

PREDIKSI PROPAGASI PERUBAHAN BERKAS PROYEK PERANGKAT LUNAK BERDASARKAN PENGGALIAN DATA REPOSITORI

Lutfi Rizal Gozali¹⁾, Jauari Akhmad Nur Hasim²⁾

¹⁾Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Jawa Timur

²⁾Jurusan Teknik Informatika, Departemen Teknik Informatika dan Komputer, Politeknik Elektronika Negeri
Surabaya, Jawa Timur

Email : lutfi.rizal12@mhs.if.its.ac.id¹⁾, jauari@eepis-its.edu²⁾

ABSTRAK

Repositori perangkat lunak menyimpan informasi terkait perubahan-perubahan yang dilakukan pengembang selama fase pengembangan dan evolusi perangkat lunak. Dalam perjalanannya, pergantian atau penambahan anggota tim pengembang seringkali terjadi pada sebuah proyek perangkat lunak. Untuk dapat beradaptasi secara cepat, anggota pengembang baru harus mempelajari keseluruhan sistem melalui dokumentasi dan kode sumber proyek serta diskusi dengan anggota tim pengembang lainnya. Penelitian terkait sebelumnya memiliki fokus pada ekstraksi dokumen proyek. Pada penelitian tersebut, setiap catatan perubahan yang dilakukan pengembang atas proyek dibuat ke dalam dokumen baru untuk dimanfaatkan lebih jauh. Namun, masih terdapat kekurangan pada informasi yang digali. Kekurangan tersebut masih menimbulkan kesulitan bagi pengembang. Penelitian ini memberikan alternatif dalam memudahkan adaptasi anggota pengembang baru dalam memahami kondisi terakhir pengembangan perangkat lunak melalui informasi propagasi perubahan berkas proyek perangkat lunak. Berdasarkan hasil pengujian pada beberapa data riwayat perubahan repositori, pola kecenderungan perubahan pada berkas proyek perangkat lunak dapat ditemukan dengan menggunakan teknik penggalian aturan asosiatif. Pola kecenderungan perubahan berkas ini selanjutnya dapat disajikan sebagai prediksi propagasi perubahan berkas proyek perangkat lunak dalam bentuk pohon propagasi perubahan berkas.

Kata Kunci: *Propagasi Perubahan Berkas, Penggalian Aturan Asosiatif, Repositori Perangkat Lunak.*

1. PENDAHULUAN

Proses pengembangan perangkat lunak menjadi bagian utama setelah sebuah produk perangkat lunak dioperasikan. Bagian kontrol dan pengawasan merupakan hal penting dalam mengetahui apakah sebuah perangkat lunak mampu dimasukkan dalam kategori berkembang atau tidak. Manajer proyek tentunya tidak hanya melakukan kontrol dan pengawasan pada sebuah proyek saja. Untuk melakukan proses tersebut, manajer proyek perlu melihat satu per satu proyek melalui dokumen yang ada. Setiap proyek tentunya memiliki dokumen yang berbeda. Semakin banyak proyek yang ditangani, waktu yang dibutuhkan semakin lama untuk melakukan proses tersebut. Hal itu, salah satunya, tergantung pada skala proyek yang dikerjakan. Selain itu, mempelajari sebuah dokumen proyek bukanlah hal yang mudah. Mempelajari proyek tidak hanya dilakukan oleh manajer proyek tetapi juga dilakukan oleh tim pengembang lainnya. Hal ini dilakukan untuk melanjutkan pengembangan perangkat lunak tersebut, baik dengan tim yang lama maupun dengan tim baru. Salah satu hal yang memudahkan untuk melakukan proses tersebut adalah meletakkan

dokumen tersebut ke dalam repositori perangkat lunak.

Banyak peneliti dan pengembang perangkat lunak melakukan penelitian mengenai repositori perangkat lunak. Server repositori semakin banyak bermunculan dengan menonjolkan keunggulannya masing-masing. Hal itu tentunya tidak terlepas dari sebuah aturan dan prinsip pengembangan perangkat lunak. Salah satu standar dalam sebuah pengembangan proyek perangkat lunak adalah adanya sebuah dokumen yang menunjukkan beberapa identitas perangkat lunak tersebut yang dapat disimpan ke dalam repositori. Repositori perangkat lunak dapat digunakan untuk membantu pengembang dalam melakukan manajemen proyek perangkat lunak. Seiring dengan berevolusinya perangkat lunak yang dikerjakan, perubahan-perubahan yang dilakukan pengembang terekam dalam repositori tersebut. Dari rekaman perubahan tersebut, sejumlah informasi terkait perubahan proyek perangkat lunak dapat diketahui. Informasi tersebut antara lain sebagai berikut:

- nama berkas yang mengalami perubahan;
- tanggal dan jam perubahan dilakukan;
- identitas pengembang yang melakukan perubahan; dan

- d. penjelasan terkait perubahan yang dilakukan pengembang.

Selama proses pengembangan dan evolusi perangkat lunak tersebut, pergantian atau penambahan anggota tim pengembang seringkali terjadi. Hal ini menuntut anggota pengembang baru segera dapat beradaptasi dengan tim dan memahami kondisi terakhir proyek perangkat lunak yang akan dikerjakan. Dengan memanfaatkan repositori tersebut, kolaborasi antaranggota tim pengembang dan penelusuran setiap perubahan dari masing-masing anggota tim pengembang dapat diketahui.

