

# PENERAPAN REKAYASA ULANG ITERATIF PADA SISTEM INFORMASI AKADEMIK FTIF ITS SURABAYA

**Siti Rochimah, Choiru Zain, Darlis Heru Murti**

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS) – Surabaya

Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5939363  
Email: siti@its-sby.edu, choy01@inf.its-sby.edu, darlis@its-sby.edu

## ABSTRAK

*Selama penggunaannya, sistem warisan (legacy) dapat terjadi banyak aktifitas pemeliharaan (maintenance) yang dapat menyebabkan penurunan kualitas sistem. Ketika sistem mengalami penurunan sampai tingkat kritis, sistem warisan perlu direkayasa ulang. Untuk mempertahankan keberadaan dan keberlangsungan sistem perlu dilakukan rekayasa ulang secara bertahap dalam domain aplikasi sistem warisan. Dalam prosesnya, pada sistem warisan dilakukan rekayasa ulang komponen demi komponen baik program maupun data membentuk sistem tujuan yang telah direkayasa ulang. Proses ini berakhir sampai akhirnya sistem warisan habis menjadi sistem tujuan. Metode ini dinamakan Rekayasa Ulang Iteratif.*

*Metode Rekayasa Ulang Iteratif memperhatikan satu hal yang sangat penting dari metode rekayasa ulang lain yaitu memastikan pembagian data dan fungsi antara sistem warisan dan sistem yang telah direkayasa ulang. Selain itu selama proses rekayasa ulang antara komponen warisan dan komponen yang telah direkayasa ulang dapat berjalan bersamaan. Implementasi dari Rekayasa Ulang Iteratif dalam penelitian ini dilakukan pada studi kasus Sistem Informasi Akademik FTIF ITS Surabaya. Hasil dari Rekayasa Ulang Iteratif ini telah dapat menyelesaikan permasalahan rekayasa ulang sistem warisan menjadi sistem yang telah direkayasa ulang sebagian yaitu pada proses login, frs, dan perwalian.*

*Rekayasa ulang yang dilakukan pada ketiga proses tersebut menunjukkan bahwa proses Rekayasa Ulang Iteratif berhasil dilakukan karena diantara proses yang sudah direkayasa ulang dan sistem warisan dapat berjalan bersama. Dengan demikian proses Rekayasa Ulang Iteratif berlangsung tanpa membekukan Sistem Informasi Akademik FTIF ITS Surabaya yang merupakan sistem warisan. Dalam hal ini sistem warisan selain proses login, frs, dan perwalian yang masih ada masih terus berjalan sampai sistem warisan secara keseluruhan berhasil direkayasa ulang.*

**Kata kunci:** *Rekayasa Ulang Iteratif, Sistem Warisan, Peremajaan Sistem Warisan.*

## 1. PENDAHULUAN

Pentingnya sistem warisan untuk suatu organisasi atau perusahaan yang memilikinya diibaratkan sebagai “tulang punggung” dari aliran informasi organisasi atau perusahaan dan memainkan peran utama untuk penyediaan informasi bisnis. Sistem dan data yang diproses di sistem warisan merupakan aset yang vital bagi organisasi atau perusahaan tersebut.

Seiring dengan penggunaannya, sistem warisan akan mengalami banyak aktifitas yang menyebabkan penurunan kualitas dari sistem. Selain karena penggunaan, penurunan sistem juga dapat disebabkan oleh aktifitas pemeliharaan (*maintenance*). Ketika penurunan kualitas sistem warisan telah melewati batas kritis, sistem perlu direkayasa ulang untuk melindungi aset yang berada di sistem warisan.

Dalam proses rekayasa ulang perlu diperhatikan bahwa aktifitas sistem harus dijamin terus berjalan dengan pertimbangan kepentingan bisnis tertentu yang akan menjadi sangat kritis apabila dihentikan. Dengan pertimbangan tersebut,

sistem warisan perlu direkayasa ulang secara bertahap, berulang dan sedikit demi sedikit. Lebih dari itu, setiap program harus direkayasa ulang dengan periode waktu sesingkat mungkin agar tidak banyak mengganggu aktifitas bisnis.

