JURNAL ILMIAH FLASH
Vol. 4 No. 1, Halaman: 48 - 52
Juni 2018

DESAIN SISTEM INFORMASI ADMINISTRASI PENDIDIKAN DI PASCASARJANA UNIVERSITAS NEGERI MALANG DENGAN METODE REKAYASA BALIK

Rendy Yani Susanto[1]*, Aji Prasetya Wibawa[2], dan Triyanna Widiyaningtyas[3]
[1,2,3]Teknik Elektro, Universitas Negeri Malang
Jl. Semarang No.5, Sumbersari, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145
*E-mail: gubuksoftware17@gmail.com

Abstrak

Administrasi Pascasarjana Universitas Negeri Malang masih menggunakan teknologi lama yaitu aplikasi PFS:Professional File. Walaupun digunakan hampir 33 tahun, aplikasi ini masihlayakdan stabil untuk pelaksanaan proses administrasi. Berjalannya waktu, aplikasi ini menjadi tidak kompatibel dengan teknologi yang ada. Oleh karena itu, pembaharuan aplikasi sangat dibutuhkan agar kompatibel dengan sistem operasi tersebut. Akan tetapi, aplikasi tersebut tidak memiliki dokumentasi yang digunakan sebagai bahan pembaharuan.Dokumentasi program yang dimaksudadalahberupaSpesifikasiKebutuhanPerangkatLunak (SKPL). Untukmendapatkan SKPL tersebutmembutuhkanmetodeuntukmenganalisisaplikasi yang sudahada. dari permasalahan tersebut, metode yang tepat digunakan adalah metode rekayasa balik. Metode ini dapat mengekstrak informasi dari desain source code pada aplikasi yang tidak terstruktur. Sehingga diharapkan dengan penggunaan metode rekayasa balik, dokumentasi dari aplikasi PFS:Professional File bisa dibuat. Hasil dari dokumentasi tersebut nantinya akan dijadikan bahan pengembangan aplikasi baru diharapkan penggunaan metode rekayasa balik pada aplikasi tersebut bisa menjadi solusi.

Kata kunci: Sistem informasi, Rekayasa Balik, Administrasi.

PENDAHULUAN

Pascasarjana menyelenggarakan pendidikan program magister dan doktor. Di dalamnya terdapat beberapa kegiatan untuk mendukung proses akademik, salah satunya proses administrasi. Kegiatan administrasi yang dilakukan adalah mengelola data akademis secara sistematis.

Pascasarjana Universitas Negeri Malang memiliki bidang yang menangani proses administrasi yaitu bidang akademik. Pada bidang ini, teknologi yang digunakan untuk aktivitas administrasi adalah aplikasi PFS:Professional File. Aplikasi tersebut dibeli pada tahun 1984. Aplikasi ini memiliki fungsi yang mendukung untuk mengelola data seperti memasukan, mengubah, menghapus dan mencetak data sesuai dengan kebutuhan. Dengan kemampuan komplek yang dimiliki aplikasi ini, aktivitas yang tidak tercakup pada portal Sistem Informasi Akademik (SIAKAD) Universitas Negeri Malang bisa berjalan dengan baik. Hingga saat ini, aplikasi ini masih berjalan stabil.

Seiring berjalannya waktu, aplikasi ini menjadi tidak kompatibel dengan sistem operasi yang ada, hal tersebut terjadi karena Pascasarjana melakukan *upgrade* pada komputer yang digunakan. Sehingga aplikasi membutuhkan pembaharuan supaya kompatibel dengan sistem operasi yang baru. Akan tetapi aplikasi ini tidak memiliki dokumentasi program sehingga sulit untuk diperbarui. Dokumentasi program yang dimaksud adalah berupa Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Untuk mendapatkan SKPL tersebut membutuhkan metode untuk menganalisis aplikasi yang sudah ada.

