebutuhan pengguna), daftar software requirement, termasuk juga dokumentasi hasil analisis dan desain, terutama desain database. Serta dokumentasi terkait versioning aplikasi. Selain itu, dokumentasi terkait penggunaan aplikasi juga sangat diperlukan untuk kegiatan pemeliharaan.

Bedasarkan hal tersebut di atas, maka dirumuskan konsep penelitian yaitu

P1: Pentingnya melakukan pemeliharaan

P2: Pemeliharaan dimudahkan jika aplikasi dilengkapi dengan dokumentasi teknis yang baik

Dapat ditarik hipotesis bahwa SITU dilengkapi dengan dokumentasi yang baik, maka dapat mempermudah proses pemeliharaan. Dengan demikian, masalah penelitian ini adalah bagaimana melakukan tahapan redocumenting SITU dalam rangka mendukung kegiatan pemeliharaan perangkat lunak. Penelitian ini bertujuan untuk membangun dokumentasi teknis SITU, yang terdiri dari dokumentasi proses bisnis, dokumentasi requirement, dokumentasi model analisis, dan dokumentasi model desain.

Secara sistematis, makalah ini terdiri dari pendahuluan, material, metode penelitian, hasil dan diskusi, serta kesimpulan.

## 2. Software Redocumentation

### 2.1. Software Documentation

Software documentation atau dokumentasi perangkat lunak merupakan dokumen yang ditulis dalam bentuk teks atau ilustrasi / diagram yang melengkapi produk perangkat lunak. Dokumen ini menjelaskan bagaimana sistem diperasikan atau bagaimana sistem ini digunakan, dan akan berbeda untuk setiap pengguna yang berbeda perannya. Dokumen juga dapat berupa dokumentasi proses untuk setiap tahapan pengembangan perangkat lunak. Karena itu, dokumen perangkat lunak dapat berupa:

- 1) Dokumentasi requirement – menjelaskan apa saja yang user butuhkan terhadap sistem (user requirement) dan bagaimana sistem mengakomodir kebutuhan user tersebut (software requirement).
- 2) Dokumentasi desain / arsitektur – merupakan gambaran dari perangkat lunak. Termasuk juga di dalamnya lingkungan dan prinsip-prinsip konstruksi yang digunakan dalam perancangan perangkat lunak.
- 3) Dokumentasi teknis – berisi dokumentasi kode program, algoritma, interface dan API yang digunakan.
- 4) Dokumentasi End user – berupa manual untuk end user, system administration dan support staf.
- 5) Dokumentasi Marketing – Bagaimana memasarkan produk termasuk juga analisis kebutuhan pasar.

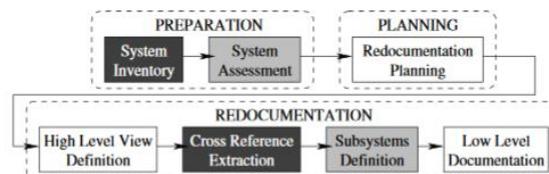
Keseluruhan dokumen umumnya mengacu kepada standar dokumen tertentu, misalkan IEEE. Untuk dokumentasi requirement, dapat digunakan standar dokumen SKPL (spesifikasi kebutuhan perangkat lunak) atau yang disebut SRS (Software Requirement Specification). Sedangkan untuk desain, digunakan SDD (Software Design Description).

### 2.2. Software Redocumentation

Software redocumentation adalah salah satu pendekatan yang digunakan sebagai alat bantu untuk memahami program dalam rangka pemeliharaan dan evolusi perangkat lunak. Elliot, JC dkk menyatakan: —Redocumentation is a creation or revision of a semantically equivalent representation within the same relative abstraction level (Nallusamy, Ibrahim, & Mahrin, 2011). Dengan kata lain, redocumenting kode adalah transformasi dari satu eksistensi (dan dokumen lain atau pengetahuan dari para stakeholder) ke dalam sebuah dokumen yang update. Ini akan menjadi alat bantu untuk proses recovery dan pemahaman software secara menyeluruh. Software comprehension adalah bagian termahal dari software maintenance, maka redocumentation merupakan kunci dari software maintainability.

