

PENGEMBANGAN SISTEM INFORMASI KEMAJUAN AKADEMIK MENGUNAKAN MODEL *INCREMENTAL* BERBASIS EVALUASI *USABILITY* DAN *WHITE BOX TESTING*

I Made Agus Oka Gunawan¹, Gede Indrawan², Sariyasa³

^{1,2,3}Program Studi Magister Ilmu Komputer, Universitas Pendidikan Ganesha
Jl. Udayana No. 11, Singaraja, Indonesia

e-mail: agusokagunawan@gmail.com¹, gindrawan@undiksha.ac.id², sariyasa@undiksha.ac.id³

Received : August, 2020

Accepted : December, 2020

Published : April, 2021

Abstract

This research aims to develop an Academic Progress Information System (SIsKA) in the Computer Science Study Program, Postgraduate Program at Ganesha University of Education. The incremental development model is used in the development of SIsKA. The evaluation stage is carried out using the Focus Group Discussions technique to validate user input for the development stage. Evaluation was carried out on 20 respondents, who were selected using simple random sampling technique from active SIsKA users. Testing of the SIsKA development code is also carried out using White Box Testing. The results of the evaluation of phase 1 and phase 2 of development succeeded in providing recommendations for improvements according to user needs. The evaluation results from phase 3 of development show that SIsKA is in accordance with user needs. The results of the White Box test show that all lecturer page functions are at a simple level. Future development and evaluation can focus on improving the complexity of the 14 functions on the administrator and student pages, as well as analyzing data security related to SIsKA.

Keywords: *development, evaluation, discussion, white box, incremental.*

Abstrak

Penelitian ini bertujuan mengembangkan Sistem Informasi Kemajuan Akademik (SIsKA) pada Program Studi Ilmu Komputer, Program Pascasarjana di Universitas Pendidikan Ganesha. Model pengembangan incremental digunakan dalam pengembangan SIsKA. Tahap evaluasi dilakukan dengan teknik Focus Group Discussions untuk memvalidasi masukan-masukan pengguna untuk tahap pengembangannya. Evaluasi dilakukan terhadap 20 responden, yang dipilih dengan menggunakan teknik simple random sampling dari pengguna aktif SIsKA. Pengujian kode hasil pengembangan SIsKA juga dilakukan menggunakan White Box Testing. Hasil evaluasi fase 1 dan fase 2 pengembangan berhasil memberikan rekomendasi perbaikan sesuai kebutuhan pengguna. Hasil evaluasi dari fase 3 pengembangan menunjukkan bahwa SIsKA sudah sesuai dengan kebutuhan pengguna. Hasil pengujian White Box menunjukkan seluruh fungsi halaman dosen ada pada tingkat sederhana. Pengembangan dan evaluasi kedepannya dapat difokuskan pada perbaikan kompleksitas 14 fungsi pada halaman administrator dan mahasiswa jada dapat dilakukan analisis terkait keamanan data pada SIsKA.

Kata Kunci: *pengembangan, evaluasi, diskusi, white box, incremental.*

1. PENDAHULUAN

Sistem Informasi Kemajuan Akademik (SIsKA) terbukti bisa membantu pengelola Program Studi Ilmu Komputer Program Pascasarjana Universitas Pendidikan Ganesha (Prodi Ilkom) dalam mengelola

data penelitian mahasiswa [1]. Penelitian terkait SIsKA berhasil menemukan permasalahan yang ada, diantaranya 1) fungsionalitas SIsKA yang belum maksimal [1] dan 2) antarmuka SIsKA yang perlu dilakukan perbaikan berdasarkan evaluasi heuristik

[2]. Hasil evaluasi heuristik menunjukkan terdapat permasalahan yang ditunjukkan pada Tabel 1 [2].

Tabel 1: Permasalahan Evaluasi Heuristik

No	Prinsip yang Dilanggar	Permasalahan
1	Visibilitas dari status sistem	Pemilihan warna pada tombol “hapus” dan “kembali” yang kurang jelas dan perbedaan aksi tombol yang kurang jelas.
2	Kesesuaian antara sistem dan dunia nyata	Penggunaan kata-kata pada sistem yang masih belum konsisten, bentuk simbol yang digunakan sebagai petunjuk kurang sesuai dengan kenyataan dan terdapat judul menu yang ambigu.
3	Kendali dan kebebasan pengguna	Fitur pencarian belum berjalan dengan baik. Pengguna juga terlalu dipaksakan untuk merubah data akunya.
4	Standar dan konsistensi	Penggunaan bahasa pada SIsKA yang tidak konsisten, penulisan nama menu yang tidak konsisten dan pemilihan judul menu yang masih ambigu.
5	Pencegahan kesalahan	Sistem belum mencegah pengguna melakukan kesalahan.
6	Bantu pengguna untuk mengenali, mendiagnosa, dan mengatasi masalah	Belum ada pesan peringatan kesalahan ketika pengguna membuat kesalahan.
7	Fleksibilitas dan efisiensi	Penempatan menu pencarian yang tidak tepat.
8	Estetika dan desain yang minimalis	Penempatan fitur pencarian yang tidak sesuai, penggunaan logo yang tidak berwarna, format tanggal dan waktu yang tidak rapi.
9	Fitur bantuan dan dokumentasi	Belum terdapat fitur bantuan pada sistem

Permasalahan fungsionalitas SIsKA yang belum maksimal juga berhasil ditemukan pada penelitian lainnya, dimana terdapat 2 kebutuhan fungsional SIsKA yang harus dipenuhi SIsKA, yaitu 1)

pengelolaan waktu akhir pengajuan dan 2) pengelolaan kebutuhan tambahan sebagai pendukung penjadwalan ujian mahasiswa seperti ujian paralel dan prasyarat [1]. Solusi terhadap permasalahan antarmuka dan fungsionalitas SIsKA ini adalah melakukan pengembangan yang berkelanjutan. Pengembangan secara berkelanjutan bertujuan agar penggunaan SIsKA lebih efektif, efisien, dan memuaskan penggunaannya [3].