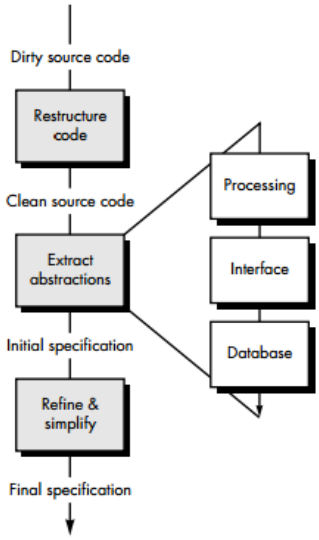
Metode yang tepat digunakan adalah metode rekayasa balik. Metode ini dapat mengekstrak informasi dari desain source code pada aplikasi yang tidak terstruktur. Sehingga diharapkan dengan penggunaan metode rekayasa balik, dokumentasi dari aplikasi *PFS:Professional File* bisa dibuat. Hasil dari dokumentasi tersebut nantinya akan dijadikan bahan pengembangan aplikasi baru.

METODE PENELITIAN

Metode rekayasa balik dapat memunculkan desain lengkap dari sebuah program yang tidak memiliki dokumentasi dan tidak terstruktur[1]. Metode ini seringkali melibatkan sistem yang ada sebagai subjeknya[2]. dalam membuat dokumentasinya dapat dilakukan dengan memetakan program dari representasi tingkat

terendah seperti *source dode*, kode *biner*, dll. Metode ini berperan penting untuk rekayasa perangkat lunak seperti migrasi sistem dan evolusi perangkat lunak[3].

Dari ketiga pendapat tersebut dapat disimpulkan bahwa rekayasa balik bisa dimanfaatkan untuk mendapatkan struktur aplikasi yang tidak memiliki dokumentasi dengan tahapan yang ditentukan. Langkah-langkah metode ini ditunjukan pada gambar 1.



Gambar 1. Langkah-langkah rekayasa balik

Dari gambar tersebut tahapan metode rekayasa balik meliputi :

1. Tahap *restructure code*

Tahap ini bertujuan untuk mendapatkan struktur kode dengan cara memetakan kode asli dari program [1]. Pemetaan kode tersebut dapat dilakukan dengan membuat *refactoring code* dari kode program[4]. Tahap ini akan diimplementasikan di Pascasarjana Universitas Negeri Malang dengan cara melakukan analisis kode program PFS:Professional File.

2. Tahap*extract abstraction*

Tahap ini bertujuan untuk menjabaran lebih lanjut tentang hasil yang didapat dari proses *restructurecode*[5]. *Extract* abstraction memiliki tahapan yaitu :

a. *Processing*

Tahap ini bertujuan untuk mendapatkan struktur kode dengan cara memetakan kode asli dari program [1]. Pemetaan kode tersebut dapat dilakukan dengan membuat *refactoring code* dari kode program[4]. Tahap ini akan diimplementasikan di Pasca sarjana Universitas Negeri Malang dengan cara melakukanan alisis kode program *PFS: Professional File*.

b. *Interfaces*

Untuk memahami secara penuh *use rinterface* yang sudah ada, struktur dan tingkahlaku *interface* harus ditentukan terlebih dahulu[1]. *Interface* dapat dianalisis dengan menggunakan *Data Flow Diagram* (DFD), diagram *usecase*, dan *sequence diagram*.

c. *Database*

Terlepas dari struktur fisiknya, *database* memiliki beberapa metode untuk membangun hubungan antar objek[1]. Dengan fungsi tersebut *database* sangat berperan dalam membangun metode rekayasa balik. Langkah-langkah untuk mengekstrak *database* yaitu (1). Membangun inisial *objek*, (2). Mendefinisikan primary key dari objek, (3). Penyaringan class yang bersifat sementara, (4). Mendefinisikan class yang bersifat umum menjadi class spesifik, (5). Menemukan hubungan data dengan menggunakan Teknik CRC (*Cyclic Redundancy Check*). Langkah tersebut nantinya akan digunakan untuk menganalisa struktur *database* dari aplikasi yang ada di Pascasarjana Universitas Negeri Malang.

3. Tahap*refine and simplify*

Tahap ini adalah penyaringan dan penyimpulan dari analisis yang didapatkan. Penyaringan pada spesifikas iawal dapat berupa restrukturisasi perangkat lunak yang bertujuan untuk (1) menjadikan program memiliki kualitas lebih tinggi dari sebelumnya, (2) meningkatkan produktivitas program, (3) mengurangi jadwal pemeliharaan, dan (4) membuat program lebih mudah diuji.[1].

HASIL DAN PEMBAHASAN

Analisis ini menghasilkan desain produk dengan menggunakan metode rekayasa engineering. Tahapan metode reverse engineering yaitu.

