

MENGGALI METRIK YANG PENTING DALAM RANGKAIAN AKTIVITAS PERUBAHAN PERANGKAT LUNAK UNTUK MEMPREDIKSI BUG DENGAN ATURAN ASOSIASI

Achmad Arwan¹⁾, Siti Rochimah²⁾

¹⁾ Jurusan Teknik Informatika, Universitas Brawijaya, Malang

²⁾ Jurusan Teknik Informatika, Fakultas Teknologi Informasi, ITS Surabaya

Email : ¹⁾arwan@ub.ac.id, ²⁾siti@its-sby.edu

ABSTRAK

Dalam evolusi perangkat lunak terdapat rangkaian aktivitas proses perubahan yang kemudian oleh Nachiapan diformulasikan menjadi metrik-metrik yang mampu memprediksi bug secara presisi. Akan tetapi secara spesifik belum ditemukan seberapa besar pengaruh masing-masing metrik tersebut terhadap hasil prediksi sebuah bug. Penelitian ini mengusulkan pengukuran jumlah kemunculan metrik dengan kemunculan bug pada proyek eclipse. Sehingga dengan penelitian ini dapat diketahui metrik-metrik mana yang penting dalam prediksi kemunculan bug. Aturan asosiasi dalam penggalian data telah dipergunakan secara luas untuk menggali variabel-variabel saling terkait dalam sampel data. Metrik-metrik dalam proyek eclipse kemudian digali dengan aturan asosiasi untuk mendapatkan metrik yang muncul bersama bug. Hasil pengujian menunjukkan bahwa metrik Numberofchangeslate rata-rata kemunculannya sebesar 46,9%, sedangkan metrik Peopletotal muncul bersama bug rata-rata sebesar 43,57% dan metrik Numberofchangesearly memiliki kemunculan rata-rata 14% sehingga ketiga metrik tersebut merupakan metrik yang penting dalam memprediksi bug.

Kata Kunci: Evolusi, Perangkat Lunak, Metrik, Prediksi, Bug.

1 PENDAHULUAN

Proses evolusi perangkat lunak merupakan serangkaian aktivitas yang bertujuan untuk meningkatkan kualitas sebuah perangkat lunak. Perubahan terjadi seiring adanya perubahan kebutuhan organisasi, teknologi, perkembangan organisasi dan sebagainya. Dalam mengembangkan perangkat lunak seringkali terdapat sebuah *bug* akibat dari beberapa faktor. Pada penelitian Nachiapan dkk [1] terdapat rangkaian aktivitas yang mempengaruhi terjadinya sebuah *bug* dalam sebuah perangkat lunak. Rangkaian aktivitas tersebut kemudian dimodelkan menjadi metrik-metrik yang selanjutnya digunakan untuk memprediksi *bug*. Dengan menggunakan metrik-metrik tersebut, *bug* mampu diprediksi dengan tingkat akurasi 91%.

Penelitian ini menganalisis metrik-metrik penting mana yang berkaitan dengan kemunculan *bug* pada proyek Eclipse. Nachiapan menyediakan *dataset* hasil ekstraksi fitur dari proyek Eclipse [2] yang tercatat dalam periode jam, harian dan mingguan. *Dataset* masing-masing terdiri dari 26 atribut, dengan rata-rata jumlah baris sebanyak 6300. Penelitian ini menggunakan data dengan periode harian selama sepuluh hari dengan jumlah data yang digunakan kurang lebih 30000 baris. Dengan

pertimbangan jumlah data yang besar tersebut, maka metode yang cocok menurut Rakesh [3] dalam penelitian ini adalah metode penggalian data menggunakan aturan asosiasi.

Aturan asosiasi menurut Pang [4] adalah metode untuk mencari aturan dalam data dengan mencari apakah ada kemunculan bersama suatu variabel dengan variabel lain dengan tingkat dukungan data minimal tertentu. Penelitian ini menggunakan metode aturan asosiasi untuk mendapatkan metrik-metrik penting mana yang berkaitan erat dengan terjadinya *bug* dalam perangkat lunak pada proyek Eclipse.