Pada tahap pengembangan sebuah sistem, banyak model pengembangan yang bisa digunakan, diantaranya *waterfall*, *incremental* dan *prototyping* [4]. Model pengembangan *incremental* didasarkan pada masukan dari pengguna untuk selanjutnya dijadikan acuan pengembangan beberapa versi sistem sampai memenuhi kebutuhan pengguna [5]. Model *incremental* meminimalisir resiko cacat selama proses pengembangan karena setiap sistem dirilis akan dilakukan evaluasi secara bertahap [6]. Model pengembangan *incremental* memiliki 4 tahapan pada setiap fase pengembangannya (rilis), yaitu 1) analisis kebutuhan, 2) perancangan, 3) implementasi dan 4) evaluasi [5].

Tahapan analisis kebutuhan dilakukan dengan menspesifikasikan kebutuhan-kebutuhan terkait sistem yang dikembangkan. Tahap perancangan akan memodelkan analisis yang dihasilkan ke dalam sebuah model aliran data dengan *Data Flow Diagram* [7] dan basis data dengan *Entity Relationship Diagrams - crow's foot* [8]. Tahap implementasi dilakukan dengan menuliskan kode program dari perancangan yang dihasilkan ke dalam bentuk perintah-perintah yang dimengerti komputer dengan menggunakan bahasa pemrograman. Hasil akhir implementasi kemudian dilakukan evaluasi untuk menilai akhir sistem yang diimplementasikan. Pada model *incremental*, tahapan evaluasi memegang peranan penting untuk mendapatkan masukan pengguna dalam pengembangan lanjutan sistem [6], salah satu metode evaluasi berbasis pengguna yang dapat digunakan adalah evaluasi *usability*.

Evaluasi *usability* merupakan salah satu metode evaluasi sistem yang berfokus mengevaluasi seberapa baik pengguna dapat belajar dan menggunakan sistem, juga merujuk pada seberapa puas pengguna dengan proses-proses pada sistem [9]. Metode evaluasi *usability* dikategorikan menjadi 3 yaitu *inspection*, *testing* dan *inquiry* [10]. *Focus Group Discussion* (FGD) merupakan salah satu metode *inquiry* yang dapat digunakan untuk mendapatkan dan memvalidasi masukan pengguna. FGD didefinisikan sebagai suatu proses

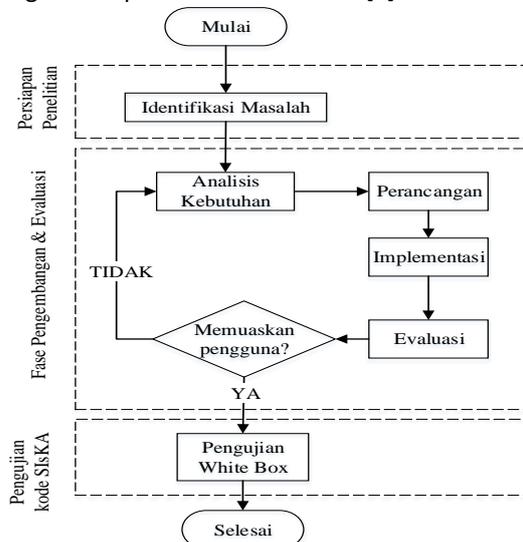
pengumpulan informasi mengenai suatu permasalahan tertentu yang sangat spesifik melalui diskusi kelompok [9]. Hasil dari FGD adalah data kualitatif terkait daftar rekomendasi perbaikan akhir yang akan harus diimplementasikan.

Hasil akhir pengembangan SIsKA juga harus ada pada tingkat kompleksitas yang rendah dari sisi kode-kode pemrogramannya. *White Box Testing* merupakan pengujian yang dapat digunakan untuk menguji setiap kode implementasi SIsKA. Pengujian ini dilakukan oleh pengembang dengan cara menguji tiap baris kode satu-persatu. Pada pengujian ini akan dicari nilai kompleksitas dari setiap kode SIsKA untuk menentukan tingkat kesulitan setiap kode untuk dipahami, diuji dan dipelihara [11].

Berdasarkan paparan diatas, maka dalam penelitian ini akan dilakukan pengembangan SIsKA yang kemudian dievaluasi dengan menggunakan teknik evaluasi *Focus Group Discussion*. Fase pengembangan SIsKA akan terus dilakukan sampai dianggap sudah sesuai dilihat dari hasil FGD. Hasil akhir pengembangan SIsKA selanjutnya diuji menggunakan *White Box Testing* untuk menganalisis kompleksitas kode SIsKA. Hasil akhir penelitian ini akan menghasilkan SIsKA dengan peningkatan pada aspek antarmuka dan fungsionalnya, serta mengetahui tingkat kompleksitas kode SIsKA untuk dipahami, diuji dan dipelihara.

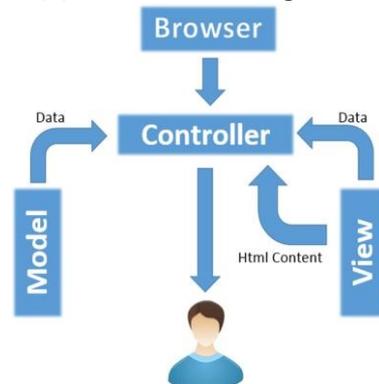
2. METODE PENELITIAN

Penelitian ini akan dilakukan dengan melakukan beberapa tahapan penelitian seperti ditunjukkan pada Gambar 1. Tahapan awal penelitian dimulai dengan mengidentifikasi permasalahan SIsKA, yaitu pada pada antarmuka SIsKA [2] dan fungsional SIsKA yang belum berjalan dengan maksimal dalam pengelolaan penelitian mahasiswa [1].