Tujuan utama dari proses software redocumentation adalah sebagai alternatif view dari sistem untuk meningkatkan pemahaman. Misalkan mengenerate diagram use case dari source code. Selain itu, redocumentation digunakan juga untuk meningkatkan dokumentasi itu sendiri. Idealnya, dokumentasi harusnya diproduksi pada saat pengembangan sistem dan terupdate ketika sistem berubah. Hal tersebut masih belum banyak dilakukan. Ketika, pendokumentasian untuk program yang termodifikasi. Ini untuk memfasilitasi kegiatan pemeliharaan di kemudian hari, terutama jika akan dilakukan preventive maintenance. Karena itu, pemahaman tentang sistem perangkat lunak oleh seorang software maintainer merupakan hal penting dalam sebuah evolusi perangkat lunak. Namun, orang yang terlibat dalam software maintainance umumnya tidak berpartisipasi dalam pembangunan sebuah sistem. Maintainer tidak memiliki pengetahuan yang cukup tentang sistem dan menjadi tidak efektif dan efisien.

Berikut Gambar 1 tentang proses software redocumentation :



Gambar 1 Proses Redocumentation Software redocumentation process [1]

### 2.3. Penelitian Terdahulu

Yang & Chen [1] mengkaji redocumentation legacy system dan mengusulkan pendekatan berorientasi model untuk mengenerate dokumentasi. Pendekatan ini memproduksi model dari sistem eksisting dan mengenerate dokumentasi dari model tersebut.

Kienle & Muller [2] menjelaskan lingkungan Rigi, yang merupakan kakas penelitian yang sudah mature, digunakan untuk para reverse engineer sistem perangkat lunak. Dengan menggunakan Rigi, sistem yang besar dapat dianalisis, eksplor secara interaktif, diringkas dan didokumentasikan. Makalah ini menjelaskan komponen utama Rigi dan fungsi-fungsinya dan mengassess dampaknya pada kegiatan reverse engineering. Nallusamy, Ibrahim, & Mahrin [3] mendeskripsikan framework untuk meredokumentasikan legacy source code dan komponen perangkat lunak terkait lainnya untuk mendukung pemeliharaan perangkat lunak. Framework ini generate ontology untuk menghane pemahaman program dalam level semantik. Proses redocumentation itu sendiri dilakukan semi otomatis menggunakan Software Work Products (SWP), memetakan metadata dan menghasilkan ontologinya. Ontologi ini akhirnya dipetakan ke dalam standard dokumen perangkat lunak, dalam hal ini MIL-STD-498.

Sugumaran & Ibrahim [4] mendeskripsikan evaluasi dari dokumentasi perangkat lunak yang menggunakan pendekatan dan kakas redocumentation. menjelaskan komponen utama dari software redocumentation . Beberapa pendekatan dan kakas redocumentation dijelaskan pada makalah ini. Pada akhirnya, malah ini menghasilkan identifikasi beberapa aspek dari DQA yang berguna untuk perbaikan.

Van Geet, Ebraert, & Demeyer [5] menjelaskan pengalaman dalam mengaplikasikan teknik redocumentation dalam industri. Menjelaskan bagian apa yang perlu atau tidak perlu didokumentasikan dan bagian apa yang perlu atau tidak perlu diotomatisasikan, dan bagaimana mengotomatisasikannya.

Su & Yeh [6] mengusulkan sebuah metode yang dilakukan oleh pusat kesehatan yang terletak di selatan Taiwan untuk memulihkan arsitektur perangkat lunak dan membangun kembali dokumentasi Sistem Informasi Rumah Sakit. Arsitektur dan dokumentasi akan berfungsi sebagai dasar dari migrasi aplikasi SIRS ke arsitektur berbasis komponen. Penelitian dimulai dengan mendefinisikan arsitektur perangkat lunak dengan mempelajari dokumen sistem dan wawancara analis software. Informasi arsitektur kemudian dilengkapi dengan menganalisis kode sumber untuk mengatasi masalah dokumen sistem yang salah. Informasi arsitektur yang telah betul kemudian disimpan ke dalam repositori secara top-down. Setelah rekonstruksi arsitektur, pengembang perangkat lunak kemudian memberikan deskripsi tambahan untuk melengkapi dokumentasi.