Dari informasi di atas, muncul sebuah gagasan untuk melakukan proses penggalian informasi dari proyek perangkat lunak tersebut pada repositori. Penelitian terkait sebelumnya memiliki fokus pada ekstraksi dokumen proyek. Pada penelitian tersebut, setiap catatan perubahan yang dilakukan pengembang atas proyek dibuat ke dalam dokumen baru untuk dimanfaatkan lebih jauh [1]. Namun, masih terdapat kekurangan pada informasi yang digali. Kekurangan tersebut masih menimbulkan kesulitan bagi pengembang, seperti keterkaitan sebuah fungsi dengan fungsi lainnya, ketergantungan sebuah fungsi ke fungsi lain yang saling berhubungan, dan pengaruh perubahan itu sendiri.

Oleh karena itu, peneliti mengajukan sebuah metode yang dapat memberikan alternatif dalam membantu anggota pengembang baru dalam memahami proyek perangkat lunak yang akan dikerjakan. Fokus penelitian ini adalah mengembangkan sebuah metode yang dapat memberikan prediksi terkait propagasi perubahan berkas proyek perangkat lunak berdasarkan penggalian data pada repositori. Dengan informasi prediksi propagasi perubahan berkas tersebut, proses adaptasi anggota pengembang baru, dalam hal memahami proyek perangkat lunak yang akan dikerjakan, diharapkan dapat dilakukan dengan lebih mudah.

2. KAJIAN PUSTAKA

Berikut ini adalah beberapa kajian pustaka terkait pengembangan metode prediksi propagasi perubahan berkas proyek perangkat lunak berdasarkan penggalian data repositori.

2.1 Propagasi Perubahan Perangkat Lunak

Salah satu aktivitas dalam proses evolusi perangkat lunak ketika terdapat sebuah permintaan perubahan perangkat lunak adalah propagasi perubahan (*change propagation*) [2]. Ahmed E. Hassan dan Richard C. Holt mendefinisikan propagasi perubahan sebagai perubahan-perubahan yang perlu dilakukan pada entitas-entitas lain dari sistem perangkat lunak untuk menjamin konsistensi setelah adanya perubahan pada suatu entitas dari sistem perangkat lunak tersebut [3]. Dalam penelitian

tersebut, propagasi perubahan diprediksi pada tingkat entitas variabel dan fungsi dengan menggunakan pendekatan heuristik. Pengujiannya dilakukan dengan melibatkan riwayat pengembangan dari lima sistem perangkat lunak sumber terbuka berskala besar.

Pada penelitian ini, prediksi propagasi perubahan dilakukan pada tingkat berkas proyek perangkat lunak yang tersimpan pada sebuah repositori. Dengan metode prediksi propagasi perubahan pada tingkat entitas berkas, diharapkan repositori perangkat lunak yang dapat diproses dengan metode ini tidak terbatas pada bahasa pemrograman tertentu .

2.2 Repositori Perangkat Lunak

Repositori perangkat lunak menyimpan rekaman perubahan dari suatu proyek perangkat lunak. Repositori tersebut dapat berupa sistem kontrol kode sumber (seperti CVS), sistem penelusuran bug (seperti BugZilla), atau arsip komunikasi antarpengembang (seperti mailing list) [4]. Sistem kontrol kode sumber menyimpan rekaman perubahan kode sumber program dari suatu proyek perangkat lunak. Beberapa contoh sistem kontrol kode sumber adalah Atlassian Bitbucket dan SourceForge. Berikut ini beberapa informasi yang dapat diidentifikasi pada setiap rekaman perubahan pada sistem kontrol kode sumber:

- nomor kode perubahan,
- pengembang yang melakukan perubahan,
- tanggal perubahan,
- deskripsi perubahan, dan
- berkas-berkas yang mengalami perubahan.

Pada penelitian yang dilakukan Michael Fischer dan rekan-rekannya, dijelaskan sebuah metode untuk membentuk basis data riwayat rilis perangkat lunak. Basis data tersebut dibentuk dengan mengkombinasikan data riwayat pada repositori kode sumber dengan data riwayat pada sistem penelusuran bug [5].

Pada penelitian ini, repositori perangkat lunak yang digunakan sebagai obyek penelitian adalah sistem kontrol kode sumber. Sisi ketersediaan dari data riwayat perubahan perangkat lunak pada repositori daring, seperti Atlassian Bitbucket dan SourceForge, menjadi pertimbangan utama dipilihnya sistem kontrol kode sumber sebagai obyek penelitian. Dengan menggunakan *web crawler*, data riwayat tersebut akan disimpan ke dalam basis data relasional lokal.

2.3 Penggalian Data

Aktivitas pada proses-proses rekayasa perangkat lunak menghasilkan banyak data. Teknik penggalian data dapat diterapkan pada data-data tersebut untuk menghasilkan informasi pendukung proses-proses rekayasa perangkat lunak [6]. Teknik-teknik pada penggalian data secara umum meliputi klasifikasi, pengelompokan, dan penggalian aturan asosiatif. Teknik klasifikasi dapat digunakan untuk

membangun sebuah model prediksi berdasarkan data latih untuk mengenali kelas dari data uji. Teknik pengelompokan dapat digunakan untuk mengetahui kelompok pada sebuah data transaksi berdasarkan kedekatan nilai data dengan nilai pusat kelompok. Sedangkan teknik penggalian aturan asosiatif dapat digunakan untuk mengetahui korelasi antardata dan pola yang sering terjadi pada suatu data transaksi.