Tujuan penelitian ini adalah menerapkan model Rekayasa Ulang Iteratif pada studi kasus Sistem Informasi Akademik FTIF ITS Surabaya. Penelitian ini menerapkan proses rekayasa ulang pada domain aplikasi sistem warisan. Proses rekayasa ulang ini dilakukan secara iteratif, bertahap dengan sedikit fungsi atau prosedur pada setiap waktunya dan sesingkat mungkin. Sedangkan manfaat dari Rekayasa Ulang Iteratif adalah perubahan sistem warisan ke sistem yang baru (tujuan) tidak mengganggu pengguna sistem dalam melakukan proses bisnis.

Dalam penelitian ini, permasalahan yang dibahas adalah:

1. bagaimana melakukan proses rekayasa ulang dengan metode Iteratif pada studi kasus Sistem Informasi Akademik FTIF ITS Surabaya; dan
2. bagaimana menganalisis proses Rekayasa Ulang Iteratif dimana sistem tetap berjalan

walaupun proses rekayasa ulang masih berlangsung.

Dari permasalahan di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut.

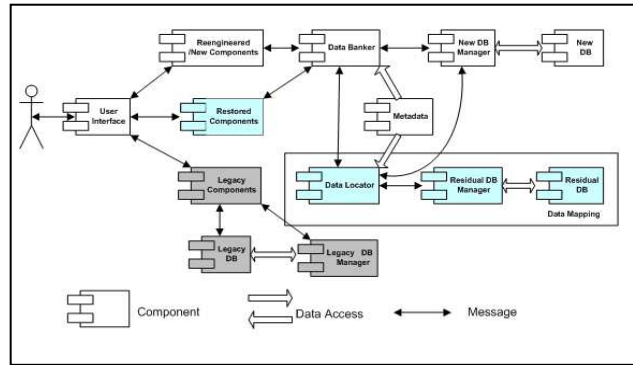
1. Proses Rekayasa Ulang Iteratif dilakukan dengan menitikberatkan pada berjalannya dua sistem, yaitu sistem warisan dan sistem tujuan yang telah direkayasa ulang, tidak secara khusus membahas masalah rekayasa ulang data.
2. Rekayasa ulang hanya melakukan permodelan dari proses Rekayasa Ulang Iteratif tanpa lebih rinci membahas masalah kinerja sistem antara sistem yang lama (warisan) dan sistem baru (tujuan).
3. Sebagai studi kasus, rekayasa ulang ini dilakukan hanya pada proses login, frs, dan perwalian pada Sistem Informasi Akademik FTIF ITS Surabaya.

## 2. PENGERTIAN REKAYASA ULANG ITERATIF

Proses Rekayasa Ulang Iteratif adalah rekayasa ulang sistem secara bertahap. Dalam Rekayasa Ulang Iteratif yang dilakukan secara bertahap tersebut terdapat berbagai macam tahap komponen. Berdasarkan spesifikasi UML komponen adalah bagian dari sistem yang dapat digantikan yang mengimplementasikan dan menyediakan realisasi dari sekumpulan antarmuka. Setiap komponen dalam sistem dapat terdiri dari beberapa keadaan selama proses eksekusi Rekayasa Ulang Iteratif berlangsung. Komponen terdiri dari beberapa fase evolusi sistem yaitu [1]:

- *Warisan*, adalah komponen dari sistem warisan atau komponen yang belum direkayasa ulang.
- *Restored*, adalah komponen yang dimunculkan kembali dari sistem warisan dan bertujuan agar dapat beradaptasi dengan data yang telah direkayasa ulang.
- *Reengineered*, adalah komponen dari sistem warisan yang sudah dimodifikasi dan menjadi bentuk yang baru dan mencapai level tertentu.
- *New*, adalah komponen yang tidak terdapat di sistem warisan, tetapi ditambahkan untuk melengkapi sistem yang telah ada di domain aplikasi yang sama.