Gambar 1. Tahapan Penelitian

Fase pengembangan dengan model *incremental* dilakukan dengan beberapa tahapan, yaitu 1) analisis kebutuhan, 2) perancangan, 3) implementasi dan 4) evaluasi. Pada tahap analisis kebutuhan awal, masukan yang digunakan berasal dari hasil studi literatur, yang dalam penelitian ini diperoleh dari hasil penelitian sebelumnya terkait evaluasi antarmuka [2] dan kebutuhan fungsional SIsKA [1].



Gambar 2. Konsep MVC

[Sumber : Tutorialpoints.com]

Pada tahapan perancangan akan dilakukan perancangan sistem sesuai dengan analisis kebutuhan menggunakan *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD). Hasil perancangan kemudian diimplementasikan ke dalam kode-kode pemrograman dan pada tahap akhir fase pengembangan akan dilakukan evaluasi sistem. Proses implementasi pada penelitian ini akan dilakukan dengan memanfaatkan konsep Model View Controller pada *framework CodeIgniter* seperti ditunjukkan Gambar 2 [12].

Pada tahapan evaluasi pengguna, pengumpulan data dilakukan dengan menguji sistem oleh pengguna, lalu mencatat dan memvalidasi masukan yang diberikan. Hal-hal yang dibahas dalam FGD pada penelitian ini adalah sebagai berikut.

- Identifikasi rekomendasi yang dihasilkan dari evaluasi kebermanfaatan pada SIsKA yang telah dilakukan.
- Pengumpulan masukan dan solusi perbaikan pada rekomendasi yang ada.
- Prioritas terhadap rekomendasi yang perlu diimplementasikan

Pengumpulan pada dengan teknik *White Box Testing* akan menghasilkan data kuantitatif terkait kompleksitas kode SIsKA. Beberapa tahapan yang dilakukan dalam pengujian kode menggunakan *White Box Testing* adalah sebagai berikut [13].

- Menggambar *flowgraph* dari kode program yang diuji.

- b. Menghitung kompleksitas $V(G)$ untuk *flowgraph* yang telah dibuat. Kompleksitas $V(G)$ dapat dihitung dengan persamaan (1).
 $V(G) = P + 1$ (1)
 Keterangan:
 P = Jumlah predikat simpul pada *flowgraph*
- c. Analisis kompleksitas dari *flowgraph*.

Kompleksitas yang tinggi menunjukkan prosedur kompleks. Hubungan antara kompleksitas dan resiko dalam suatu prosedur, seperti ditunjukkan pada Tabel 2 [11].

Tabel 2: Hubungan Kompleksitas dan Resiko
 [Sumber : Lala, 2015]

Kompleksitas	Evaluasi Resiko
1-10	Program sederhana, tanpa banyak resiko
11-20	Agak kompleks, resiko sedang
21-50	Kompleks, program resiko tinggi
Lebih dari 50	Program resiko sangat tinggi

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengembangan Fase 1 *Incremental*

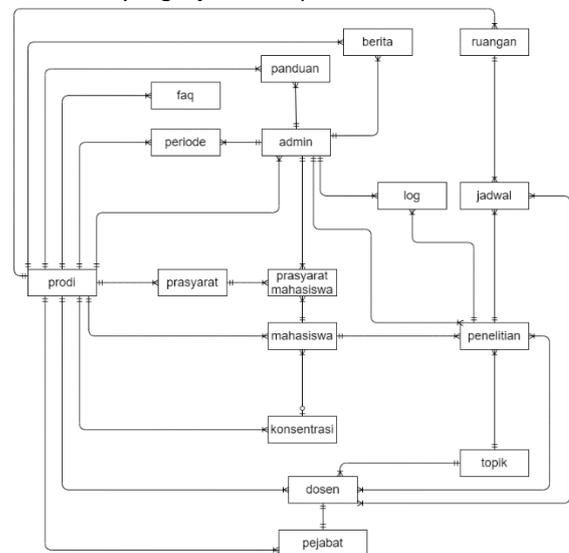
Berdasarkan analisis awal yang dilakukan, maka didefinisikan 23 kebutuhan fungsional yang harus dipenuhi SisKA, ditunjukkan pada Tabel 3.

Tabel 3: Kebutuhan fungsional

No	Kebutuhan Fungsional
1	Autentikasi
2	Pengaturan akun
3	Kelola admin
4	Kelola dosen
5	Kelola mahasiswa
6	Kelola prasyarat mahasiswa
7	Kelola pengajuan
8	Kelola proposal
9	Kelola pra ujian tesis
10	Kelola tesis
11	Kelola konsentrasi
12	Kelola pejabat
13	Kelola ruangan
14	Kelola prasyarat
15	Kelola topik penelitian
16	Kelola periode pengajuan
17	Statistik
18	Kelola jadwal proposal
19	Kelola jadwal pra ujian tesis
20	Kelola jadwal tesis
21	Kelola berita
22	Kelola panduan
23	Kelola <i>Frequently Asked Questions</i> (FAQ)

Perancangan terhadap kebutuhan fungsional tersebut dilakukan dengan menggunakan DFD dan