Pada penelitian yang dilakukan oleh Sokratis Kotsiantis dan Dimitris Kannelopoulos, dibahas sejumlah algoritma terkait penggalian aturan asosiatif, kategori-kategori basis data yang dapat diterapkan penggalian aturan asosiatif, dan perkembangan terakhir dari penggalian aturan asosiatif [7]. Pada penelitian yang dilakukan oleh Xiu-yu Zhong, dijelaskan penerapan teknik penggalian aturan asosiatif pada data riwayat aktivitas pengunjung situs penjunan daring untuk mengetahui perilaku pengunjung terkait penjualan barang pada situs tersebut [8].

Pada penelitian ini, teknik penggalian data dilibatkan untuk mengetahui prediksi propagasi perubahan berkas proyek perangkat lunak pada repositori. Dari ketiga teknik tersebut, penggalian aturan asosiatif dapat digunakan untuk mengetahui pola kecenderungan perubahan berkas proyek perangkat lunak berdasarkan data riwayat perubahan pada repositori. Pola kecenderungan perubahan berkas ini selanjutnya akan diproses lebih lanjut untuk kemudian disajikan sebagai prediksi propagasi perubahan berkas proyek perangkat lunak.

3. METODOLOGI

Metode prediksi propagasi perubahan berkas proyek perangkat lunak berdasarkan penggalian data pada repositori perangkat lunak ditunjukkan pada Gambar 1. Metode ini terdiri dari beberapa tahapan, antara lain sebagai berikut:

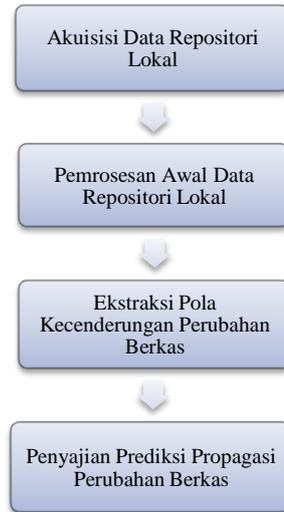
- a. akuisisi data repositori lokal;
- b. pemrosesan awal data repositori lokal;
- c. ekstraksi pola kecenderungan perubahan berkas; dan
- d. penyajian prediksi propagasi perubahan berkas.

3.1 Akuisisi Data Repositori Lokal

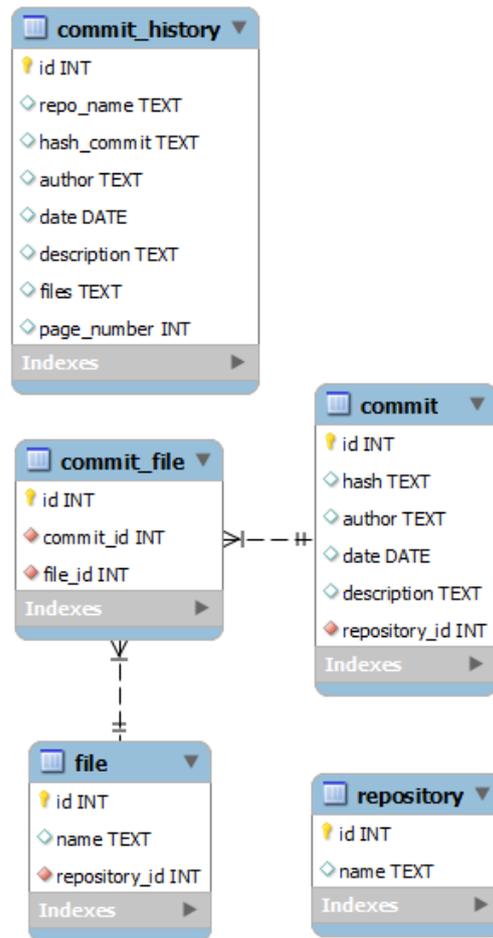
Tahapan ini melibatkan repositori perangkat lunak, dalam hal ini repositori kode program; dan *web crawler* yang membantu mendapatkan data riwayat perubahan dari repositori untuk disimpan ke dalam basis data relasional lokal. Rancangan basis data relasional tersebut ditunjukkan pada Gambar 2.

Pada tahapan ini, *web crawler* mengakuisisi data perubahan meliputi:

- a. nama proyek perangkat lunak;
- b. nomor kode perubahan;
- c. berkas-berkas kode program yang berubah;
- d. nama pengembang yang melakukan perubahan;
- e. tanggal perubahan dilakukan; dan
- f. deskripsi perubahan.



Gambar 1. Metode Prediksi Propagasi Perubahan Berkas



Gambar 2. Rancangan Basis Data Repositori Lokal

Data perubahan tersebut kemudian disimpan pada tabel `commit_history` dari basis data relasional lokal.

Tabel 1. Detail Data Riwayat Perubahan Repositori XYZ-1.0

Kode	Pengembang	Tanggal	Berkas
cz09ge	programmer_1	2013-11-10	a.php;c.php; d.php;
cg10dh	programmer_3	2013-11-11	b.php;c.php; e.php;
ct11ok	programmer_2	2013-11-12	a.php;b.php; c.php;e.php;
dg13df	programmer_2	2013-11-13	b.php;e.php;

3.2 Pemrosesan Awal Data Repositori Lokal Mulus

Tahapan ini melibatkan hasil akuisisi data repositori pada basis data relasional lokal. Contoh detail data riwayat perubahan repositori XYZ-1.0, yakni repositori buatan yang dilibatkan dalam pengujian kotak hitam penelitian ini, ditunjukkan pada Tabel 1. Baris pertama dari contoh detail data riwayat perubahan tersebut memberikan informasi perubahan perangkat lunak XYZ-1.0 dengan kode perubahan cz09ge, yakni programmer_1 melakukan perubahan berkas a.php, c.php, dan d.php pada tanggal 10 Nopember 2013.