Metode Rekayasa Ulang Iteratif dilakukan berdasarkan evolusi secara bertahap dari sistem warisan. Dengan metode Rekayasa Ulang Iteratif ini komponen sistem warisan ini akan terjamin berjalan dalam semua proses. Arsitektur sistem yang mengalami evolusi pada saat rekayasa ulang berlangsung dijelaskan dalam gambar di bawah ini



**Gambar 1. Arsitektur sistem pada saat rekayasa ulang berlangsung**

Gambar 1 menunjukkan sistem arsitektur perangkat lunak ketika eksekusi dari proses Rekayasa Ulang berjalan. Arsitektur yang ditunjukkan pada Rekayasa Ulang terpisah baik dari komponen warisan maupun dari komponen yang direkayasa ulang. Dalam hal ini, baik warisan maupun komponen yang direkayasa ulang masing-masing berjalan.

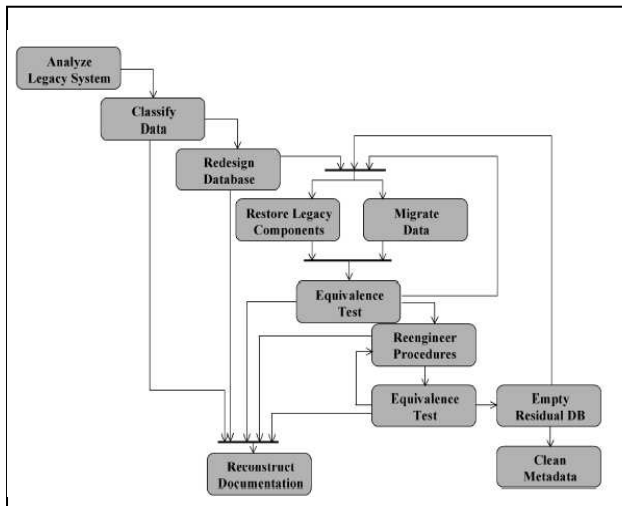
Eksekusi sistem dioperasikan baik di komponen warisan, maupun di komponen yang telah direkayasa ulang. Masing-masing komponen ini harus sama-sama berjalan dan bekerjasama untuk meyakinkan keberlangsungan sistem.

## 3. METODE DAN PROSES REKAYASA ULANG ITERATIF

Metode Rekayasa Ulang Iteratif melibatkan dua hal yang sangat penting, yaitu Rekayasa Ulang data dan Rekayasa Ulang fungsi, prosedur, atau komponen. Fitur inovatif yang ditawarkan dalam Rekayasa Ulang data adalah dapat mengeliminasi gejala penuaan di basis data warisan. Sedangkan Rekayasa Ulang prosedur adalah melakukan perubahan dari program warisan menjadi program yang sudah diadaptasikan sehingga sedikit demi sedikit dapat berintegrasi dengan data yang telah direkayasa ulang.

Proses Rekayasa Ulang Iteratif, termasuk *breaking down* sistem, terbagi menjadi beberapa bagian. Proses ini adalah proses *reverse engineering*. Yaitu menganalisis data, mendesain ulang data, adaptasi data warisan, perpindahan data, tes ekivalensi, kemudian iterasi kembali yang menjadi ciri khas metode ini. Untuk penghapusan metadata dan basis data residual dapat dilakukan setelah proses rekayasa ulang secara keseluruhan selesai.

Adapun proses Rekayasa Ulang Iteratif digambarkan dalam Gambar 2. berikut ini.



**Gambar 2. Proses Rekayasa Ulang Iteratif**

Masing-masing pengertian dari proses-proses pada gambar 2 adalah sebagai berikut ini.

**(a) Analisis Sistem Warisan**

Fase ini membutuhkan identifikasi, analisis, dan interpretasi dari komponen dan data di sistem warisan. Proses adalah analisis komponen dari sistem warisan untuk menentukan komponen-komponen yang dapat dipecah. Komponen ini dipartisi dan bertujuan agar tidak ada saling ketergantungan antar komponennya. Komponen yang berdiri sendiri ini dapat mulai direkayasa ulang. Setelah dilakukan proses Rekayasa Ulang akan menghasilkan komponen baru dan berada di sistem tujuan. Dalam hal ini Rekayasa Ulang komponen antara iterasi yang pertama tidak tergantung oleh Rekayasa Ulang komponen pada tahap yang lainnya. Selanjutnya setelah satu komponen direkayasa ulang harus dipastikan bahwa komponen tersebut dapat berinteraksi dengan komponen di sistem warisan yang masih tersisa tanpa menghilangkan fungsionalitas sistem warisan yang pernah ada. Tentu saja jika pernah ada proses rekayasa ulang dari iterasi sebelumnya, komponen yang baru direkayasa ulang ini dapat juga berinteraksi dengan komponen-komponen di sistem tujuan.