2 METRIK RANGKAIAN AKTIVITAS PERUBAHAN PERANGKAT LUNAK, ATURAN ASOSIASI

Nachiapan dkk [1] memperkenalkan metrik-metrik yang digunakan untuk mengukur kegiatan dalam rangkaian perubahan perangkat lunak. Metrik-metrik tersebut masing-masing memiliki definisi pengukuran sendiri. Tabel 1 menunjukkan definisi metrik yang digunakan oleh Nachiapan.

2.1 Aturan Asosiasi

Tabel 1 Definisi Metrik Nachiapan

	Metrik	Penjelasan
1	MaxChurnInBurst	Jumlah maksimal baris kode yang berubah dalam rangkaian perubahan.
2	TotalChurnInBurst	Jumlah keseluruhan baris kode yang berubah dalam semua rangkaian perubahan.
3	ChurnTotal	Total kombinasi yang terjadi pada komponen kode tertentu.
4	TotalPeopleInBurst	Jumlah semua orang yang pernah melakukan perubahan kode.
5	MaxPeopleInBurst	Dari serangkaian perubahan yang terjadi lalu dicari jumlah orang terbanyak yang pernah terlibat.
6	PeopleTotal	Jumlah semua orang yang terlibat dalam pengembangan komponen tersebut.
7	MaximumChangeBurstLate	Jumlah maksimal perubahan pembangunan sebuah kode pada fase 20% sebelum rilis.
8	NumberOfChanges	Jumlah perubahan dalam sebuah kode.
9	TotalBurstSizeLate	Jumlah keseluruhan perubahan yang terjadi pada fase 20% sebelum rilis.
10	NumberOfChangeBurstsLate	Jumlah rangkaian perubahan dalam periode waktu tertentu & jumlah tertentu pada fase 20% sebelum rilis.
11	NumberOfChangesLate	Jumlah berapa kali membangun komponen yang mengalami perubahan pada fase 20% sebelum rilis.
12	NumberOfChangesEarly	Jumlah berapa kali membangun komponen yang mengalami perubahan pada fase 80% dari lama waktu pengerjaan proyek.
13	MaximumChangeBurstEarly	Jumlah maksimal perubahan pembangunan sebuah kode pada fase 80% dari lama waktu pengerjaan proyek.
14	NumberOfChangeBurstsEarly	Jumlah rangkaian perubahan dalam periode waktu tertentu & jumlah tertentu pada fase 80% dari lama waktu pengerjaan proyek.
15	MaximumChangeBurst	Jumlah maksimal perubahan pembangunan sebuah kode.
16	NumberOfConsecutiveChanges Early	Jumlah pembangunan yang berturut-turut terjadi dalam kurun waktu tertentu pada fase 80% dari lama waktu pengerjaan proyek.
17	NumberOfConsecutiveChanges Late	Jumlah pembangunan yang berturut-turut terjadi dalam kurun waktu tertentu pada pada fase 20% sebelum rilis.
18	TotalBurstSizeEarly	Jumlah keseluruhan perubahan yang terjadi pada fase 80% dari lama waktu pengerjaan proyek.
19	TotalBurstSize	Jumlah keseluruhan perubahan yang terjadi.
20	NumberOfConsecutiveChanges	Jumlah pembangunan yang berturut-turut terjadi dalam kurun waktu tertentu.
21	NumberOfChangeBursts	Jumlah rangkaian perubahan dalam periode waktu tertentu & jumlah tertentu.
22	TimeFirstBurst	Waktu kapan pertama terjadi rangkaian perubahan pertama.
23	TimeMaxBurst	Waktu kapan terjadi rangkaian perubahan paling banyak.
24	TimeLastBurst	Waktu kapan terjadi rangkaian perubahan pada rangkaian perubahan terakhir.