Torchiano, Ricca, & Tonella [7] memfokuskan kepada dua pendekatan yaitu menggunakan grafis ataupun anotasi dari kode sumber. Dalam kasus pertama, diagram diproduksi secara interaktif, mulai dari informasi rekayasa terbalik. Dalam kasus kedua, informasi desain ditambahkan dalam bentuk penjelasan kode. Diagram dapat dihasilkan, jika diperlukan, oleh alat anotasi pengolahan, yang menafsirkan penjelasan sebelumnya dimasukkan ke dalam kode dan menghasilkan pandangan grafis. Hasil penelitian menunjukkan bahwa pendekatan pertama lebih banyak disukai dan terasa bermanfaat.

Wang, Lai, & Liu [8] melakukan penelitian lebih kepada membangun link traceability antara dokumentasi dan kode sumber yang dapat sangat membantu untuk manajemen rekayasa perangkat lunak, seperti persyaratan traceability, analisis dampak, dan penggunaan kembali software. Terdapat empat strategi berdasarkan karakteristik khusus dari dokumentasi dan kode sumber, yaitu, kode sumber clustering, identifier mengklasifikasikan, tesaurus kesamaan, dan peningkatan struktur hirarkis. Hasil penelitian menunjukkan bahwa tiga strategi pertama dapat meningkatkan ketepatan link secara lebih baik.

### **3. Metode Penelitian**

Penelitian ini akan menerapkan model redocumentation yang tercantum pada Gambar 1 di atas. Berikut adalah tahapan penelitian erdiri dari:

- Preparation: System Inventory, System Assessment:
- Planning
- Redocumentation
  - o High Level View Definition;
  - o Cross References Extraction;
  - o Subsystems Definition;
  - o Low Level Documentation.

## **4. Hasil dan Pembahasan**

### **4.1 Preparation**

Pada kegiatan ini dilakukan system inventory dan assessment. Tujuan dari aktivitas system inventory ini adalah untuk mendapat gagasan terkait besarnya permasalahan dan informasi dasar yang dibutuhkan oleh sistem berikutnya.

#### **4.1.1 System Inventory**

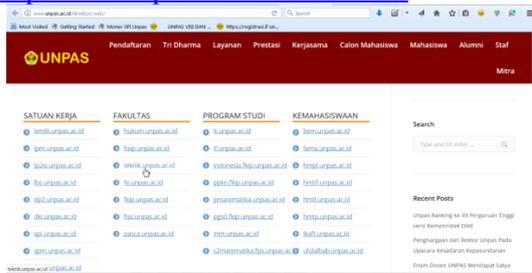
Umumnya, pertanyaan mendasar pada kegiatan ini berupa:

What exactly constitutes the system? Apa sebenarnya yang merupakan sistem? SITU, Sistem Informasi Terpadu Unpas, yang merupakan Sistem Informasi Akademik yang digunakan di lingkungan Universitas Pasundan, terus mengalami perubahan dan perluasan seiring dengan tuntutan kebutuhan pengguna yang semakin meluas [10].

Where to find these informations and new ones?

SITU merupakan aplikasi berbasis web yang dapat diakses melalui internet melalui beberapa akses:

1) Melalui website utama Unpas  
<http://www.unpas.ac.id/direktori-web/>



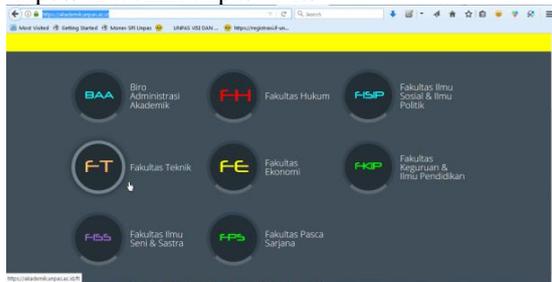
Gambar 1 Akses via website utama Unpas

2) Melalui website Fakultas Teknik  
<http://teknik.unpas.ac.id/>



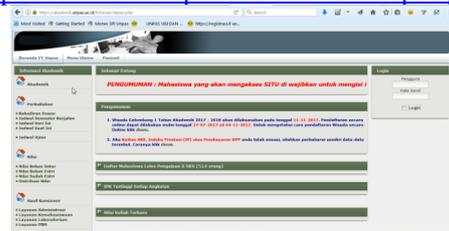
Gambar 2 Akses via website Fakultas Teknik