**(b) Klasifikasi Data**

Fase ini terdiri dari identifikasi, dan interpretasi dari data yang disimpan di basis data warisan dan relasinya. Fase ini juga biasa dikenal dengan Rekayasa Balik (*reverse engineering*). Pengklasifikasian data ini ditekankan pada pengelempokan data menjadi data essensial atau primer dan data residual. Pada umumnya data adalah data primer, oleh karena itu lebih mudah jika pengklasifikasian data difokuskan pada penentuan data residual, yaitu data kontrol, data redundan struktural, data redundan semantik, data redundan komputasi.

**(c) Desain Ulang Data**

Selama desain ulang basis data, berbagai macam data yang sudah diklasifikasikan harus direstrukturisasi di basis data tujuan agar lebih efektif. Restrukturisasi ini adalah mendesain ulang organisasi data, identifikasi relasi, dependensi, *primary key*, *foreign key* diantara entitas pada basis data tujuan. Proses ini dinamakan normalisasi.

**(d) Mengadaptasikan Program Warisan**

Setiap program di sistem warisan yang mengakses data yang direkayasa ulang sebelumnya harus diadaptasikan agar dapat mengakses basis data tujuan. Hal ini menunjukkan bahwa setiap program harus dianalisis untuk mengidentifikasi instruksi pengaksesan data. Instruksi ini akan diganti oleh *service request* melalui data locator.

**(e) Perpindahan Data**

Perpindahan data penting di basis data warisan secara bertahap dipindahkan ke basis data tujuan, sedangkan data yang tidak penting dipindahkan ke residual basis data.

**(f) Desain Ulang Fungsi**

Dalam tahap ini dilakukan proses desain ulang fungsi dan komponen, sehingga fungsi secara struktur menjadi lebih baik. Hal-hal yang dilakukan pada tahap ini adalah merestrukturisasi komponen, menyatukan prosedur yang berbeda-beda namun memiliki fungsi yang sama, melakukan update data manajemen akses, memperbaiki algoritma yang digunakan, memperbaiki user interface dan modul interface, melakukan update bahasa pemrograman ke bahasa yang lebih modern, dan mengeksekusi hasil komponen yang dimunculkan kembali.

**(g) Tes Ekuivalensi**

Sistem informasi yang diperoleh di tahap sebelumnya yang telah siap untuk di jalankan, yang memiliki fungsi yang sama dengan sistem warisan di tes untuk memastikan bahwa sistem yang direkayasa ulang memiliki persamaan dengan sistem warisan.

**(h) Membersihkan Data Residual**

Pada akhir Rekayasa Ulang data residual akan dihapus. Karena tidak ada fungsi yang mengakses data ini.

**(i) Iterasi**

Setiap rekayasa ulang komponen dilakukan dan data telah dipindahkan dari sistem warisan ke sistem tujuan dan telah mengalami tahap analisis sampai dengan tes ekuivalensi, akan dilakukan hal yang sama sampai keseluruhan komponen dan data

direkayasa ulang. Hal ini yang menjadi ciri khas dari metode Rekayasa Ulang Iteratif.

**(j) Membersihkan Metadata**

Ketika seluruh sistem warisan direkayasa ulang, *metadata* hanya terdiri dari data yang terfokus pada data yang dicatat pada desain basis data tujuan. Semua data dalam metadata akan tidak berguna, dan data ini dapat dihapus pada akhir rekayasa ulang.