1. Hasil *restructure code*

Tahap ini adalah melakukan abstraksi fungsionalitas dan prosedural terhadap

aplikasi yang lama. Abstraksi fungsionalitas dilakukan dengan menganalisis fungsi yang terdapat pada aplikasi lama. Hasil dari abstraksi fungsionalitas ini ditunjukan pada tabel 1.

Tabel 1. FungsionalitasAplikasi

No	Menu	Keterangan
1	*Master Pagefile*	Menambahkan halaman untuk input data
2	*Master Form File*	Penambahan komponen input

Berdasarkan Tabel 1, fungsionalitas menunjukan bahwa fitur utama dari aplikasi PFS: Professional File yaitu penambahan master pagefile dan master file. Pagefile berfungsi untuk halaman latar belakang yang nantinya berisi komponen form, master form file yaitu sehingga untuk komponen lain, pengguna harus memasukkan lagi ke dalam aplikasi tersebut. Identifikasi pola prosedural aplikasi dapat ditunjukkan dengan menggunakan *pseudocode* untuk tiap fungsinya. *Pseudocode* fungsi *setmaster page file* ditunjukan pada tabel 2.

Tabel 2. *Pseudocode* fungsi*Master Pagefile*

Pseudocode 1 : Master Pagefile
Input : data component:string(user) *Output : a set of function(func)* *For each user do* *Components.add()* *End* *For each components do* *func.add(components)* *End* *Return func;*

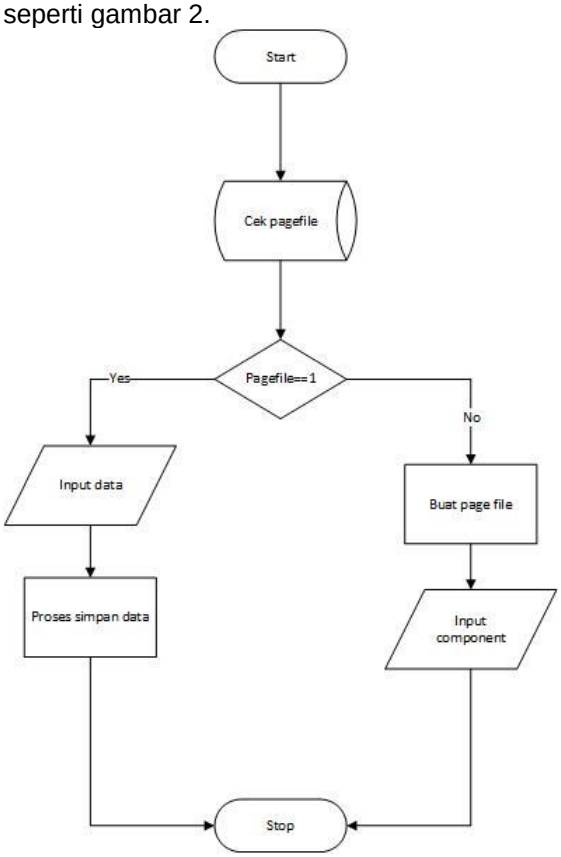
Pada table tersebut menunjukan bahwa, pengguna memiliki hak untuk membuat komponen program sesua idengan kebutuhannya. Aturannya adalah jika pengguna belum membuat page file yang berfungsi sebagai latar belakang aplikasi maka harus membuat page file terlebih dahulu, kemudian membuat komponen form yang dibutuhkan.

2. Hasil *extract abstraction*

Tahap *extract abstraction* ini berguna untuk melakukan penjabaran lebih lanjut hasil yang didapat dari proses *restructure code*. Hasil yang didapat pada *restructure code* yaitu berupa table fungsionalitas dan *pseudocode*, akan dijabarkan lebih lanjut kedalam tahap *processing, interface* dan *database.*

a. *Processing*

Tahapan yang terdapat pada aktifitas processing yaitul dentifikasi pola procedural aplikasi dengan menggunakan model diagram. Pada aplikasi PFS:Professional File pola procedural secara keseluruhan ditunjukan dengan menggunakan diagram alir (*flowchart)* seperti gambar 2.



