

Pemanfaatan Algoritma WIT-Tree dan HITS untuk Klasifikasi Tingkat Keberhasilan Pemberdayaan Keluarga Miskin

Siti Khomsah^{*1}, Edi Winarko²

¹Prodi S2/S3, Jurusan Ilmu Komputer, FMIPA UGM, Yogyakarta

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta Sekip Utara Yogyakarta

e-mail: ^{*1}sitikhomsah5@gmail.com, ²ewinarko@ugm.ac.id

Abstrak

Tingkat keberhasilan pemberdayaan keluarga miskin dapat diklasifikasikan oleh pola karakteristik yang diekstrak dari database keluarga miskin. Tujuan penelitian ini adalah membangun model klasifikasi untuk memprediksi tingkat keberhasilan dari keluarga miskin, yang akan menerima bantuan pemberdayaan.

Model klasifikasi dibangun menggunakan metode *weighted association rule mining* (WARM), yang mengkombinasikan metode *Hyperlink Induced Topic Search* (HITS) dan *Weighted Itemset TidSet tree* (WIT-tree). Secara umum, bobot atribut dalam WARM ditentukan langsung oleh pengguna tanpa mengetahui bagaimana menentukan bobot yang tepat. HITS digunakan untuk memperoleh bobot atribut dari database. Bobot tersebut digunakan untuk menentukan bobot atribut pada metode WIT-tree. WIT-tree membentuk rule yang memenuhi ambang batas minimum *weight support* dan minimum *weight confidence*. Rule tersebut akan digunakan untuk mengklasifikasikan data baru.

Data yang digunakan dalam penelitian ini adalah 831 data sampel keluarga miskin, yang dibagi menjadi dua kelas, yaitu keluarga miskin dalam kategori "berkembang" dan "belum berkembang". Akurasi model diukur menggunakan *k-fold cross validation* dengan ambang batas minimum *weight support* adalah 0,1; 0,2; 0,3 dan minimum *weight confidence* 0,5. Pembobotan atribut menggunakan HITS, akurasi 86,45% dan pembobotan atribut yang ditetapkan oleh pengguna akurasi 66,13%. Penelitian ini menunjukkan bahwa bobot atribut yang diperoleh dari metode HITS lebih baik daripada bobot atribut yang ditentukan oleh pengguna.

Kata kunci—pemberdayaan-keluarga miskin, WARM, WIT-tree, HITS

Abstract

The successful rate of the poor families empowerment can be classified by characteristic patterns extracted from the database that contains the data of the poor families empowerment. The purpose of this research is to build a classification model to predict the level of success from poor families, who will receive assistance empowerment of poverty.

Classification models built with WARM, which is combining two methods, they are HITS and WIT-tree. HITS is used to obtained the weight of the attributes from the database. The weights are used as the attributes's weight on methods WIT-tree. WIT-tree is used to generate the association rules that satisfy a minimum *weight support* and minimum *weight confidence*. The data used was 831 sample data poor families that divided into two classes, namely poor families in the standard of "developing" and poor families in the level of "underdeveloped".

The performance of classification model shows, weighting attribute using HITS approaches the accuracy of 86.45% and weighted attributes defined by the user approaches the accuracy of 66.13%. This study shows that the weight of the attributes obtained from HITS is better than the weight of the attributes specified by the user.

Keywords—poverty reduction, Association Rule Classifier, Weighted Association Rule Classifier, WIT-tree, HITS