**4. PROSES REKAYASA ULANG ITERATIF PADA SISTEM INFORMASI AKADEMIK FTIF ITS SURABAYA**

**(a) Analisis Sistem Warisan**

Dalam proses analisis sistem warisan, terdapat pemisahan sistem menjadi beberapa sub komponen yang saling berdiri sendiri, sehingga proses Rekayasa Ulang pada suatu komponen tidak akan tergantung pada komponen yang lain. Pemisahan ini juga berfungsi untuk menentukan, komponen apa saja yang sebaiknya direkayasa ulang terlebih dahulu sehingga meminimalisir waktu pembekuan dari komponen tersebut. Untuk menentukan apakah sebuah komponen layak atau tidak menjadi sebuah komponen tunggal ataukah masih perlu dipecah lagi, secara umum digunakan rumus MTMR (*mean time to maintenance request*).

$$MTMR_i = \sum_{j=1}^{n_i} \frac{\Delta t_{j,i}}{n_i}$$

**Gambar 3. Rumus penentuan komponen tunggal yang perlu untuk dipecah lagi**

Merupakan waktu rata-rata akses data setiap kali komponen sistem melakukan tugasnya. Jika perkiraan waktu merekayasa ulang suatu komponen RT<sub>i</sub> lebih dari MTTMR<sub>i</sub>, maka perlu adanya pemecahan lagi menjadi komponen yang lebih kecil. Ini digunakan sebagai acuan untuk meminimalisir waktu pembekuan komponen.

Dalam kasus Sistem Informasi Akademik FTIF ITS Surabaya tidak diperlukan karena tidak ada permintaan perubahan fungsionalitas, hanya diperlukan perubahan platform dari ASP dan ASP .Net.

Dalam proses ini juga terdapat analisis hubungan antara komponen dengan data sehingga diketahui data-data mana yang bersifat esensial maupun data yang residual.

**(b) Klasifikasi Data**

Tahap ini adalah proses pengklasifikasian basis data warisan. Proses klasifikasi data ini akan dibatasi hanya pada data yang berhubungan dengan proses Login, Frs, dan Perwalian. Data secara umum bersifat primer atau esensial, namun ada beberapa data yang bersifat residual. Data residual yang terdapat dalam entitas yang berhubungan

dengan proses login, frs, dan perwalian dapat dilihat dalam Tabel 1.

**Tabel 1. Data Redundan yang ada di BasisData**

Tabel	Kolom	Fungsi yang berhubungan	Jenis Data
Mahasiswa	MA_IPS	UpdateIPSe mesterLalu	Data Redundan Komputasi
Mahasiswa	MA_IPK	UpdateIPKM hs	Data Redundan Komputasi
Mahasiswa	MA_SKSLulus	UpdateSKSL ulus	Data Redundan Komputasi
Mahasiswa	MA_SKSTempuh	UpdateSKST empuh	Data Redundan Komputasi
Kelas	KE_Terisi	Update dari jumlah di tabel _temp	Data Redundan Komputasi

**(c) Desain Ulang Data**

Fase ini data mahasiswa.MA\_IPS, mahasiswa.MA\_IPK, mahasiswa.MA\_SKSLulus, mahasiswa.MA\_SKSTempuh dan Kelas.Ke\_terisi akan dihilangkan karena bersifat redundan. Sebagai gantinya akan dibuatkan fungsi agregasi untuk menampilkan data tersebut. Dalam kasus ini tabel \_temp dapat dihilangkan pada desain basis data tujuan.

Desain ulang data yang dilakukan di basis data tujuan adalah mendesain basis data sesuai dengan desain basis data warisan. Hanya saja dalam Rekayasa Ulang Iteratif ini desain basis data tujuan dibedakan oleh klasifikasi data primer dan data residual saja. Desain hasil rekayasa ulang ini pun sudah ditambahkan relasi yang memungkinkan konsistensi data di basis data tujuan.

**(d) Mengadaptasikan 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. Fungsi baru tersebut akan ditempatkan dalam file datalocator.asp sehingga memudahkan akses basis data.

Dalam kasus Sistem Informasi Akademik FTIF ITS Surabaya terdapat beberapa fungsi yang terdapat di datalocator.asp untuk aplikasi warisan. datalocator.asp merupakan komponen baru yang dibuat agar aplikasi warisan yang mengakses data residual dapat mendapatkan data tanpa mengakses basis data residual. Fungsi-fungsi yang terdapat di datalocator.asp seperti yang dilihat pada Tabel 2.