3) Melalui website Akademik Unpas  
<https://akademik.unpas.ac.id/>



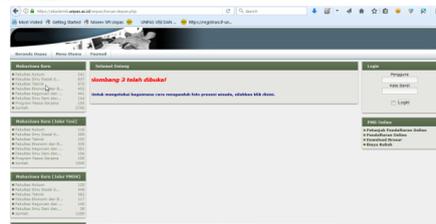
Gambar 3 Akses via website akademik Unpas

4) Langsung ke url SITU Unpas  
<https://akademik.unpas.ac.id/ft/besan.depan.php>



Gambar 4 Akses langsung ke SITU Unpas

Atau ke SITU untuk bagian akademik:



Gambar 5 Akses langsung ke FT Unpas

What is known about it?

SITU berisi informasi terkait Tridarma Perguruan Tinggi. Terdiri dari Menu utama dan menu pengumuman. Dilengkapi dengan fasilitas log in dan PMB. Khusus PMB ini hanya ada di SITU bagian akademik Unpas, tidak ada di bagian Fakultas.

#### 4.1.2 System Assessment

Tahapan kedua ini merupakan assesment terhadap kode, dokumentasi dan sumber informasi lain. Digunakan untuk merencanakan kegiatan redocumenation itu sendiri dan pemeliharaan secara umum.

- Kode program: tersedia, menggunakan bahasa pemrograman PHP
- Basis data: tersedia, menggunakan MySQL
- Dokumentasi teknis SITU: tidak ada
- Sumber informas lain: Dokumen Prosedur Mutu FT Unpas.

#### 4.2 Redocumenation

##### 4.2.1 High Level View Definition

Pada bab ini akan digambarkan hasil sementara tahapan High Level View Definition. Terdiri dari: gambaran proses bisnis sistem, gambaran fungsionalitas sistem dan gambaran database.

##### 4.2.1.1 Proses Bisnis Sistem

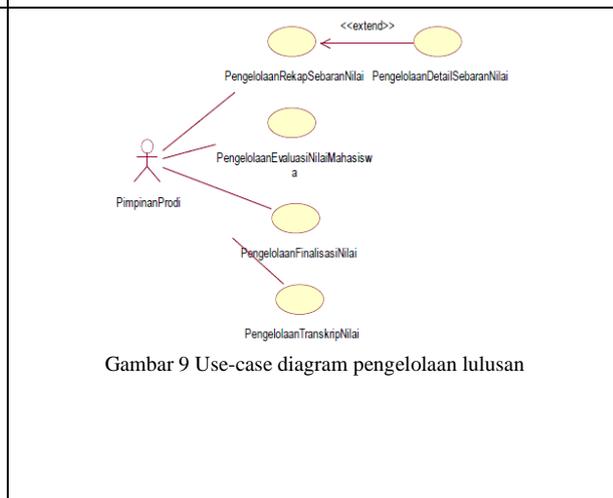
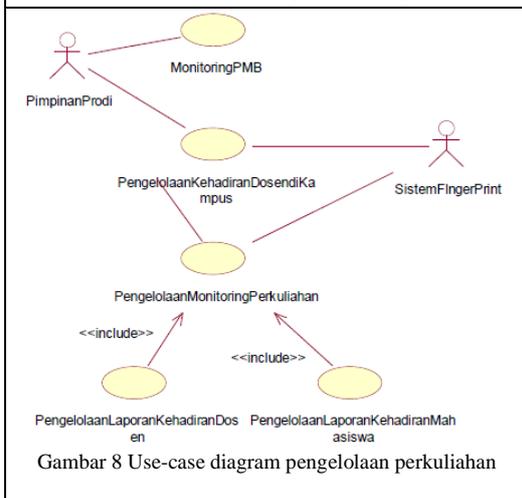
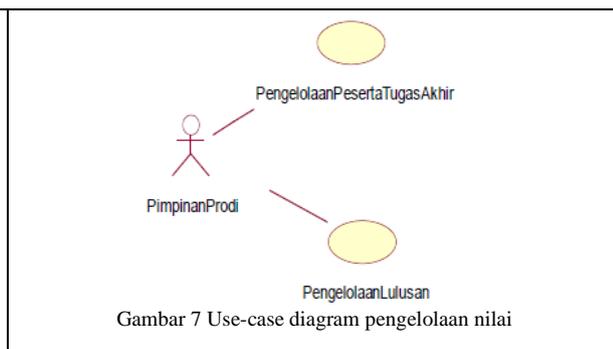
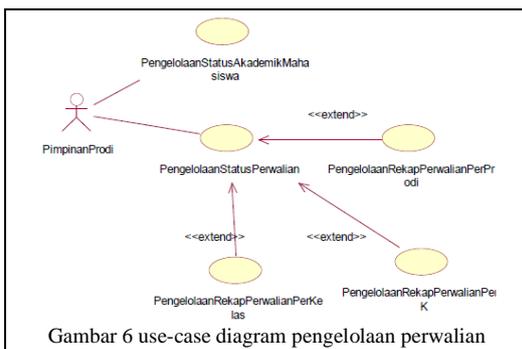
Proses bisnis akademik di lingkungan Fakultas Teknik Universitas Pasundan terdiri dari 67 SOP yang telah didokumentasikan. Dimulai dengan monitoring dan evaluasi kurikulum, sampai dengan SOP penerimaan mahasiswa baru. Dokumen SAP telah disusun oleh institusi. Berikut rangkuman Prosedur. Dari 67 SOP tersebut, terdapat 10 SOP yang bersesuaian dengan fitur atau fungsionalitas yang terdapat pada Aplikasi SITU. Yaitu terangkum pada Tabel 1 berikut ini:

Tabel 1 Daftar prosedur SITU Unpas

No	Daftar Prosedur
6	FT-PM-00.02 Pelaksanaan UAS Dan UTS
7	FT-PM-05.01-2 Penyerahan Berkasi Nilai Mahasiswa
8	FT-PM-05.03 Pendaftaran Tugas Akhir
9	FT-PM-05.04 Pengisian Kuesioner & Cetak Kartu Ujian
10	FT-PM-05.05 Penyusunan Jadwal Perkuliahan
27	FT-PM-08.01 Bimbingan Akademik dan Konseling
28	FT-PM-08.02 Proses Pembelajaran
29	FT-PM-08.03 Ujian (UTS & UAS)
30	FT-PM-08.04 Pengelolaan Tugas Akhir
40	FT-PM-16.01 Registrasi

#### 4.2.2 Fungsional Sistem

Terdapat banyak pengguna dalam SITU. Dalam penelitian ini, pengguna / actor SITU dibatasi dengan pimpinan prodi. Gambaran fungsionalitas sistem akan dituangkan dalam diagram Use-case pada Gambar 6, Gambar 8, Gambar 7, dan Gambar 9 sebagai berikut:



Berdasarkan gambaran fungsionalitas sistem di atas, dan dibandingkan dengan daftar proses bisnis yang sudah disiapkan oleh Fakultas Teknik, terlihat bahwa sudah sebagian proses bisnis untuk pimpinan prodi sudah diimplementasikan di SITU. Masih terdapat banyak proses bisnis dan fungsionalitas yang dapat dikembangkan sesuai dengan kebutuhan.

#### 4.2.3 Database SITU

Terdapat 624 table yang diidentifikasi pada kegiatan pedahuluan, terdapat 492 tabel yang digunakan pada SITU, belum termasuk view dan stored procedure. Penamaan table sudah baik, menetapkan konvensi. Terdapat kelompok utama table yaitu: besan, jalak, borang, email, feeder, dan simak. Terangkum pada Tabel 2 berikut ini:

Tabel 2 Gambaran database SITU

No	ID Tabel	Jumlah Tabel	Keterangan
1	Besan	48	Tabel SITU untuk antarmuka
2	Borang	19	Tabel SITU untuk temporary keperluan akreditasi
3	Email	6	Tabel SITU untuk email gateway
4	Jalak	150	Tabel SITU untuk migrasi dari tabel SIMAK (dalam pembangunan)
5	KP TA	13	Tabel SITU untuk kegiatan KP TA
6	Sarana	1	Tabel SITU untuk pengelolaan sarana, masih dibangun
7	SDM	13	Tabel SITU untuk pencatatan SDM
8	SIMAK	211	Tabel SITU untuk seluruh kegiatan akademik
9	SMS	7	Tabel SITU untuk sms gateway
10	WP	24	Tabel SITU untuk keperluan wordpress

#### 4. Simpulan

Berikut beberapa kesimpulan yang bisa diambil:

1) Redocumentation dilakukan pada high level view definition, yaitu view level tinggi dari sistem, berupa fungsionalitas sitem dan bagaimana sistem ini berinteraksi dengan sistem lain.