1. PENDAHULUAN

Database keluarga miskin Kabupaten Bantul menunjukkan bahwa sebagian besar keluarga miskin masih tetap tingkat kemiskinannya meskipun sudah diberikan bantuan pemberdayaan. Hal ini mengindikasikan program pemberdayaan kurang berhasil. Penelitian [1] menyebutkan, kegagalan program pemberdayaan disebabkan oleh program yang tidak tepat sasaran dan database keluarga miskin belum dimanfaatkan untuk intervensi pelaksanaan program pemberdayaan. Database keluarga miskin tersebut memuat data dasar penduduk keluarga miskin berikut 11 aspek indikator kemiskinan. Pada satuan kerja Pemdes Kabupaten Bantul terdapat database pemberdayaan keluarga miskin berisi evaluasi level keberhasilan penerima bantuan program pemberdayaan. Level evaluasi ada yaitu *berkembang* dan *belum berkembang*. Dari kedua database, pola hubungan antara 11 aspek indikator kemiskinan dan hasil evaluasi dapat digali. Pola yang diperoleh akan menjadi pengetahuan baru untuk memprediksi tingkat keberhasilan keluarga miskin lainnya yang akan menerima bantuan. Salah satu metodologi untuk menggali pola pengetahuan dari database yang besar adalah teknik *data mining* yaitu *association analysis* [2]. Teknik ini menggali pola tersembunyi antar *item* dalam sebuah *record* dan merepresentasikannya dalam bentuk *association rules* [3]. Beberapa penelitian menerapkan *association rules mining* (ARM) untuk membuat sistem klasifikasi. Hasilnya, metode baru yang disebut *Association Rule Mining Classifier* (ARMC) mampu meningkatkan akurasi klasifikasi [4][5][6][6][7][8].

Permasalahan ARMC tradisional adalah menganggap semua *item* mempunyai bobot kepentingan yang sama, hanya ditentukan ada atau tidak ada *item* tersebut dalam database [8]. Dalam beberapa domain permasalahan, menetapkan tingkat kepentingan secara merata untuk semua atribut tidak masuk akal, khususnya sistem prediksi dimana setiap atribut mempunyai perbedaan kapabilitas prediksi [9][10]. Misalnya, dalam domain prediksi tingkat keberhasilan keluarga miskin, atribut penghasilan dan pendidikan lebih berpengaruh daripada atribut kepemilikan rumah, sandang dan akses listrik. Salah satu metode untuk mengatasi masalah tersebut adalah *Weighted Association Rule Mining* (WARM) yaitu memberikan bobot atribut yang berbeda sesuai tingkat kepentingannya [9][10]. WARM bekerja lebih efisien dalam menghasilkan *frequent itemsets* dan *rule* jika dilakukan dengan metode pembobotan *item* [9][10][11][12]. Secara umum, bobot atribut pada WARM ditentukan oleh user atau pakar secara langsung. Tetapi tidak semua user atau pakar dapat memastikan bahwa bobot yang ditetapkan sudah tepat atau belum. Proses pembentukan aturan akan lebih akurat jika bobot atribut diekstrak dari database dengan metode *Hypertext-Induced Topic Search* (HITS) [13][14]. Hal ini disebabkan karena setiap atribut mempunyai derajat kepentingan yang berbeda sehingga harus diberikan bobot yang berbeda pula [14]. Atribut-atribut dengan bobot tinggi akan masuk dalam *frequent itemsets* dan berpengaruh pada akurasi *rule* dibandingkan jika tanpa bobot atribut [15][16].

WARM klasik menggunakan algoritma Apriori. Kelemahan Apriori adalah proses baca ulang database saat pembentukan setiap kandidat *frequent itemset* harus selalu dilakukan. Untuk mengatasi kelemahan tersebut digunakan struktur data *Weighted Itemset Tidset-tree* (WIT-tree) untuk membangkitkan *Frequent Weighted Itemset* (FWIs) [17][18]. WIT-tree membangun pohon *frequent itemsets* dengan melakukan interseksi antar himpunan *id record* (*Tidset*). Interseksi dilakukan pada saat membentuk kandidat *itemset*.

Berdasarkan uraian diatas penelitian ini akan membuat model prediksi kelas tingkat keberhasilan keluarga miskin calon penerima bantuan menggunakan gabungan metode asosiasi dan klasifikasi. Algoritma HITS digunakan untuk memperoleh bobot atribut 11 aspek kemiskinan berdasarkan kondisi data terkini. Algoritma WIT-tree membentuk *frequent itemsets* dengan mekanisme *pruning* menggunakan *minimum weight support* (*minws*) dan *minimum weight confidence* (*mincof*). *Frequent itemsets* yang memenuhi ambang batas *minws* dan *mincof* akan menjadi *rule*, selanjutnya digunakan untuk mengklasifikasi keluarga miskin calon penerima bantuan, apakah masuk dalam level *berkembang* atau *belum berkembang*.