Aturan asosiasi merupakan salah satu metode untuk mempelajari pola data yaitu dari kemunculan bersama dari dua atau lebih variabel dengan menerapkan aturan kemunculan minimal tertentu. Penelitian Rakesh [3] mencari aturan dalam prediksi kemunculan suatu produk toko swalayan baik sebagai akibat, sebagai penyebab dan mencari pola kemunculan. Manfaat dari penelitian Rakesh yaitu untuk mengatur penataan letak produk toko swalayan, sehingga memudahkan pembeli dalam membeli produk yang diinginkan.

Sedangkan metode untuk mengevaluasi aturan asosiasi menurut Pang [4] adalah sebagaimana Persamaan 1-5:

$$support = \frac{\sigma\{x,y|x \rightarrow y, x,y \in itemset\}}{t | t \in transaction} \quad (1)$$

$$Confidence(A \rightarrow C) = \frac{sup(A \cup C)}{sup(A)} \quad (2)$$

$$Lift(A \rightarrow C) = \frac{Conf(A \rightarrow C)}{sup(C)} \quad (3)$$

$$Leverage(A \rightarrow C) = sup(A \cup C) - sup(A) \times sup(C) \quad (4)$$

$$Conviction(A \rightarrow C) = \frac{1 - sup(C)}{1 - Conf(A \rightarrow C)} \quad (5)$$

Confidence adalah pengukuran standard dalam aturan asosiasi. *Confidence* mengukur peluang kemunculan bersama antara variabel A dan variabel C dari semua kemunculan variabel A tanpa harus mempertimbangkan variabel mana yang menjadi penyebab dan variabel mana yang menjadi akibat.

Algorithm 3. Apriori

Input: D : database over the set of items j ,
 Output: F : the set of frequent itemsets

- 1: $k = 1; C_k = j$
- 2: while $C_k \neq \emptyset$ do
- 3: support_count(D, C_k)
- 4: for all candidates $c \in C_k$ do
- 5: if $c.support \geq minsup$ then
- 6: $F_k = c$
- 7: end if
- 8: end for
- 9: $C_{k+1} = candidate_generation(F_k)$
- 10: $k = k + 1$
- 11: end while
- 12: $F = \bigcup_{j=1}^k F_j$

Gambar 1. Algoritma apriori

Jadi dalam penelitian ini akan dicari semua variabel yang menyebabkan kemunculan *bug* dengan ambang batas tertentu. Sedangkan *conviction* digunakan untuk mengukur kemunculan variabel penyebab *bug* dengan *bug* akan tetapi dengan mempertimbangkan penyebab dan akibat. Sehingga *bug* akan selalu menjadi variabel akibat, dan metrik yang lain akan selalu menjadi variabel penyebab. Kedua pengukuran tersebut dipilih untuk mengukur oleh karena keduanya mendapatkan peringkat tertinggi dalam proses klasifikasi dalam data menggunakan aturan asosiasi [5].