Pada tahapan ini, dilakukan identifikasi nama repositori, berkas-berkas yang mengalami perubahan, dan *metadata* rekaman perubahan per nomor kode perubahan. Hasil identifikasi nama repositori disimpan pada tabel *repository*. Hasil identifikasi berkas yang mengalami perubahan disimpan pada tabel *file*. Hasil identifikasi rekaman perubahan per nomor kode perubahan disimpan dalam tabel *commit*. Hasil identifikasi hubungan antardata pada tabel *file* dan tabel *commit* berdasarkan tabel *commit_history* disimpan pada tabel *commit_file*.

3.3 Ekstraksi Pola Kecenderungan Perubahan Berkas

Tahapan ini melibatkan hasil pemrosesan awal data riwayat perubahan pada basis data relasional lokal dan teknik penggalan aturan asosiatif untuk menemukan pola kecenderungan perubahan berkas. Contoh hasil pemrosesan awal data riwayat perubahan dari repositori XYZ-1.0 dengan ambang batas jumlah kemunculan = 2 dan ambang batas kepercayaan = 50% ditunjukkan pada Tabel 2.

Pada tahapan ini, algoritma Apriori, digunakan untuk menggali pola kecenderungan perubahan berkas berdasarkan data riwayat perubahan repositori. Dengan menentukan nilai ambang batas jumlah kemunculan dan nilai ambang batas kepercayaan tertentu, pola kecenderungan perubahan berkas pada repositori dapat dihasilkan. Selanjutnya, pola kecenderungan perubahan berkas ini disajikan sebagai prediksi propagasi perubahan berkas proyek perangkat lunak.

Tabel 2. Hasil Pemrosesan Awal Data Riwayat Perubahan Repositori XYZ-1.0

Tabel repository				
id	name			
1	XYZ-1.0			
Tabel file				
id	name	repo_id		
1	a.php	1		
2	c.php	1		
3	d.php	1		
4	b.php	1		
5	e.php	1		
Tabel commit				
id	hash	author	date	repo_id
1	cz09ge	programmer_1	2013-11-10	1
2	cg10dh	programmer_3	2013-11-11	1
3	ct11ok	programmer_2	2013-11-12	1
4	dg13df	programmer_1	2013-11-13	1
Tabel commit_file				
id	commit_id	file_id		
1	1	1		
2	1	2		
3	1	3		
4	2	4		
5	2	2		
6	2	5		
7	3	1		
8	3	4		
9	3	2		
10	3	5		
11	4	4		
12	4	5		

3.4 Penyajian Prediksi Propagasi Perubahan Berkas

Tahapan ini melibatkan hasil ekstraksi pola kecenderungan perubahan berkas. Contoh hasil ekstraksi pola kecenderungan berkas dari repositori XYZ-1.0 ditunjukkan pada Tabel 3. Baris pertama hasil ekstraksi pola kecenderungan berkas tersebut memberikan informasi propagasi perubahan berkas a.php, yakni jika perubahan dilakukan pada berkas a.php maka 100% perlu dilakukan perubahan pada berkas c.php.

Selanjutnya, pada setiap data berkas proyek perangkat lunak yang mengalami perubahan, ditampilkan pohon propagasi perubahan berkas dengan informasi sebagai berikut:

- Simpul, merepresentasikan berkas yang mengalami perubahan pada repositori. Terdapat dua jenis simpul, yakni induk dan anak. Induk merupakan simpul yang merepresentasikan berkas sumber perubahan. Anak merupakan simpul yang merepresentasikan berkas terkait yang perlu diubah.
- Busur, merepresentasikan propagasi perubahan dari induk ke anak yang dilengkapi dengan persentase tingkat kepercayaan dari propagasi perubahan berkas tersebut.

Tabel 3. Contoh Hasil Ekstraksi Pola Kecenderungan Perubahan Berkas Repositori XYX-1.0

Anteseden	Konsekuensi	Tingkat Kepercayaan
a.php	c.php	100%
b.php	e.php	100%
e.php	b.php	100%
c.php;b.php	e.php	100%
c.php;e.php	b.php	100%
c.php	a.php	66,6666641%
c.php	b.php	66,6666641%
b.php	c.php	66,6666641%
c.php	e.php	66,6666641%
e.php	c.php	66,6666641%
c.php	b.php;e.php	66,6666641%
b.php	c.php;e.php	66,6666641%
e.php	c.php;b.php	66,6666641%
b.php	e.php	66,6666641%

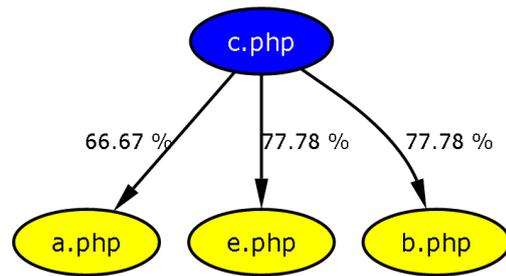
Pohon propagasi perubahan berkas ini dibentuk berdasarkan pola kecenderungan berkas yang telah ditemukan sebelumnya. Berikut ini adalah kode pseudo pembentukan pohon propagasi perubahan berkas:

```

1  set induk;
2  init tTree(`file`, `conf`, `c pattern`);
3  foreach (rPattern in tPattern.Rows)
4  {
5    antecedents = rPattern[`antecedent`];
6    if (antecedents.Consist(induk))
7    {
8      set pattern;
9      set conf;
10     consequents = rPattern[`consequent`];
11     foreach(c in consequents)
12     {
13       if(tTreeItem.Consist(c))
14       {
15         set avgCnf;
16         tTreeItem.UpdateCnfDnce(c, avgCnf);
17         tTreeItem.add(c, avgConf, pattern);
18       }
19     } Else
20     {
21       tTreeItem.add(c, confid, pattern);
22     }
23   }
24   init graph;
25   foreach(rTree in tTree)
26   if (!graph.ExistEdge(rTree))
27   {
28     lblEdge = rTree[`conf`];
29     anak = rTree[`file`];
30     graph.AddEdge(induk, lblEdge, anak);
31   }