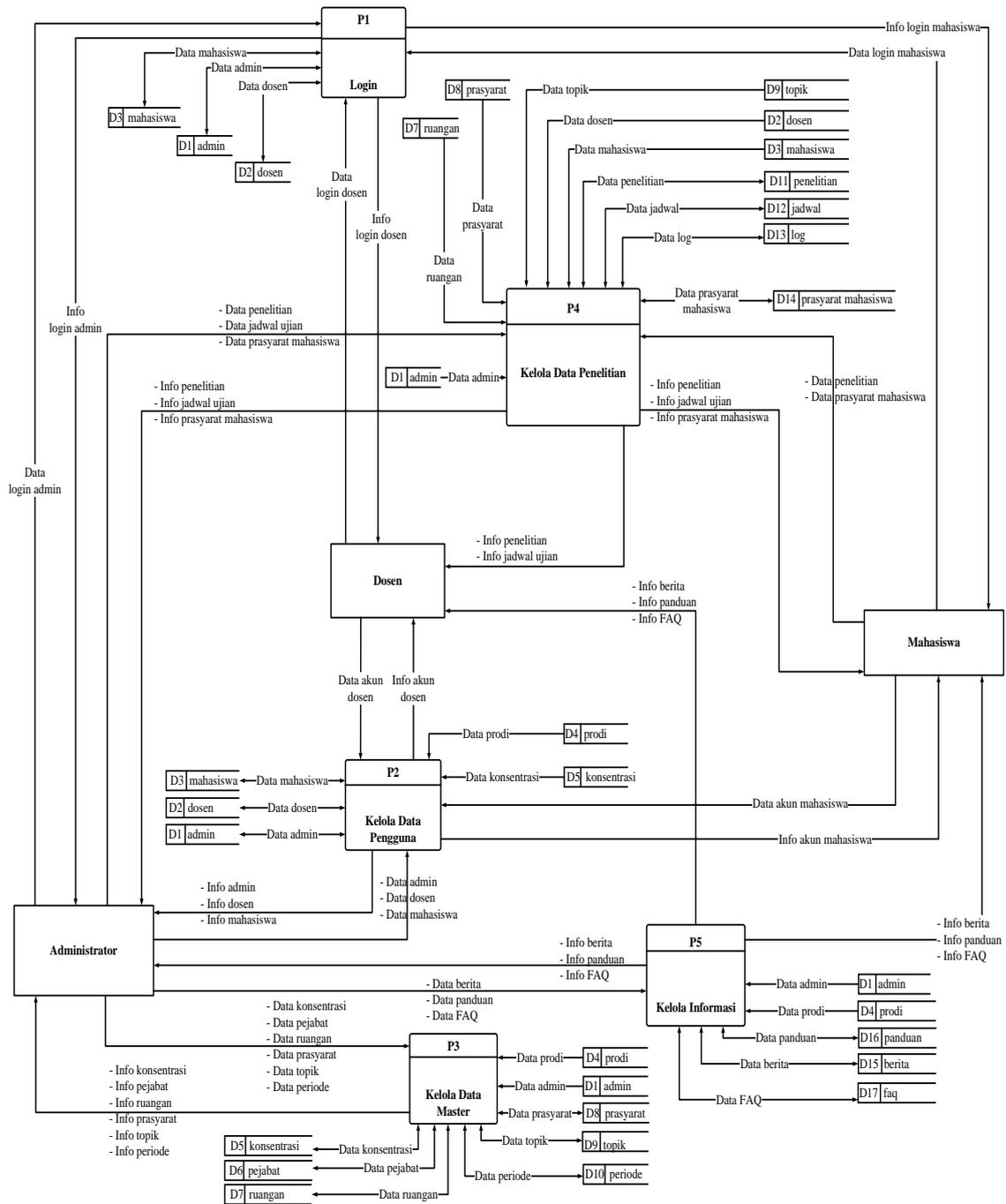
ERD. Pada DFD Level 0 yang ditunjukkan pada Gambar 4, dijabarkan 5 proses utama yang membangun SisKA dari pengelompokan kebutuhan fungsional, yaitu 1) *login*, 2) kelola data pengguna, 3) kelola data master, 4) kelola data penelitian, dan 5) kelola informasi. Pada DFD Level 0 juga dijabarkan *datastore* yang digunakan untuk melakukan penyimpanan data pada setiap proses yang ada. Pada pengembangan awal SisKA ini akan digunakan 17 *datastore* yang selanjutnya dirancang relasinya dengan menggunakan ERD. ERD merupakan diagram yang menggambarkan hubungan antar dua entitas atau lebih yang menyusun sistem. Gambar 3 merupakan ERD fase 1, berdasarkan analisis *datastore* yang dijabarkan pada DFD Level 0.



Gambar 3. Fase 1 – ERD SisKA

Model *Crow's Foot* digunakan dalam menggambarkan relasi antar entitas yang ada, yang dijabarkan sebagai berikut.

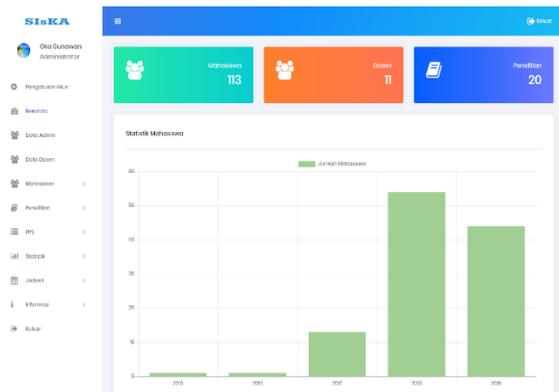
- a. Entitas administrator berelasi 1 ke banyak dengan entitas panduan, berita, periode, *log*, penelitian dan prasyarat mahasiswa.
- b. Entitas prodi berelasi 1 ke banyak dengan entitas ruangan, berita, panduan, FAQ, periode, admin, prasyarat, mahasiswa, konsentrasi, dosen dan pejabat.
- c. Entitas mahasiswa berelasi 1 ke banyak dengan entitas prasyarat mahasiswa dan penelitian. Serta berelasi 1 ke 1 dengan konsentrasi, namun relasi dengan konsentrasi ini bersifat tidak wajib yang berarti mahasiswa bisa tidak memiliki konsentrasi.
- d. Entitas dosen berelasi 1 ke 1 dengan pejabat, berelasi banyak ke 1 dengan topik, berelasi banyak ke banyak dengan penelitian dan jadwal.