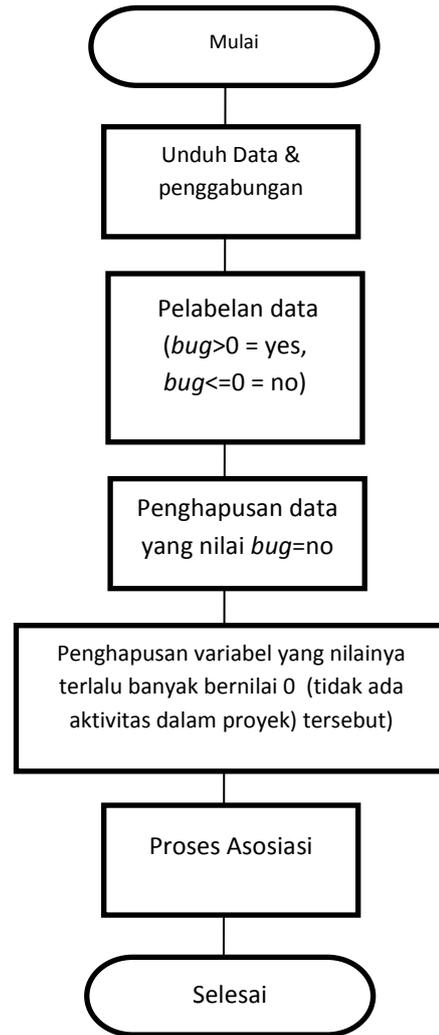
2.2 Algoritma Apriori

Algoritma apriori merupakan algoritma yang digunakan dalam menggali data dengan aturan asosiasi. Menurut Pang [4] apriori berprinsip jika sebuah *item* sering muncul maka semua kombinasi yang mengandung *item* tersebut juga akan sering muncul, begitu juga sebaliknya jika suatu *item* jarang muncul maka kombinasi yang mengandung *item* tersebut juga jarang muncul.

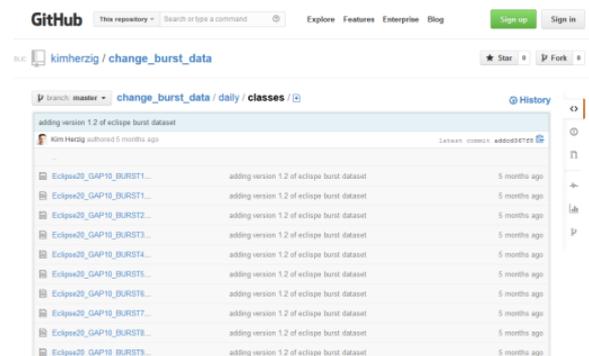
Algoritma apriori menggunakan dukungan minimal dalam mencari frekuensi kemunculan. Semisal frekuensi minimal adalah tiga, maka setiap kali iterasi akan dilakukan proses pengecekan, jika kemunculan pada iterasi tersebut kurang dari tiga maka *item/kombinasi item* akan dihapus dari data untuk iterasi berikutnya. Gambar 1 menunjukkan algoritma apriori.

3 METODOLOGI

Dalam penelitian ini, terdapat beberapa tahapan diantaranya yaitu praproses dan proses. Praproses terdiri dari pengunduhan dan penggabungan data, pelabelan kelas, penghapusan data yang label *bug*-nya adalah 'no', serta penghapusan atribut yang banyak bernilai nol yang artinya tidak ada aktivitas. Proses terdiri dari percobaan mencari aturan asosiasi dengan metrik pengukuran *confidence* dan *conviction*. Gambar 2 menunjukkan tahapan proses dalam penelitian ini.



Gambar 2. Tahapan penelitian.



Gambar 3. Dataset yang digunakan.

3.1 Unduh data dan penggabungan

Pada tahap ini yang pertama adalah mengunduh data set Eclipse. Letak repositori datanya berada di situs repositori GitHub [6]. Data yang diunduh adalah *dataset* periode harian sebanyak sepuluh hari. Gambar 3 menunjukkan *dataset* yang digunakan.

```

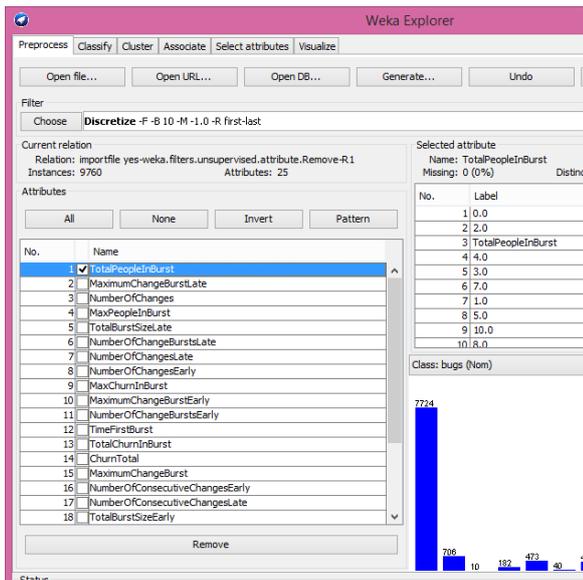
InBurst,TotalBurstSizeLate,NumberCodeBurstsLate,NumberOfChangesLate,NumberOfChai
akpointFilterEditor.java,0,0,12,0,0,0,3,9,0,0,0,-1,0,141,0,7,2,0,0,-1,9,-1,0,4,4
,0,0,2,1,0,0,0,-1,0,10,0,0,1,0,0,-1,1,-1,0,3,0
),0,8,0,0,0,-1,0,224,0,0,7,0,0,-1,7,-1,0,4,1
),0,-1,0,312,0,3,1,0,0,-1,4,-1,0,1,0
tructureVisitor.java,2,0,16,2,0,0,2,14,380,0,1,176,380,391,14,13,1,14,14,176,14,176,1,2,1
2.0.0.0.-1.0.37.0.0.0.0.0.-1.0.-1.0.4.0
    
```