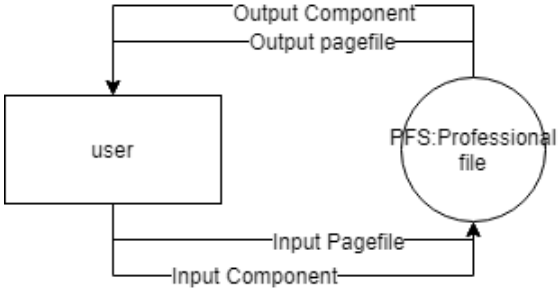
Gambar 2. Diagram alir pola procedural aplikasi PFS: Professional File

Pada flowchart tersebut, langkah pertama adalah melakukan pengecekan apakah pagefile sudah dibuat. apabila pagefile belum ada maka tahap selanjutnya adalah membuat pagefile baru. Langkah selanjutnya adalah menambahkan komponen untuk form yang digunakan untuk input data. Akan tetapi, jika pagefile sudah ada maka proses selanjutnya adalah menambahkan komponen.

b. *Interface*

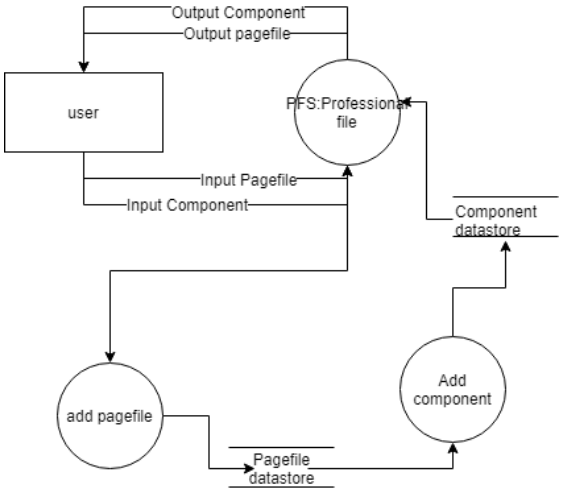
Interface dapat dianalisis dengan menggunakan data flow diagram (DFD), diagram *use case*, dan *sequence diagram*.

DFD berguna untuk menggambarkan suatu sistem yang dikembangka nberhubungan dengan proses, alir data dan penyimpanan data. Analisis *PFS : Professional file* menggunakan DFD level 0 dan DFD level 1, DFD level 0 ditunjukanpadagambar3.



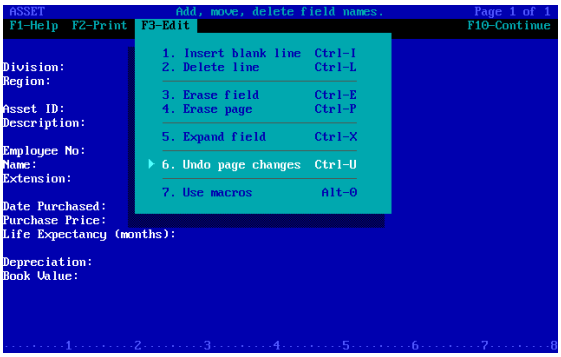
Gambar 3. DFD level 0aplikasiPFS:Professional File

Alur yang terdapat pada DFD level 0 yaitu user melakukan input *page file* dan *component* kepada aplikasi, dan aplikasi memberikan *feedback* berupa *output page file* dan *component* kepada user. DFD level 0 iniakan dijabarkan lebih detail menjadi DFD level 1. Tujuannya adalah melihat alur proses lebih jauh lagi. DFD level 1 ditunjukan pada gambar 4.



Gambar 4. DFD level 1 aplikasi PFS : Professional File

Selain melakukan analisa terhadap interaksi pengguna dengan aplikasi, tahap ini juga menganalisa *user interface* yang ada pada aplikasi lama. Tujuannya adalah untuk menjabarkan dengan kebutuhan yang terdapat pada hasil analisa kebutuhan. Desain user interface ditunjukan pada gambar 5.