Gambar 4. Fase 1 – DFD Level 0

Hasil perancangan kemudian diimplementasikan ke dalam kode pemrograman memanfaatkan *framework CodeIgniter*. Gambar 5 merupakan halaman administrator yang berhasil diimplementasikan pada fase 1. Pada halaman beranda, akan ditampilkan data statistik mahasiswa dengan jumlah data mahasiswa, dosen dan penelitian yang dapat dikelola. Fungsionalitas SIsKA yang dapat dikelola oleh administrator ini disesuaikan dengan rancangan alur proses yang telah ditentukan, diantaranya pengelolaan data

penelitian, data pengguna, data program studi, data jadwal, dan data informasi.



Gambar 5. Fase 1 – Implementasi Halaman Administrator Hasil evaluasi FGD dilakukan dengan menguji SIsKA dengan responden yang merupakan pengguna SIsKA. Responden yang dipakai dalam pengumpulan data ini adalah sebanyak 20 orang [14] [15]. Hasil rekapitulasi validasi masukan yang diberikan oleh semua pengguna ditunjukkan pada Tabel 4.

Tabel 4: Fase 1 – Daftar Rekomendasi Akhir

No	Halaman	Rekomendasi
1.	Administrator	<ol style="list-style-type: none"> 1. Penamaan dan pengelompokkan menu diperjelas dan dikelompokkan. 2. Posisi tombol aksi dipindahkan menjadi ke bagian kiri halaman dan minimalis penggunaan tombol tampilan. 3. Semua <i>field</i> yang diisi oleh administrator diisikan penjelasan. 4. Tabel awal data perlu diperjelas. 5. <i>Tab</i> revisi perlu diberikan warna yang jelas untuk tab aktif. 6. Pemberitahuan <i>error</i> perlu diatur untuk terus muncul. 7. Penambahkan fitur untuk mengakses akun pengguna yang lebih jelas. 8. Atur prasyarat sesuai tahapannya. 9. Terdapat pratinjau saat mengunggah foto. 10. Nama periode pengajuan otomatis berdasarkan bulan dan tahun yang dipilih.

2. Dosen

11. Perbaiki *error* kode pengurutan data di pengajuan.
12. Daftar revisi diunggah oleh administrator.
13. Tambahkan fitur unggah .csv data pengguna.
14. Cetak jadwal dibuat sesuai dengan desain yang sudah ada.

3. Mahasiswa

1. Menu-menu utama perlu dikelompokkan.
2. Tombol aksi diatur ke kiri halaman dan dilengkapi dengan ikon.
3. Tombol simpan perlu diletakkan pada bagian bawah halaman.
4. Minimalisir penggunaan tombol tampilan.
5. Penambahan fitur seleksi data pada tahapan penelitian dan jadwal ujian.
6. Penggunaan bahasa Indonesia pada keseluruhan fitur.
7. FAQ perlu dikategorikan sesuai penggunaannya.
8. Tambahkan fitur verifikasi kelayakan oleh dosen pembimbing.
1. Menu-menu perlu dikelompokkan.
2. Tombol aksi dipindah ke bagian kiri halaman.
3. Tombol simpan data dipindahkan ke bagian bawah halaman.
4. Perjelas *Tab* yang aktif dan tidak aktif dengan memberikan warna.
5. *Field file* pendukung dihapus.
6. Ringkasan tahap sebelumnya otomatis ditampilkan.
7. *Field* wajib diberikan keterangan tambahan.

-
8. Jadwal ujian langsung menampilkan detail jadwal mahasiswa.
 9. Bahasa yang digunakan dalam SIsKA harus jelas dan mudah dimengerti.
 10. Pemberitahuan *error* terus muncul, dengan pengguna memiliki kebebasan untuk menutupnya.
 11. Riwayat penelitian pada beranda diperjelas.
 12. Permudah skema prasyarat mahasiswa.
 13. Integrasi prasyarat mahasiswa dengan penelitian mahasiswa.
-

Berdasarkan Tabel 4, maka hasil pengembangan fase 1 SIsKA menunjukkan bahwa perlu dilakukan pengembangan lanjutan ke fase 2 SIsKA dengan mengimplementasikan rekomendasi perbaikannya.