**Tabel 2. Fungsi yang berada di datalocator.asp**

Nama Fungsi	Fungsi	Tabel yang berhubungan
getIPSLalu()	Menghasilkan IPS Mahasiswa	kuliah, matakuliah, val_nilai
getIPK()	Menghasilkan IPK Mahasiswa	Kuliah, MataKuliah, val_nilai, NilaiMinimalLulus
getIPKMhsEquivalensi()	Menghasilkan IPK hasil ekivalensi	Kuliah, MataKuliah, val_nilai, NilaiMinimalLulus
getSKSLulus()	Memproses jumlah SKS lulus	MataKuliah
getSKSTempuh()	Mendapatkan Jumlah SKS tempuh	MataKuliah
getJumlahPesertaKelas()	Menghasilkan Jumlah Peserta Kelas	Kuliah

**(e) Perpindahan Data**

Proses perpindahan data dilakukan dengan memanfaatkan fasilitas DTS (Data Transformation Service) pada SQL Server. Pada proses ini dilakukan pemindahan data dari SQL Server ke Oracle. Dalam implementasinya, keseluruhan tabel dipindahkan dari basis data warisan ke basis data termasuk data residual yang ada. Dalam hal ini dapat dilakukan pemindahan data tabel secara sekaligus, karena pada dasarnya pemindahan data tidak mempengaruhi basis data warisan yang masih dipakai oleh aplikasi warisan.

Setelah tabel dipindahkan ke basis data tujuan, basis data tujuan berisi struktur tabel yang sama dengan basis data warisan. Selanjutnya ditambahkan relasi-relasi serta *primary key* untuk menjaga konsistensi data. Dalam hal ini basis data yang ada siap untuk digunakan oleh komponen warisan yang siap direkayasa ulang.

**(f) Desain Ulang Fungsi**

Dalam pengerjaan desain ulang fungsi ini tiap komponen utama akan dibuatkan dua kelas. Kelas pertama bertanggung jawab menangani masalah tampilan dan kelas *control* yang bertanggung jawab menangani masalah fungsionalitas. Untuk kemudahan pemeriksaan kesamaan fungsi, penamaan file pada sistem tujuan akan disamakan dengan file yang lama, misalkan frs.asp menjadi frs.aspx. Untuk kelas *control* apabila diperlukan akan dinamakan CIsfrs.vb.

Dalam kasus Sistem Informasi Akademik FTIF ITS Surabaya terdapat beberapa proses penting dalam desain ulang fungsi di sistem

Desain ulang fungsi dalam masing-masing proses adalah sebagai berikut:

1. Data Banker

Data banker adalah kelas yang dibuat untuk menyediakan akses basis data bagi aplikasi yang direkayasa ulang.

2. Manajemen session  
 Dalam interaksi antara sistem warisan dan sistem tujuan diperlukan manajemen session yang dibuat oleh aplikasi yang sudah untuk digunakan oleh aplikasi warisan.
3. Desain Ulang Sistem Login  
 Pengecekan login dan penyimpanan session untuk dipakai aplikasi warisan
4. Desain Ulang Sistem FRS  
 Pembuatan kelas dalam sistem yang baru dengan fungsi yang berhubungan dengan proses FRS
5. Desain Ulang Sistem Perwalian  
 Pembuatan kelas dalam sistem yang baru dengan fungsi yang berhubungan dengan proses Perwalian
6. Desain Ulang Sistem Header  
 Pembuatan kelas dalam sistem yang baru dengan fungsi yang berhubungan dengan menu yang ada di header

**(g) Tes Ekivalensi**

Tes ekivalensi adalah pengecekan hasil rekayasa ulang apakah terdapat kesalahan atau ketidaksesuaian dengan aplikasi lama. Untuk melakukan pengetesan, komponen yang baru diletakkan dalam domain warisan dalam hal ini ASP. Dalam konteks ini, file default.aspx menggantikan file default.asp, dan frs.aspx menggantikan file frs.asp dan jika fungsi-fungsi didalamnya memiliki kesamaan, maka tes ekivalensi dapat dikatakan berhasil. Dalam proses ini dilakukan pembekuan komponen lama untuk sementara waktu. Dan jika berhasil maka file default.asp dan file frs.asp sudah direkayasa ulang.