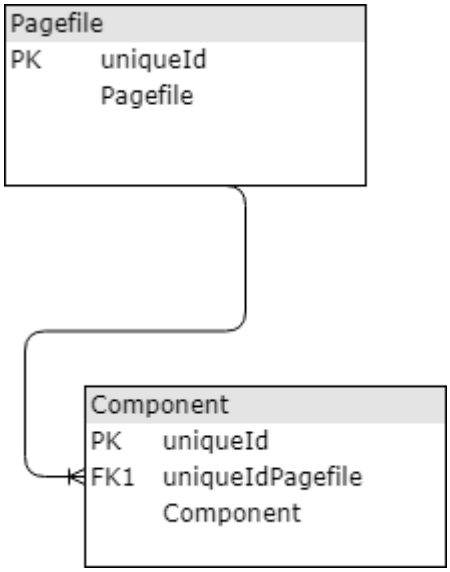


Gambar 5. *User interface* aplikasi PFS : Professional File

Desain tersebut menunjukan fungsi pembuatan page file dan komponen *form*. Dari menu tersebut terdapat beberapa komponen utama dalam membangun sebuah *page file* yaitu menambahkan baris baru, menghapus baris, menghapus halaman, dan menggunakan *macro*.

c. *Database*

Tahap ini adalah membuat desain database pada aplikasi lama dengan *entity relation diagram* (ERD) guna mengetahui hubungan data antar komponen. ERD ditunjukan pada gamba r6.



Gambar 5. *User interface* aplikasi PFS: Professional File

Diagram tersebut menunjukan 2 komponen yang terdapat pada kebutuhan user, yaitu page file dan *component*. Hubungan data antara page file dan *component* menggunakan *onetomany*, yaitu 1 page file bisa memiliki banyak *component*.

3. Hasil *refine and simplify*

Tahap ini adalah menyimpulkan dan menyederhanakan proses rekayasa balik. Penyederhanaan ditujukan untuk meningkatkan kinerja aplikas idengan mengurangi proses yang tidak perlu. Untuk menyimpulkan dan menyederhanakan struktur aplikasi yaitu menggunakan *code refactoring*. Tujuannya mengetahui kode maupun proses yang memungkinkan untuk menimbulkan proses menjadi ambat. *Code refactoring* pada aplikasi ini ditunjukan pada tabel 3.

Tabel 3. *Code refactoring* Aplikasi PFS

Methods	*Refactoring mechanism*	*Advantages*
Menghapus *code clones* (Tahap *restructure code)*	Beberapa kode yang sama akan pusatkan pada 1 *function*	Menghilangkan *redundancy* kode
Membagi fungsionalitas kemasing-masing data-*store* (Tahap interface)	Membuat halaman khusus untuk masing-masing fungsi	Menjadikan alur program lebih terstruktur
Mengurangi penggunaan *statement GOTO* *(*Tahap *restructure code)*	Menghapus beberapa *sintaxgoto.*	Agar tidak terjadi proses yang berulang

PENUTUP

Penggunaan metode rekayasa balik sangat berguna untuk pengembangan ulang suatu perangkat lunak yang tidak memiliki dokumentasi lengkap. Dengan hasil penggunaan metode tersebut, penggunaakan mendapat spesifikasi akhir dari aplikasi yang telah dianalisis. Hasil yang didapatkan dari penggunaan metode rekayasa balik pada aplikasi PFS : Professional File telah menjadi bukti, bahwa metode ini memudahkan pengguna untuk mendapatkan kerangka yang terdapat pada perangkat lunak.

Selain digunakan untuk mendapatkan kerangka aplikasi yang tidak terdokumentasi dengan baik, metode ini juga dapat digunakan sebagai landasan untuk melakukan *maintenance* terhadap suatu aplikasi. Landasant ersebut diambil dari langkah terakhir yaitu menggunakan *code refactoring* sehingga dapat diketahui apa saja yang perlu diperbaiki.

UCAPAN TERIMAKASIH

Penulis mengucapkan terimakasih kepada bapak AjiPrasetya Wibawa dan ibu Triyana Widyaningtyas yang telah membimbing dan member dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009.

M. A. Bari and S. Ahamad, "Process of Reverse Engineering of Enterprise Information System Architecture," vol. 8, no. 5, 2011.

A. Van Deursen and E. Burd, "Software reverse engineering," *J. Syst. Softw.*, vol. 77, no. 3, pp. 209–211, 2005.

P. Kulkarni, P. Rane, S. Patil, and B. B. Meshram, "Code Restructuring : Tool for Quality Improvment," vol. 8491, pp. 122–126, 2011.

A. Singhal and G. Shlok, "Reverse engineering," *Int. J. Comput. Appl.*, vol. 108, no. 9, pp. 1–2, 2014.