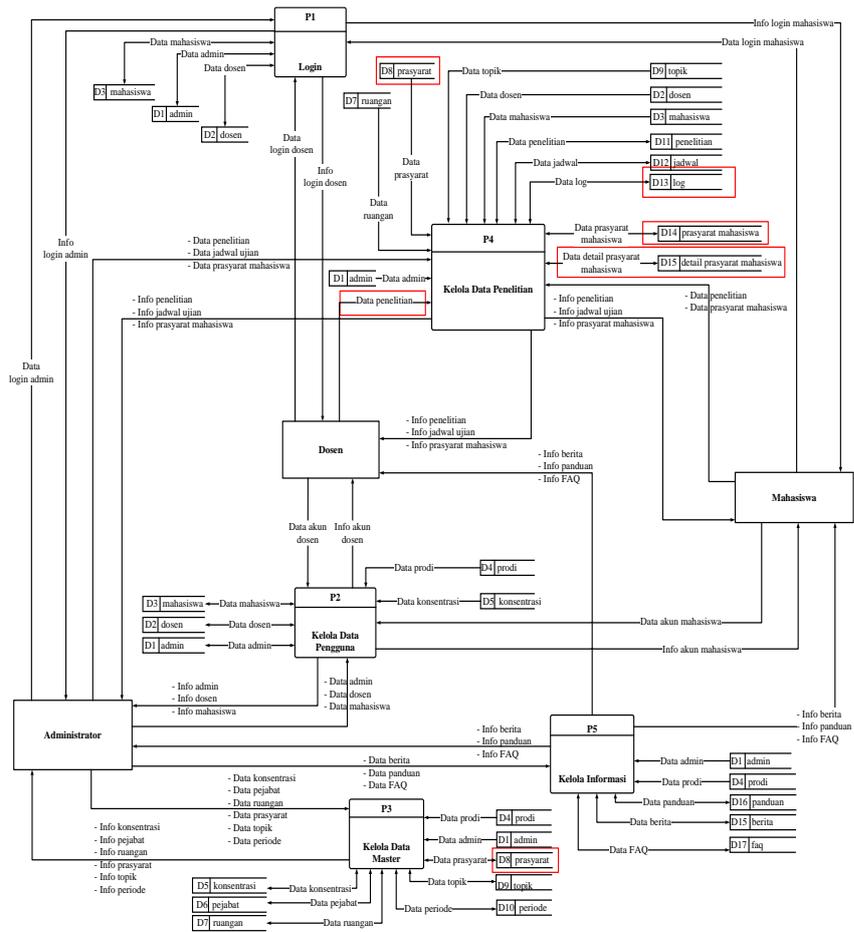
3.2 Hasil Pengembangan Fase 2 *Incremental*

Pada daftar rekomendasi yang diberikan, hanya beberapa rekomendasi saja yang mempengaruhi analisis kebutuhan terkait proses dan basis data yang sudah dirancang pada fase 1, seperti Tabel 5.

Tabel 5: Fase 2 – Analisis Kebutuhan

No	Halaman	Rekomendasi
1.	Administrator	Pengelompokan prasyarat agar sesuai tahapan penelitiannya.
2.	Dosen	Tambahkan fitur verifikasi kelayakan oleh dosen pembimbing pada setiap tahapan yang dilalui.
3.	Mahasiswa	Permudah prasyarat mahasiswa, dimana mahasiswa tinggal mengunggah <i>file</i> prasyarat sesuai prasyarat setiap tahapannya.

Perancangan terhadap kebutuhan tersebut dilakukan dengan menggunakan DFD dan ERD. DFD Level 0 fase 2 yang pada Gambar 6, menunjukkan 2 data masukan baru ke dalam proses kelola data penelitian yaitu data penelitian dari dosen dan data detail prasyarat mahasiswa dari *datastore* detail prasyarat mahasiswa.



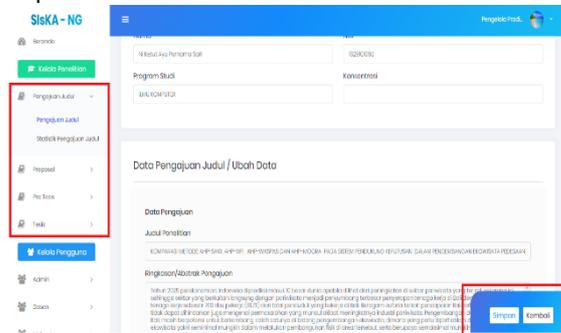
Gambar 6. Fase 2 – DFD Level 0

	3. Tombol simpan diatur agar mudah diakses.
	4. Perbaiki <i>error</i> kode pada fitur tambah ruangan.
2. Dosen	1. Tombol simpan diatur agar mudah diakses. 2. Perbaiki <i>error</i> kode pada detail data tesis.
3. Mahasiswa	1. Tombol simpan diatur agar mudah diakses. 2. Tambahkan fitur <i>loading</i> pada setiap proses penyimpanan data. 3. Pengaturan nama menu "Pengajuan Proposal" menjadi "Pengajuan Judul".

Berdasarkan Tabel 6, maka hasil pengembangan fase 2 SIsKA menunjukkan bahwa perlu dilakukan pengembangan lanjutan ke fase 3 SIsKA dengan mengimplementasikan rekomendasi perbaikan yang telah diberikan.

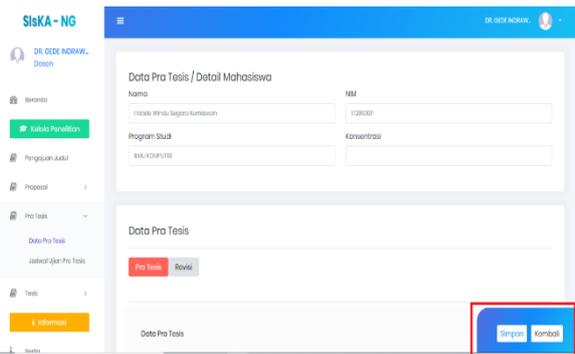
3.3 Hasil Pengembangan Fase 3 Incremental

Analisis kebutuhan awal fase ini didapatkan dari hasil rekomendasi perbaikan fase 2. Pada daftar rekomendasi yang diberikan, tidak terdapat rekomendasi perbaikan yang mempengaruhi analisis kebutuhan terkait proses dan basis data yang sudah dirancang. Rekomendasi perbaikan yang diberikan dari fase 2 pengembangan menunjukkan bahwa diperlukan perbaikan dari sisi antarmuka sistem, sehingga pada fase 3 pengembangan langsung dilakukan tahapan implementasi.