Gambar 4. Data awal.

```

xPeopleInBurst,TotalBurstSizeLate,NumberCodeBurstsLate,NumberOfChangesLate,NumberOfChan
ationBreakpointFilterEditor.java,0,0,12,0,0,0,3,9,0,0,0,-1,0,141,0,7,2,0,0,-1,9,-1,0,4,4,
wa,0,0,3,0,0,0,2,1,0,0,0,-1,0,10,0,0,1,0,0,-1,1,-1,0,3,0,
,0,8,0,0,0,0,0,-1,0,224,0,0,7,0,0,-1,7,-1,0,4,4,
0,3,4,0,0,0,-1,0,312,0,3,1,0,0,-1,4,-1,0,1,0
bstractStructureVisitor.java,2,0,16,2,0,0,2,14,380,0,1,176,380,391,14,13,1,14,14,176,14,176,1,2,4,
    
```

Gambar 5. Data berlabel.



Gambar 6. Data awal yang berisi semua variabel.

3.2 Pelabelan kelas

Selanjutnya adalah proses pelabelan kelas dalam data. Perlu diketahui bahwa dalam data terdapat variabel *bug* dimana isinya adalah jumlah *bug* dalam sebuah kelas. Sehingga jika jumlah *bug* lebih dari nol maka *bug* diberi label 'yes', sedangkan yang tidak diberi label 'no'. Proses pelabelan cukup penting karena dalam aturan asosiasi terutama apriori mengharuskan nilai nomina dimana harus ada perubahan dari numerik menjadi kategorikal/nomina.

3.3 Penghapusan data

Data sebagian besar memiliki label kelas 'no'. Jika dilakukan proses asosiasi maka kemunculan label 'yes' akan kecil sekali, sehingga menyulitkan percobaan. Proses berikutnya adalah penghapusan data, sehingga data yang berlabel 'no' dihapus dari data dan yang ada tinggal yang berlabel 'yes'.

3.4 Penghapusan variabel

Proses selanjutnya adalah penghapusan variabel yang nilainya terlalu banyak bernilai nol. Hal ini

dilakukan agar aturan yang muncul adalah aktivitas yang memiliki nilai. Jika proses ini tidak dilakukan maka akan banyak metrik yang bernilai nol (tidak ada aktivitas) muncul bersama dengan *bug*. Hal ini akan menyulitkan pengambilan kesimpulan karena ternyata dalam data banyak variabel yang bernilai 0.

3.5 Proses Asosiasi

Setelah proses sebelumnya selesai maka berikutnya adalah proses pencarian aturan asosiasi dengan algoritma apriori pada data. Proses pencarian dilakukan dengan menggunakan kakas bantu WEKA. Proses asosiasi dilakukan dengan beberapa metrik evaluasi yaitu *confidence* dan *conviction*. Selain itu juga dilakukan dengan proses pengaturan parameter dalam masing-masing metrik evaluasi tersebut.