**(h) Membersihkan Data Residual**

Pada implementasi penelitian ini tempat data residual berada pada basis data tujuan bersamaan dengan data primer. Data residual disimpan dalam tempat yang sama dengan data primer, namun hanya dianggap sebagai catatan bahwa data tertentu merupakan data residual yang pada akhir proses rekayasa ulang akan dihapus karena tidak ada komponen warisan yang menggunakannya.

**(i) Iterasi**

Proses kembali kepada analisis sistem warisan dan melakukan hal yang sama berdasarkan pada hal yang sudah dilakukan pada tahap sebelumnya, karena proses Rekayasa Ulang Iteratif hanya melibatkan komponen kecil dan data yang sedikit pada setiap proses iterasinya.

**(j) Membersihkan Metadata**

Pada implementasi penelitian ini metadata tidak perlu dicatat, karena baik basis data warisan dalam hal ini SQL Server maupun basis data tujuan Oracle telah mencatat sistem metadatanya sendiri. Sehingga jika semua sistem telah direkayasa ulang dan basis data warisan sudah tidak dipakai lagi, secara otomatis metadatanya juga sudah terhapus.

**5. IMPLEMENTASI DAN UJI COBA**

Implementasi disini berkaitan dengan hal-hal yang bersifat implementatif dalam Rekayasa Ulang Iteratif yaitu :

1. Mengadaptasikan sistem warisan
2. Perpindahan data
3. Rekayasa ulang prosedur/komponen
4. Tes ekivalensi yang merupakan bagian dari uji coba di setiap tahap iterasi.

Tahapan implementasi ini diterapkan pada beberapa sub sistem, yaitu:

1. Sub Sistem Login

- Adaptasi Sistem Warisan
  - a. Mengubah query yang mendukung sql server menjadi mendukung oracle. Perubahan query dalam sistem login pada aplikasi warisan dan aplikasi yang sudah direkayasa ulang seperti dilihat pada Gambar 4. dan Gambar 5.

```
"SELECT DISTINCT HAP_KodeJurusan, PS_Nama FROM
HakAksesPegawai, ProgramStudi WHERE
PS_FA_ID+PS_JU_ID+PS_ID=HAP_KodeJurusan AND
HAP_PE_Nip="" +NIP+ "" AND HAP_GU_ID=""
&GU_ID&"" ORDER BY HAP_KodeJurusan"
```

**Gambar 4. Query pada aplikasi warisan SQL Sever**

```
"SELECT DISTINCT HAP_KodeJurusan, PS_Nama FROM
HakAksesPegawai, ProgramStudi WHERE
PS_FA_ID||PS_JU_ID||PS_ID=HAP_KodeJurusan AND
HAP_PE_Nip="" +NIP+ "" AND HAP_GU_ID=""
&GU_ID&"" ORDER BY HAP_KodeJurusan"
```

**Gambar 5 Query pada aplikasi tujuan Oracle**

- b. Mengubah conection string sehingga dapat mengakses basis data baru
- Perpindahan data
 

Memindahkan data Mahasiswa, Pegawai, Group User, HakAksesPegawai ke basis data tujuan
- Rekayasa ulang prosedur/komponen
 

Permbuatan kelas Data Banker, pembuatan manajemen session, perubahan kode program pada login yaitu prosedur SimpanHakAksesJurusan dan SimpanArrayHakAksesGroup.
- Tes ekivalensi yang merupakan bagian dari uji coba di setiap tahap iterasi.

Pengetesan dengan memasukkan ID (NRP/NIP) dan password, apabila berhasil, maka masuk ke halaman utama.asp.

2. Sub Sistem FRS

- Adaptasi Sistem Warisan
  - a. Pembuatan datalocator untuk program yang mengakses data residual
  - b. Perubahan connection string
  - c. Perubahan sintaks sql sehingga mendukung oracle
- Perpindahan data
 

Memindahkan tabel \_sekarang, Kelas, Kuliah, Matakuliah, Pegawai
- Rekayasa ulang prosedur/komponen
 

Pembuatan kelas ClsFrs untuk mengatur fungsionalitas FRS
- Tes ekivalensi yang merupakan bagian dari uji coba di setiap tahap iterasi.
 