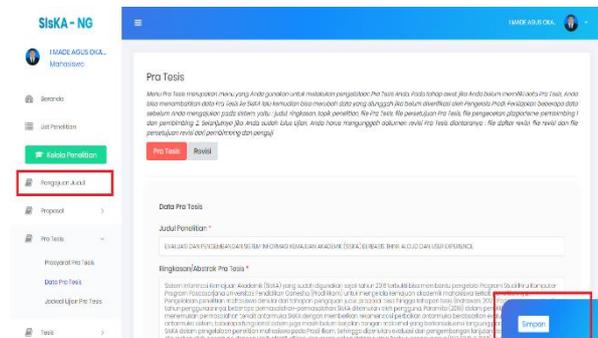
Gambar 9. Fase 3 - Implementasi Halaman Administrator

Gambar 9 merupakan hasil implementasi akhir halaman administrator yang telah dilakukan perbaikan terkait 1) pengelompokan menu penelitian yang telah disesuaikan dengan tahapan penelitiannya dan 2) penempatan tombol simpan yang bisa diakses dari posisi manapun. Rekomendasi lain terkait perubahan label ruangan prodi dan perbaikan *error* kode pada proses menambah data ruangan juga telah dilakukan.



Gambar 10. Fase 3 - Implementasi Halaman Dosen

Gambar 10 merupakan hasil implementasi akhir halaman dosen yang telah dilakukan perbaikan terkait 1) penempatan tombol simpan yang bisa diakses dari posisi manapun. Perbaikan dari rekomendasi terkait *error* kode pada halaman detail data tesis juga telah dilakukan.



Gambar 11. Fase 3 - Implementasi Halaman Mahasiswa

Gambar 11 merupakan hasil implementasi akhir halaman mahasiswa yang telah dilakukan perbaikan terkait 1) penempatan tombol simpan yang bisa diakses dari posisi manapun dan 2) perubahan nama menu pengajuan proposal menjadi pengajuan judul. Rekomendasi lain terkait fitur *loading* pada setiap proses penyimpanan data juga telah dilakukan.

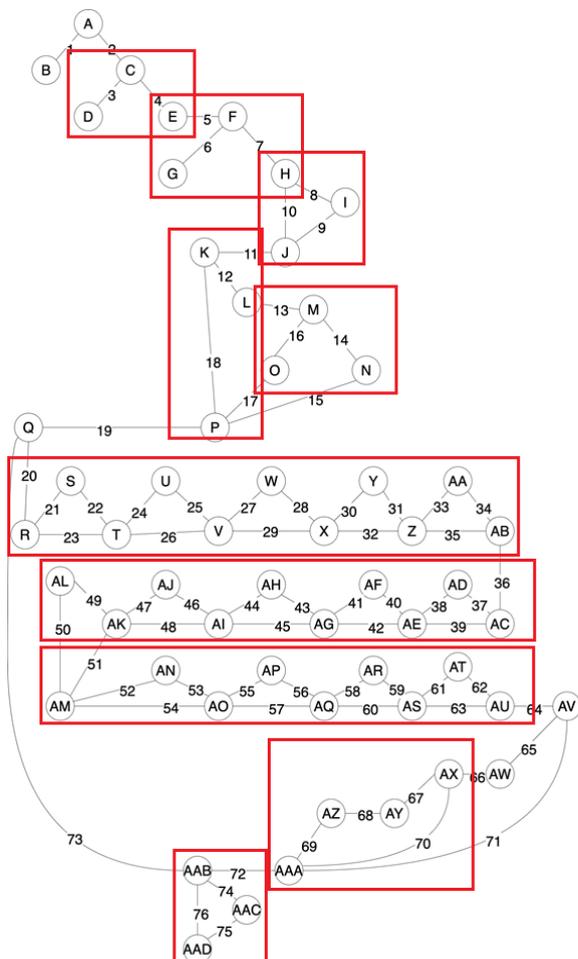
Hasil evaluasi FGD dilakukan dengan menguji SisKA yang telah dikembangkan di fase 3 dengan responden yang merupakan pengguna SisKA. Hasil validasi masukan yang diberikan oleh semua pengguna menunjukkan bahwa hasil fase 3 pengembangan sudah optimal dan dapat diimplementasikan di server Pascasarjana Undiksha.

3.4 Hasil White Box Testing

Hasil akhir pengembangan dan evaluasi pada fase 2 selanjutnya dilakukan pengujian dari sisi kode program. Pengujian kode program ini dilakukan dengan menguji struktur kode program menggunakan metode pengujian *White Box*. Tabel 7 merupakan hasil pengujian kode program yang telah dilakukan pada masing-masing modul SisKA.

Tabel 7: Hasil *White Box Testing*

Halaman	Modul	Fungsi	Fungsi dengan resiko
Administrator	22	84	5
Dosen	10	23	0
Mahasiswa	14	28	9



Gambar 12. *Flowgraph Update Data Proposal*

Hasil pengujian dan analisis kode SisKA menunjukkan bahwa beberapa fungsi perlu diperbaiki untuk menurunkan tingkat kompleksitasnya. Berdasarkan analisis yang dilakukan, kompleksitas pada beberapa fungsi SisKA dipengaruhi oleh penggunaan *if-else* pada validasi *file* yang dimasukkan pada halaman pengguna. Salah satu fungsi yang memiliki tingkat kompleksitas yang cukup tinggi adalah fungsi *update data proposal*, yang ditunjukkan pada Gambar 12.