```

Sebelum proses pembentukan pohon propagasi perubahan berkas dilakukan, induk ditentukan terlebih dahulu. Ketika proses pembentukan pohon propagasi perubahan berkas berlangsung, anak ditentukan dari konsekuensi pola kecenderungan perubahan berkas. Terdapat dua kemungkinan nilai tingkat kepercayaan propagasi perubahan berkas yang ditampilkan pada busur pohon propagasi perubahan berkas, yakni: nilai *confidence* dan nilai rata-rata *confidence*.



Gambar 3. Contoh Pohon Propagasi Perubahan Berkas c.php dari Repositori XYZ-1.0

Tabel 4. Informasi Detail Propagasi Perubahan Berkas c.php dari Repositori XYZ-1.0

Berkas Terkait	Pola Kecenderungan Perubahan Berkas
a.php	c.php → a.php (66,66666 %)
b.php	c.php;e.php → b.php (100 %)
b.php	c.php → b.php (66,66666 %)
b.php	c.php → b.php;e.php (66,66666 %)
e.php	c.php;b.php → e.php (100 %)
e.php	c.php → e.php (66,66666 %)
e.php	c.php → b.php;e.php (66,66666 %)

Nilai *confidence* digunakan jika induk merupakan anggota anteseden pola kecenderungan perubahan berkas tunggal. Sedangkan rata-rata nilai *confidence* digunakan jika induk merupakan anggota anteseden sejumlah pola kecenderungan perubahan berkas. Contoh pohon propagasi perubahan berkas dengan induk c.php dan beberapa anak, seperti a.php, e.php, dan b.php, ditunjukkan pada Gambar 3. Informasi detail propagasi perubahan berkas c.php ditunjukkan pada Tabel 4.

4. UJICOBAN DAN EVALUASI

Berikut ini dijelaskan mengenai pengujian dan evaluasi metode prediksi propagasi perubahan berkas proyek perangkat lunak berdasarkan penggalan data pada repositori.

4.1 Rancangan Pengujian

Pengujian metode prediksi propagasi perubahan berkas proyek perangkat lunak ini terdiri dari dua jenis pengujian, yakni:

- pengujian kotak hitam penerapan metode prediksi propagasi perubahan berkas; dan
- pengujian kinerja metode prediksi propagasi perubahan berkas pada data repositori nyata.

4.1.1 Pengujian Kotak Hitam Penerapan Metode Prediksi Propagasi Perubahan Berkas

Pengujian ini melibatkan sejumlah skenario yang terbagi dalam masing-masing tahapan metode prediksi propagasi perubahan berkas. Setiap skenario memiliki masukan dan luaran yang diharapkan tertentu. Jika luaran sistem pada suatu tahapan sesuai dengan luaran yang diharapkan pada skenario pengujian maka pengujian pada tahapan tersebut dinyatakan berhasil dan sebaliknya.

Sejumlah prototipe telah dikembangkan untuk menerapkan setiap tahapan dalam metode prediksi propagasi perubahan berkas. Selanjutnya, prototipe-prototipe tersebut diuji dengan pengujian kotak hitam. Berikut ini adalah prototipe-prototipe yang merepresentasikan setiap tahapan dalam metode prediksi propagasi perubahan berkas.

- a. **BRD Crawler dan SFRD Crawler**, prototipe berbasis *desktop* yang merepresentasikan penerapan tahapan akuisisi data repositori lokal. Prototipe ini membantu menyimpan data riwayat perubahan dari repositori yang ada pada halaman Atlassian Bitbucket (BRD Crawler) dan SourceForge (SFRD Crawler) ke basis data relasional lokal.
- b. **FCH Normalizer**, prototipe berbasis *desktop* yang merepresentasikan penerapan tahapan pemrosesan awal data repositori lokal. Prototipe ini membantu melakukan normalisasi tabel `commit_history` pada basis data relasional lokal.
- c. **FCP Miner**, prototipe berbasis *desktop* yang merepresentasikan tahapan ekstraksi pola kecenderungan perubahan berkas dan penyajian prediksi propagasi perubahan berkas. Prototipe ini membantu proses penggalan pola kecenderungan perubahan berkas dengan menerapkan algoritma Apriori dan menyajikan hasil akhir prediksi propagasi perubahan berkas dalam bentuk pohon propagasi perubahan berkas.

Daftar rancangan pengujian kotak hitam penerapan metode prediksi propagasi perubahan berkas ditunjukkan pada Tabel 5. Pada pengujian kotak hitam ini, masing-masing pengujian terdiri dari sejumlah skenario dengan tujuan, masukan, dan target luaran yang telah ditentukan. Hasil pengujian skenario dinyatakan berhasil jika luaran prototipe sesuai dengan target luaran atau luaran yang diharapkan dan sebaliknya. Detail rancangan pengujian kotak hitam proses akuisisi data repositori lokal dari repositori daring Bitbucket ditunjukkan pada Tabel 6.