4 UJI COBA

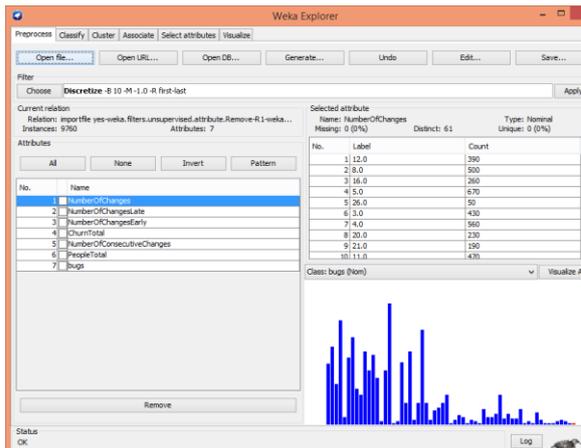
Uji coba dilakukan dengan data yang telah disiapkan. Data bug sepuluh hari digabung menjadi satu file. Kemudian dilakukan proses pelabelan dengan melihat jika variabel *bug* lebih dari nol maka diberi label 'yes' jika nol maka diberi label 'no'. Gambar 4 menunjukkan struktur dan isi data awal. Setelah proses pelabelan, *bug* menjadi memiliki nilai 'yes' atau 'no'. Gambar 5 menunjukkan bentuk data yang sudah berlabel.

Dari 30000 data yang ada hanya menyisakan 9760 data dengan label 'yes' yang akan diproses. Gambar 6 menunjukkan data awal yang berisi semua variabel dari data set tersebut.

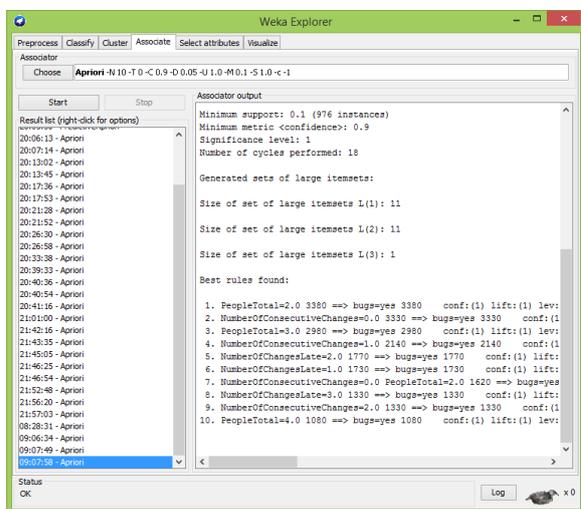
Langkah berikutnya adalah proses pemilihan variabel. Variabel dalam dataset ini ada 26, akan tetapi tidak semua variabel memiliki data yang sesuai. Ada variabel yang terlalu unik seperti *filename* dimana variabel ini berisi nama-nama file dalam proyek *eclipse*. Variabel ini harus dihilangkan karena tidak terlalu penting dan dalam proses asosiasi tidak dapat dilakukan jika dalam data terdapat variabel yang terlalu unik.

Selanjutnya variabel yang lain adalah variabel yang nilainya kurang dari sama dengan nol. Variabel ini dihilangkan karena akan sering muncul tetapi tidak mencerminkan adanya aktivitas karena bernilai nol. Dari hasil proses eliminasi variabel dan data maka hasil variabel yang tersisa tinggal *numberofchanges*, *numberofchangeslate*, *numberofchangesearly*, *churn-total*, *numberofconsecutivechanges*, *peopletotal* and *bugs*. Gambar 7 menunjukkan gambar setelah proses eliminasi data.

Langkah berikutnya adalah proses pencarian aturan dengan aturan asosiasi. Proses pencarian dilakukan dengan menggunakan bantuan WEKA. Beberapa pengukuran dilakukan diantaranya menggunakan metrik *confidence* and *conviction*. Gambar 8 menunjukkan gambar proses pencarian aturan.



Gambar 7. Data setelah penghapusan.



Gambar 8. Pencarian aturan dengan aturan asosiasi.

5 HASIL PERCOBAAN

Selanjutnya dilakukan proses pencarian aturan menggunakan aturan asosiasi. Percobaan dilakukan dengan algoritma apriori dengan metrik *confidence* bervariasi mulai dari 0,6 sampai dengan 0,9 dan metrik *conviction* 1,1 dan 1,2. Gambar 9 adalah gambar hasil percobaannya.