Gambar 12 merupakan *flowgraph* dari fungsi *update data proposal* pada halaman administrator. Pada fungsi tersebut, terdapat banyak predikat node dari penggunaan *if-else* dalam melakukan validasi data saat *update data proposal*. Proses penghitungan nilai *cyclomatic complexity* yang berdasarkan persamaan (3) adalah sebagai berikut.

$$V(G) = P + 1$$

$$V(G) = 24 \text{ predikat node} + 1$$

$$V(G) = 25$$

Nilai yang didapat dari perhitungan *cyclomatic complexity* adalah 25, sehingga terdapat 25 jalur pengujian yang didapat berdasar perhitungan dari *flowgraph* tersebut. *Cyclomatic complexity* yang bernilai 25 dapat digolongkan bahwa fungsi *update data* dari modul pengelolaan data proposal adalah kompleks, program resiko tinggi [11].

4. KESIMPULAN

Fase pengembangan SisKA pada penelitian ini dilakukan dengan model pengembangan incremental untuk menghasilkan SisKA yang sesuai dengan kebutuhan pengguna. Pada setiap fase pengembangan, analisis dan perancangan yang dilakukan menggunakan Data Flow Diagram dan Entry Relationship Diagram telah berhasil menspesifikasikan kebutuhan – kebutuhan yang harus dipenuhi SisKA. Hasil evaluasi fase 1 dan fase 2 pengembangan memberikan rekomendasi perbaikan. Hasil evaluasi dari fase 3 pengembangan menunjukkan bahwa SisKA sesuai dengan kebutuhan pengguna.

Hasil pengujian dan analisis kode SisKA menunjukkan bahwa beberapa fungsi perlu diperbaiki untuk menurunkan tingkat kompleksitasnya. Kompleksitas pada beberapa fungsi SisKA dipengaruhi oleh penggunaan *if-*

else pada validasi *file* yang dimasukkan pada halaman pengguna. Penelitian selanjutnya dapat melakukan perbaikan pada setiap fungsi yang memiliki tingkat kompleksitas tinggi dan analisis keamanan data pada SIsKA.

PERNYATAAN PENGHARGAAN

Penulis mengucapkan terima kasih kepada Kementerian Pendidikan dan Kebudayaan Indonesia yang telah mendukung penelitian ini melalui dana hibah Penelitian Tesis Magister 2020.

DAFTAR PUSTAKA

- [1] G. Indrawan *et al.*, "SIsKA: Mobile Based Academic Progress Information System," in *2nd International Conference on Innovative Research Across Disciplines*, 2017, vol. 134, no. Icirad, pp. 126–130.
- [2] A. A. I. I. Paramitha, G. R. Dantes, and G. Indrawan, "The evaluation of web based academic progress information system using heuristic evaluation and user experience questionnaire (UEQ)," in *Proceedings of the 3rd International Conference on Informatics and Computing, ICIC 2018*, 2018, pp. 1–6.
- [3] International Organisation for Standardisation (ISO), "Ergonomics of human-system interaction - Part 11: Usability: Definitions and Concepts," in *Iso 9241-11:2018(E)*, 2018, p. 2.
- [4] H. S. Modi, N. K. Singh, and H. P. Chauhan, "Comprehensive Analysis of Software Development Life Cycle Models," *Int. Res. J. Eng. Technol.*, vol. 4, no. 6, pp. 117–122, 2017.
- [5] I. Sommerville, *Software engineering (10th edition)*, 10th ed. Pearson Education Limited, 2016.
- [6] M. Fanani. M. F. Sholiq, "Implementasi Metode Incremental Dalam Membangun Aplikasi Use Case Point Pada Perusahaan DTS," in *Seminar Nasional Sistem Informasi Indonesia*, 2015, pp. 2–3.
- [7] Dwi Putra Githa and D. P. S. Putri, "Rancang Bangun Sistem Rekomendasi Destinasi Wisata Di Bali," *SINTECH (Science Inf. Technol. J.)*, vol. 2, no. 2, pp. 81–90, 2019.
- [8] P. Merson, "Data Model as an Architectural View," 2009.
- [9] L. D. Farida, "Pengukuran User Experience Dengan Pendekatan Usability [Kasus: Website Pariwisata Di Asia Tenggara]," in *Seminar Nasional Teknologi Informasi dan Multimedia*, 2016, no. February 2016, pp. 55–60.
- [10] S. Gupta, "A Comparative study of Usability Evaluation Methods," *Int. J. Comput. Trends Technol.*, vol. 22, no. 3, pp. 103–106, 2015.
- [11] H. Lala, P. Y. Trisnanto, and A. Candra, "Cyclomatic Complexity Test Design Flowgraph Registration of Emergency Installation Patients in Wava Husada Hospital Using SEM," *Int. J. Sci. Res.*, vol. 6, no. 8, pp. 1983–1987, 2017.
- [12] tutorialspoint, "CodeIgniter - MVC Framework," 2020. [Online]. Available: www.tutorialspoint.com/codeigniter/codeigniter_mvc_framework.htm. [Accessed: 27-Jul-2020].
- [13] R. S. Pressman and B. R. Maxim, *Software Engineering A PRACTITIONER 'S APPROACH*, EIGHTH. New York: McGraw-Hill Education, 2015.
- [14] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 4, pp. 40–44, 2017.
- [15] N. L. P. R. Indriyani, G. R. Dantes, and K. Y. Ernanda, "Analisis Kebermanfaatan Website Sekolah Tinggi Pariwisata (Stipar) Triatma Jaya Menggunakan Metode Usability Testing," *Int. J. Nat. Sci. Eng.*, vol. 1, no. 2, pp. 55–64, 2017.