Tabel 5. Daftar Rancangan Pengujian Kotak Hitam Penerapan Metode Prediksi Propagasi Perubahan Berkas

Prototipe	Pengujian
BRD Crawler	BT-01. Uji coba proses akuisisi data repositori Bitbucket.
SFRD Crawler	BT-02. Uji coba proses akuisisi data repositori SourceForge.
FCH Normalizer	BT-02. Uji coba proses normalisasi basis data lokal repositori buatan.
FCP Miner	BT-03. Uji coba proses penggalan pola kecenderungan perubahan berkas dan pembentukan pohon propagasi perubahan berkas dari repositori buatan.

Tabel 6. Detail Rancangan Pengujian Kotak Hitam Proses Akuisisi Data Repositori Daring Bitbucket

BT-01. Uji Coba Proses Akuisisi Data Repositori Bitbucket			
Tujuan	Menyimpan lima data terakhir riwayat perubahan dari repositori Bitbucket ke basis data relasional lokal.		
BT-01.1. Skenario 1: Alamat repositori tersedia pada Atlassian Bitbucket dan dapat diakses secara publik			
Masukan	a. URL: https://bitbucket.org/tortoisehg/thg/ b. Halaman mulai: 1		
Luaran yang diharapkan			
No.	Kode	Pengembang	Berkas Berubah
1	7af5474	Yuya Nishihara	merge.py
2	0137e51	Yuya Nishihara	backout.py
3	d82a733	Yuya Nishihara	compress.py
4	5736665	Yuya Nishihara	compress.py
5	c5038b6	Yuya Nishihara	compress.py
BT-01.2. Skenario 2: Alamat repositori tersedia pada Atlassian Bitbucket dan tidak dapat diakses secara publik (privat)			
Masukan	a. URL: https://bitbucket.org/lutfirizal/pcis/ b. Halaman mulai: 1		
Luaran yang diharapkan	Tidak ada data riwayat perubahan yang disimpan.		
BT-01.3. Skenario 3: Alamat repositori tidak tersedia pada Atlassian Bitbucket			
Masukan	a. URL: https://bitbucket.org/lutfirizal/fcpanalyzer/ b. Halaman mulai: 1		
Luaran yang diharapkan	Tidak ada data riwayat perubahan yang disimpan.		

Detail rancangan pengujian kotak hitam proses akuisisi data repositori lokal dari repositori daring SourceForge ditunjukkan pada Tabel 7. Detail rancangan pengujian kotak hitam normalisasi basis data relasional lokal dari repositori buatan ditunjukkan pada Tabel 8. Sedangkan detail rancangan pengujian kotak hitam proses penggalan pola kecenderungan perubahan berkas pada repositori buatan dan proses pembentukan pohon propagasi perubahan berkas sesuai pola kecenderungan perubahan berkas repositori buatan ditunjukkan pada Tabel 9.

4.1.2 Pengujian Kinerja Metode Prediksi Propagasi pada Data Repositori Nyata

Pengujian ini melibatkan data riwayat perubahan beberapa proyek perangkat lunak sumber terbuka pada repositori Atlassian Bitbucket yang dapat diakses publik dan repositori SourceForge, antara lain sebagai berikut:

- a. **Whoosh** [9], repositori ini tersedia daring di situs web Atlassian Bitbucket. Repositori ini memiliki 1.567 riwayat perubahan mulai dari tanggal 19 Februari 2008 sampai dengan

Tabel 7. Detail Rancangan Pengujian Kotak Hitam Proses Akuisisi Data Repositori Daring SourceForge

BT-02. Uji Coba Proses Akuisisi Data Repositori SourceForge			
Tujuan	Menyimpan lima data pertama riwayat perubahan dari repositori SourceForge ke basis data relasional lokal.		
BT-02.1. Skenario 1: Alamat repositori tersedia pada SourceForge			
Masukan	a. URL: http://sourceforge.net/p/primitives2d/ b. No. <i>commit</i> mulai: 1		
Luaran yang diharapkan			
No.	Kode	Pengembang	Berkas Berubah
1	[r1]	jcpmcDonald	
2	[r2]	jcpmcDonald	Primitives2D.csproj; Primitives2D.cs; AssemblyInfo.cs; Primitives2D.sln
3	[r3]	gtexmo	Primitives2D.cs
4	[r4]	gtexmo	Primitives2D.cs
5	[r5]	gtexmo	Primitives2D.cs
BT-02.2. Skenario 2: Alamat repositori tidak tersedia pada SourceForge			
Masukan	a. URL: http://sourceforge.net/p/fcpanalyzer/ b. No. <i>commit</i> mulai: 1		
Luaran yang diharapkan	Tidak ada data riwayat perubahan yang disimpan.		

Tabel 8. Detail Rancangan Pengujian Kotak Hitam Proses Normalisasi Basis Data dari Repositori Buatan

BT-03. Uji Coba Proses Normalisasi Basis Data Repositori Buatan	
Tujuan	Melakukan normalisasi tabel <code>commit_history</code> yang menyimpan data repositori buatan.
BT-03.1. Skenario 1: Terdapat data pada tabel <code>commit_history</code>	
Masukan	Seperti yang ditunjukkan pada Tabel 1.
Luaran yang diharapkan	Seperti yang ditunjukkan pada Tabel 2.

tanggal 06 Nopember 2013 dan berkas yang mengalami perubahan berjumlah 325 berkas.