2) Kendala yang dihadapi adalah tidak adanya dokumentasi teknis sama sekali, SOP yang belum dimapping ke sistem aplikasi, serta kompleksitas sistem yang sudah membesar; terlihat dari struktur menu serta banyaknya table yang digunakan..

3) Berdasarkan gambaran fungsionalitas sistem di atas, dan dibandingkan dengan daftar proses bisnis yang sudah disiapkan oleh Fakultas Teknik, terlihat bahwa sudah sebagian proses bisnis untuk pimpinan prodi sudah diimplementasikan di SITU. Masih terdapat banyak proses bisnis dan fungsionalitas yang dapat dikembangkan sesuai dengan kebutuhan.

Berikut beberapa pengembangan lebih lanjut:

1) Terdapat fungsionalitas spesifik yang diandalkan oleh SITU. Yaitu, sinkronisasi keuangan dan pewalihan. Terdapat 5 model data yang menggambarkan fungsionalitas perwalihan. Kelima model data tersebut dapat ditinjau kembali untuk proses normalisasi.

2) Beberapa pekerjaan yang bisa dilakukan berikutnya adalah:

- Mapping SOP dan fungsionalitas sistem untuk fungsionalitas lainnya
- Melengkapi gambaran fungsionalitas sistem

3) Tahapan cross references extraction, yaitu aktivitas mengidentifikasi referensi silang dari rutin ke rutin (call graph), rutin ke data (tabel CRUD), data ke data (data model), serta fungsionaliti ke rutin, akan menjadi penelitian berikutnya.

#### Daftar Pustaka

- [1] Nicolas Anquetil and de Oliveira, "Software Re-Documentation Process and Tool," in CAiSE Short Paper Proceedings., 2005.
- [2] Hongji Yang and Feng Chen, "Model oriented evolutionary redocumentation," in 31st Annual International Computer Software and Applications Conference, vol. 1, 2007, pp. 543--548.
- [3] Holger M Kienle and Hausi A Muller , "Rigi—An environment for software reverse engineering, exploration, visualization, and redocumentation," Science of Computer Programming, vol. 75, no. 4, pp. 247--263, 2010.
- [4] S Nallusamy, S Ibrahim, and M. N. R. Mahrin, "A Software redocumentation process using ontology based approach in software maintenance," International Journal of Information and Electronics Engineering, vol. 1, no. 2, p. 133, 2011.
- [5] N Sugumaran and S Ibrahim, "An Evaluation on Software redocumentation Approaches and Tools in Software Maintenance," Communications of the IBIMA, 2011.
- [6] Joris Van Geet, Peter Ebraert, and Se Demeyer, Redocumentation of a Legacy Banking System., 2010.
- [7] C. T. Su and D Yeh, "Software architecture recovery and redocumentation tool of a hospital information system," in International Conference on Computer and Communication Engineering, 2012, pp. 143-146.
- [8] M Torchiano, F Ricca, and P Tonella, "Empirical comparison of graphical and annotation-based redocumentation approaches," IET Software, pp. 15-31, 2010.
- [9] X Wang, G Lai, and C Liu, "Recovering relationships between documentation and source code based on the characteristics of software engineering," vol. 243, pp. 121-137, 2009.
- [10] SPTIK Unpas. (2017) SITU Unpas. [Online]. <https://akademik.unpas.ac.id/>
- [11] A Tang, P Liang, and H Van Vliet, "Software architecture documentation: The road ahead," in 9th Working IEEE/IFIP Conference on Software Architecture, 2011, pp. 252-255.
- [12] Software Engineering Body of Knowledge SWEBOK. (2003) SWEBOK, Software Engineering Body of Knowledge. [Online]. <https://www.computer.org/web/swebok>