Pengetesan yang dilakukan pada semua fungsi yang mengatur fungsionalitas FRS seperti penambahan mata kuliah, pengecekan fungsi frsdisetujui, pengecekan pembayaranSPP.

3. Sub Sistem Perwalian

- Adaptasi Sistem Warisan
  - a. Menggantikan data residual komputasi untuk akses terhadap data redundan
  - b. Perubahan sintaks sql sehingga mendukung oracle
- Perpindahan data
 

Memindahkan tabel FrsDisetujui ke basis data tujuan
- Rekayasa ulang prosedur/komponen
 

Pembuatan kelas clsPerwalian untuk menangani fungsionalitas perwalian dengan prosedur menampilkan data anak wali dan da nrp sebagai parameter link
- Tes ekivalensi yang merupakan bagian dari uji coba di setiap tahap iterasi.
 

Pengetesan fungsionalitas sistem dalam menampilkan data anak wali beserta link-link yang menyertainya

4. Sub Sistem Header

- Adaptasi Sistem Warisan
 

Pengubahan connection string
- Perpindahan data
 

Pemindahan tabel Modul, ModulGroup, dan HakAksesModul ke basis data tujuan
- Rekayasa ulang prosedur/komponen
 

Merekayasa ulang komponen yang telah direkayasa ulang yaitu header.aspx untuk menggantikan komponen warisan yaitu \_header.asp. Header yang telah direkayasa ulang ini tetap memiliki fungsi menampilkan pilihan modul dan menampilkan data UID.
- Tes ekivalensi yang merupakan bagian dari uji coba di setiap tahap iterasi.
 

Uji Coba dilakukan dengan membandingkan apakah prosedur dalam header dapat mengubah

tampilan seperti yang juga dilakukan header pada sistem warisan.

Sedangkan untuk sub sistem yang lain tetap mempergunakan yang telah ada dalam sistem warisan.

## 6. KESIMPULAN DAN SARAN

Adapun kesimpulan penelitian ini adalah sebagai berikut.

1. Rekayasa ulang yang telah dilakukan telah menunjukkan bahwa sistem dapat direkayasa ulang sedikit demi sedikit, dengan melibatkan komponen yang kecil dan bertahap dengan proses yang telah dijabarkan walaupun hanya sub sistem login, perwalian, dan frs.
2. Dalam Rekayasa Ulang Iteratif diperlukan analisis platform sistem warisan terlebih dahulu sebelum menentukan sistem tujuan agar sistem warisan dan sistem yang sedang dalam proses rekayasa ulang dapat berkomunikasi dan berjalan bersama
3. Perpindahan data yang dilakukan dalam rekayasa ulang ini dilakukan ternyata juga dapat dilakukan dengan duplikasi data. Duplikasi ini dilakukan dengan perpindahan tabel per tabel dengan penambahan struktur data berupa *constraint* seperti *primary key*.

Saran dari penelitian ini adalah sebagai berikut.

1. Perlu dilakukan studi banding mengenai metode rekayasa ulang lain yang pernah ada sebelumnya yaitu Chicken Little Strategy dan Butterfly Methodology.
2. Perlu penelitian lebih lanjut mengenai Rekayasa Ulang Iteratif terutama yang berkaitan dengan analisis data di sistem warisan, karena pada Tugas Akhir ini menitikberatkan pada saling berinteraksinya dua sistem, yaitu sistem warisan dan sistem tujuan.

## 7. DAFTAR PUSTAKA

1. Bianchi, Alessandro, Danilo Caivano, Society, Vittorio Marengo, *Iterative Reengineering of Legacy Systems*, IEEE Transactions on Software Engineering, 2003.
2. Sommerville, Ian, *Software Engineering*, 5th edition. Chapter 34, 1995.
3. G.Visaggio, "Ageing of a Data Intensive Legacy System: Symptoms and Remedies," Journal on Software Maintenance and Evolution, vol. 13, no. 5, pp. 281-308, 2001.