- b. **MoinMoin** [10], repositori ini tersedia daring di situs web Atlassian Bitbucket. Repositori ini memiliki 2.301 riwayat perubahan mulai dari tanggal 25 Desember 2010 sampai dengan tanggal 30 Oktober 2013 dan berkas yang mengalami perubahan berjumlah 904 berkas.
- c. **NastySteroid** [11], repositori ini tersedia daring di situs web SourceForge. Repositori ini memiliki 1.064 riwayat perubahan mulai dari tanggal 2 Nopember 2010 sampai dengan tanggal 27 Juli 2013 dan berkas yang mengalami perubahan berjumlah 625 berkas.
- d. **iTextSharp** [12], repositori ini tersedia daring di situs web SourceForge. Repositori ini memiliki 655 riwayat perubahan mulai dari tanggal 20 Juli 2008 sampai dengan tanggal 25

Tabel 9. Detail Rancangan Pengujian Kotak Hitam Proses Penggalian Pola Kecenderungan Perubahan Berkas dan Pembentukan Pohon Propagasi Perubahan Berkas

BT-04. Uji Coba Proses Penggalian Pola Kecenderungan Perubahan Berkas dan Pembentukan Pohon Propagasi Perubahan Berkas	
Tujuan	Mendapatkan pola kecenderungan perubahan berkas dari data repositori buatan dan membentuk pohon propagasi perubahan berkas berdasarkan pola tersebut.
BT-04.1. Skenario 1: Melakukan penggalian pola kecenderungan perubahan berkas dengan ambang batas jumlah kemunculan dan ambang batas <i>confidence</i> tertentu pada repositori buatan	
Masukan	Ambang batas <i>support count</i> =2 dan ambang batas <i>confidence</i> =50%.
Luaran yang diharapkan	Seperti yang ditunjukkan pada Tabel 3.
BT-04.2. Skenario 2: Memilih berkas yang mengalami perubahan untuk mengetahui pohon propagasi perubahan berkas	
Masukan	Memilih berkas yang mengalami perubahan.
Luaran yang diharapkan	Seperti yang ditunjukkan pada Gambar 3.

Tabel 10. Hasil Pengujian Kotak Hitam Aplikasi yang Menerapkan Metode Prediksi Propagasi Perubahan Berkas Proyek Perangkat Lunak

Prototipe	Pengujian	Hasil
BRD Crawler	BT-01.1	Berhasil
BRD Crawler	BT-01.2	Berhasil
BRD Crawler	BT-01.3	Berhasil
SFRD Crawler	BT-02.1	Berhasil
SFRD Crawler	BT-02.2	Berhasil
FCH Normalizer	BT-03.1	Berhasil
FCP Miner	BT-04.1	Berhasil
FCP Miner	BT-04.2	Berhasil

Nopember 2013 dan berkas yang mengalami perubahan berjumlah 1.240 berkas.

Pada pengujian ini, diamati sejumlah variabel bebas dan variabel terikat. Variabel-variabel bebas tersebut antara lain sebagai berikut:

- a. Jumlah riwayat perubahan yang diproses.
- b. Jumlah berkas yang berubah.
- c. Batas atas, batas bawah, rata-rata jumlah kemunculan L1.
- d. Nilai ambang batas jumlah kemunculan yang digunakan untuk melakukan penggalian pola kecenderungan perubahan berkas.
- e. Nilai ambang batas kepercayaan yang digunakan untuk melakukan penggalian pola kecenderungan perubahan berkas.

Sedangkan variabel-variabel terikat tersebut antara lain sebagai berikut.

Tabel 11. Hasil Pengujian Kinerja Metode Prediksi Propagasi Perubahan Berkas Proyek Perangkat Lunak

1. Whoosh		
Masukan	Jumlah data riwayat perubahan	1.567
	Jumlah berkas yang berubah	325
	Batas atas jumlah kemunculan L1	258
	Batas bawah jumlah kemunculan L1	1
	Rata-rata jumlah kemunculan L1	20,33
	Ambang batas jumlah kemunculan	75
	Ambang batas kepercayaan	50%
Luaran	Jumlah pohon propagasi	1
	Waktu yang dibutuhkan dalam penggalan pola kecenderungan perubahan berkas	1 menit 42 detik
2. MoinMoin		
Masukan	Jumlah data riwayat perubahan	2.301
	Jumlah berkas yang berubah	904
	Batas atas jumlah kemunculan L1	381
	Batas bawah jumlah kemunculan L1	1
	Rata-rata jumlah kemunculan L1	14,87
	Ambang batas jumlah kemunculan	50
	Ambang batas kepercayaan	50%
Luaran	Jumlah pohon propagasi	9
	Waktu yang dibutuhkan dalam penggalan pola kecenderungan perubahan berkas	4 menit 20 detik
3. NastySteroid		
Masukan	Jumlah data riwayat perubahan	1.064
	Jumlah berkas yang berubah	625
	Batas atas jumlah kemunculan L1	155
	Batas bawah jumlah kemunculan L1	1
	Rata-rata jumlah kemunculan L1	7,99
	Ambang batas jumlah kemunculan	20
	Ambang batas kepercayaan	50%
Luaran	Jumlah pohon propagasi	11
	Waktu yang dibutuhkan dalam penggalan pola kecenderungan perubahan berkas	2 menit 3 detik
4. iTextSharp		
Masukan	Jumlah data riwayat perubahan	655
	Jumlah berkas yang berubah	1.240
	Batas atas jumlah kemunculan L1	103
	Batas bawah jumlah kemunculan L1	1
	Rata-rata jumlah kemunculan L1	2.9
	Ambang batas jumlah kemunculan	15
	Ambang batas kepercayaan	30%
Luaran	Jumlah pohon propagasi	2
	Waktu yang dibutuhkan dalam penggalan pola kecenderungan perubahan berkas	3 menit 39 detik

- Jumlah pohon propagasi perubahan berkas yang terbentuk berdasarkan pola kecenderungan perubahan berkas yang ditemukan.
- Waktu yang diperlukan dalam proses penggalan pola kecenderungan perubahan berkas.