Dari percobaan yang dilakukan, kemudian dilakukan perhitungan kemunculan metrik dalam masing-masing percobaan. Tabel 2 menunjukkan jumlah kemunculan metrik-metrik pada tiap percobaan. Gambar 10 menunjukkan prosentase kemunculan metrik-metrik dalam percobaan.

6 KESIMPULAN DAN PENELITIAN LANJUTAN

Dari hasil percobaan menunjukkan bahwa setelah dilakukan proses eliminasi maka hanya tinggal enam metrik saja yang muncul bersama *bug*. Dengan 9760 data kemudian dilakukan beberapa kali proses pencarian aturan. Metrik *numberofchangeslate* rata-rata memiliki kemunculan sebesar 46,90% dimana metrik ini merupakan metrik yang paling banyak muncul bersama *bug*. Sementara metrik *peopletotal* rata-rata muncul bersama *bug* sebesar 43,57%. Sedangkan untuk metrik *numberofchangesearly* rata-rata kemunculan terkecil yaitu hanya sebanyak 14%. Sehingga kesimpulannya yaitu ketiga metrik tersebut merupakan metrik yang penting diantara metrik-metrik yang lain dalam proyek *Eclipse*.

Penelitian lanjutan dapat dilanjutkan dengan menambahkan dataset dari proyek yang lain untuk meningkatkan keakuratan perhitungan kemunculan metrik-metrik tersebut. Selain itu untuk penelitian selanjutnya juga bisa dilakukan bagaimana memprediksi kemunculan *bug* dengan menggunakan metrik *Peopletotal*, *Numberofchangeslate* maupun metrik *Numberofchangesearly*.

7 DAFTAR PUSTAKA

- [1] Nachiappan Nagappan, Andreas Zellery, Thomas Zimmermann, Kim Herzig, Brendan Murphy "Change bursts as defect predictors." *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on.* IEEE, 2010. p. 309-318.
- [2] https://github.com/kimherzig/change_burst_data
- [3] Rakesh A, Tomasz Imielinski, Arum Swami, "Mining Association rules between sets of Items in large databases", *Proceedings of ACM SIGMOD*, ACM, 1993.
- [4] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining", Addison-Wesley, 2005
- [5] Azevedo, Paulo J., and Alípio M. Jorge. "Comparing rule measures for predictive association rules." *Machine Learning: ECML 2007*. Springer Berlin Heidelberg, 2007. 510-517.
- [6] <http://www.st.cs.uni-saarland.de/softevo/burst-data/eclipse/>

Minimum metric <confidence>: 0.9						Minimum metric <confidence>: 0.7					
Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conf. Level	Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conf. Level
PeopleTotal	2	3380	Bugs	yes	1	PeopleTotal	2	3380	Bugs	yes	1
PeopleTotal	3	2980	Bugs	yes	1	PeopleTotal	3	2980	Bugs	yes	1
NumberOfChangesLate	2	1770	Bugs	yes	1	NumberOfChangesLate	2	1770	Bugs	yes	1
NumberOfChangesLate	1	1730	Bugs	yes	1	NumberOfChangesLate	1	1730	Bugs	yes	1
NumberOfChangesLate	3	1330	Bugs	yes	1	NumberOfChangesLate	3	1330	Bugs	yes	1
PeopleTotal	4	1080	Bugs	yes	1	PeopleTotal	4	1080	Bugs	yes	1
NumberOfChangesEarly	4	1020	Bugs	yes	1	NumberOfChangesEarly	4	1020	Bugs	yes	1