4.2 Hasil Pengujian

Hasil pengujian *black-box* dari sejumlah prototipe yang menerapkan tahapan-tahapan dalam metode prediksi propagasi perubahan berkas proyek perangkat lunak ditunjukkan pada Tabel 10.

Sedangkan hasil pengujian kinerja metode prediksi propagasi perubahan berkas proyek perangkat lunak dengan melibatkan data repositori nyata dari beberapa proyek perangkat lunak sumber terbuka yang terdapat pada repositori daring Atlassian Bitbucket dan SourceForge ditunjukkan pada Tabel 11.

4.3 Analisis Hasil Pengujian

Berdasarkan hasil pengujian kotak hitam metode prediksi propagasi perubahan berkas dapat diketahui bahwa penggalan aturan asosiatif dapat dilibatkan untuk menemukan pola kecenderungan perubahan berkas dengan nilai ambang batas jumlah kemunculan dan nilai ambang batas kepercayaan tertentu. Pola kecenderungan perubahan berkas ini selanjutnya dapat disajikan sebagai prediksi propagasi perubahan berkas dalam bentuk pohon propagasi perubahan berkas.

Namun demikian, konfigurasi nilai ambang batas jumlah kemunculan dan nilai ambang batas kepercayaan cukup sulit ditentukan untuk mendapatkan pola kecenderungan perubahan berkas. Sebagaimana hasil pengujian performa metode propagasi perubahan berkas yang ditunjukkan pada Tabel 11, tidak ditemukan konfigurasi umum nilai ambang batas jumlah kemunculan dan nilai ambang batas kepercayaan yang dipastikan dapat menghasilkan pola kecenderungan perubahan berkas.

5. KESIMPULAN

Berdasarkan hasil pengujian dan evaluasi metode prediksi propagasi perubahan berkas proyek perangkat lunak berdasarkan penggalan data pada repositori dapat diambil kesimpulan berikut ini:

- Teknik penggalan aturan asosiatif dapat digunakan untuk menggali pola kecenderungan perubahan berkas pada data riwayat perubahan repositori perangkat lunak.
- Pola kecenderungan perubahan berkas dapat diproses lebih lanjut untuk kemudian disajikan pohon propagasi perubahan berkas yang merepresentasikan prediksi propagasi perubahan berkas.
- Konfigurasi umum ambang batas jumlah kemunculan dan kepercayaan tidak dapat ditentukan berdasarkan karakteristik repositori, seperti jumlah data riwayat perubahan, jumlah berkas yang mengalami perubahan, batas atas jumlah kemunculan L1, batas bawah jumlah kemunculan L1, dan rata-rata jumlah kemunculan L1.

6. RENCANA PENELITIAN SELANJUTNYA

Sebagai tindak lanjut dari hasil penelitian ini, penelitian lanjutan akan dilakukan, yakni

penyempurnaan prototipe representasi tahapan metode prediksi propagasi perubahan berkas, terutama prototipe FCP Miner. Pada prototipe FCP Miner, terdapat kekurangan dari sisi waktu komputasi, yakni lamanya waktu penggalan pola kecenderungan perubahan berkas pada data repositori berukuran besar.

7. DAFTAR PUSTAKA

- [1] Sharma, Shalini., Bajpai, Neha (2011), "Project Management Process Repository for Projects Monitoring", *International Journal Of Computer Trends and Technology* 1, 1: 36-45.
- [2] Mens, Tom., Demeyer, Sergey (2008). "Software Evolution". Springer-Verlag Berlin Heidelberg.
- [3] Hassan, E. Ahmed., Holt, Richard C (2004). "Predicting Change Propagation in Software Systems". *International Conference on Software Maintenance. ICSM 2004*.
- [4] Kagdi, Huzefa., Collard, Michael L., Maletic, Jonathan I (2007). "A survey and taxonomy of approaches for mining software repositories in the context of software evolution". *Journal of Software Maintenance and Evolution: Research and Practice* 19: 77-131.
- [5] Fischer, Michael., Pinzger, Martin., Gall, Harald (2003). "Populating a Release History Database from Version Control and Bug Tracking Systems". *Proceedings of the International Conference on Software Maintenance. ICSM 2003*.
- [6] Taylor, Quinn (2010). "Applications of data mining in software engineering". *International Journal Data Analysis Techniques and Strategies* 2, 3: 243-257.
- [7] Kotsiantis, Sotiris., Kanellopoulos, Dimitris. (2006). "Association Rules Mining: A Recent Overview". *GESTS International Transactions on Computer Science and Engineering* 32, 1: 71-82.
- [8] Zhong, Xiu-yu. (2011). "The research and application of web log mining based on the platform weka". *Procedia Engineering* 15: 4073-4078.
- [9] Chaput, Matt. (2008). whoosh, <URL:<https://www.bitbucket.org/mchaput/whoosh/>>.
- [10] Waldmann, Thomas. (2011). moin-2.0, <URL:<https://www.bitbucket.org/thomaswaldmann/moin-2.0/>>.
- [11] Lili, Suhadi., Dibyanugraha, Ariesty., Montolalu, Billy. (2010). Nasty Steroid, <URL:<http://sourceforge.net/p/nastystteroid/>>.
- [12] Hens, Raf., Soares, Paulo., Goosens, Hilde., Chingarev, Alexander., Lowagie, Bruno. (2008). iTextSharp, <URL:<http://sourceforge.net/p/itextsharp/>>.