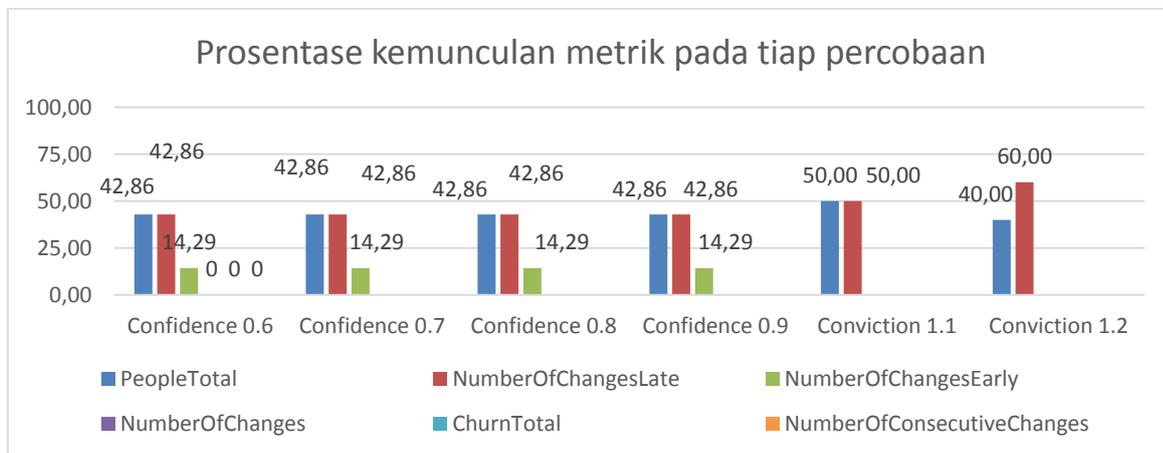
Minimum metric <confidence>: 0.8						Minimum metric <confidence>: 0.6					
Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conf. Level	Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conf. Level
PeopleTotal	2	3380	Bugs	yes	1	PeopleTotal	2	3380	Bugs	yes	1
PeopleTotal	3	2980	Bugs	yes	1	PeopleTotal	3	2980	Bugs	yes	1
NumberOfChangesLate	2	1770	Bugs	yes	1	NumberOfChangesLate	2	1770	Bugs	yes	1
NumberOfChangesLate	1	1730	Bugs	yes	1	NumberOfChangesLate	1	1730	Bugs	yes	1
NumberOfChangesLate	3	1330	Bugs	yes	1	NumberOfChangesLate	3	1330	Bugs	yes	1
PeopleTotal	4	1080	Bugs	yes	1	PeopleTotal	4	1080	Bugs	yes	1
NumberOfChangesEarly	4	1020	Bugs	yes	1	NumberOfChangesEarly	4	1020	Bugs	yes	1

Minimum metric <conviction>: 1.1						Minimum metric <conviction>: 1.2					
Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conviction Level	Metrik sebab	Nilai	Jumlah Data	Metrik Akibat	Nilai	Conviction Level
PeopleTotal	2	3380	Bugs	yes	3.46	PeopleTotal	2	3380	Bugs	yes	3.46
PeopleTotal	3	2980	Bugs	yes	3.05	PeopleTotal	3	2980	Bugs	yes	3.05
NumberOfChangesLate	2	1770	Bugs	yes	1.81	NumberOfChangesLate	2	1770	Bugs	yes	1.81
NumberOfChangesLate	1	1730	Bugs	yes	1.77	NumberOfChangesLate	1	1730	Bugs	yes	1.77
NumberOfChangesLate	3	1330	Bugs	yes	1.36	NumberOfChangesLate	3	1330	Bugs	yes	1.36
PeopleTotal	4	1080	Bugs	yes	1.11						

Gambar 9. Hasil percobaan.

Tabel 2. Jumlah kemunculan metrik-metrik pada tiap percobaan.

Metrik	Percobaan						Rata2
	1	2	3	4	5	6	
	Confidence 0.6	Confidence 0.7	Confidence 0.8	Confidence 0.9	Conviction 1.1	Conviction 1.2	
PeopleTotal	42,86	42,86	42,86	42,86	50,00	40,00	43,57
NumberOfChangesLate	42,86	42,86	42,86	42,86	50,00	60,00	46,90
NumberOfChangesEarly	14,29	14,29	14,29	14,29			14,29
NumberOfChanges	0						0,00
ChurnTotal	0						0,00
NumberOfConsecutiveChanges	0						0,00



Gambar 10. Grafik prosentase kemunculan metrik dalam tiap percobaan