2. METODE PENELITIAN

2.1. Analisa Masalah

Indikator untuk menentukan tingkat kemiskinan di Kabupaten Bantul ada 11 atribut. Masing-masing atribut mempunyai bobot yang sudah ditentukan oleh BKKPPKB seperti pada Tabel 1.

Tabel 1 Bobot atribut aspek penyebab kemiskinan Kabupaten Bantul

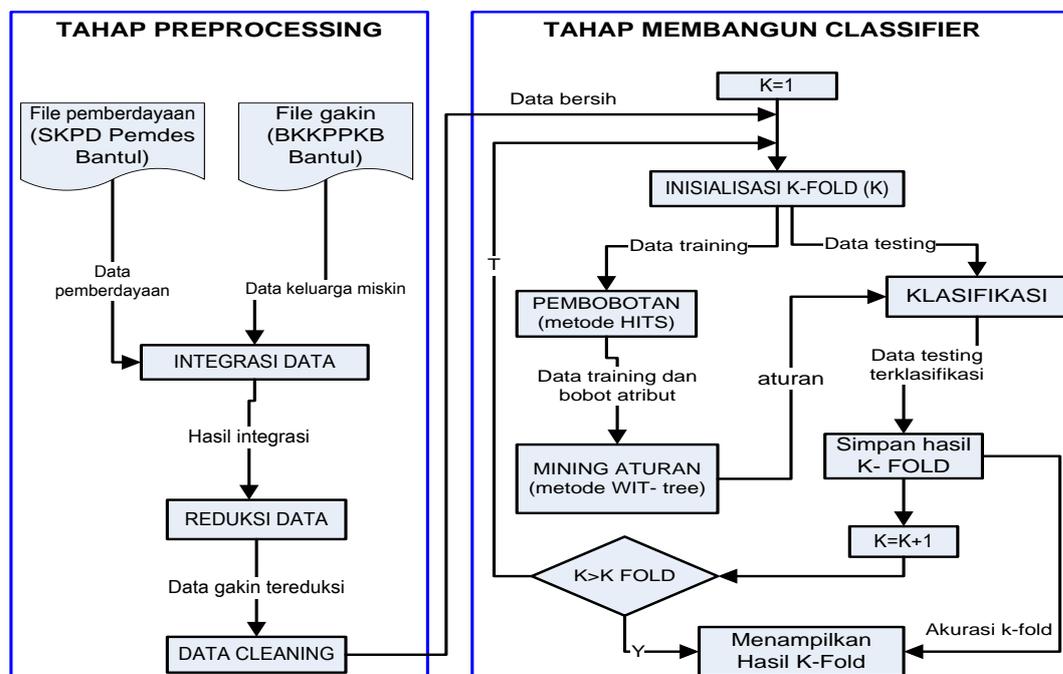
Kode	Atribut	Bobot	Kode	Atribut	Bobot
A	AsPangan	12	G	AsPerumahan 1	5
B	AsSandang	9	H	AsPerumahan 2	6
C	AsPapan	9	I	As Air Bersih	4
D	AsPenghasilan	35	J	AsListrik	4
E	AsKesehatan	6	K	AsJumlahJiwa	5
F	AsPendidikan	6			

(Sumber: BKKPPKB Kabupaten Bantul)

Permasalahannya, bobot atribut oleh BKKPPKB tidak dapat dikonfirmasi alasan dan dasarnya secara rinci sehingga bisa jadi bobot tersebut tidak tepat digunakan. Oleh karena itu, pembobotan atribut secara dinamis berdasarkan kondisi *database*, bisa jadi lebih tepat. Beberapa penelitian menerapkan pembobotan dinamis menggunakan metode HITS. Hasilnya, bobot atribut yang diperoleh secara dinamis berdasarkan kondisi database akan lebih akurat [13][14][15][16].

2.2. Arsitektur Sistem

Arsitektur sistem model yang akan dibangun ditunjukkan oleh Gambar 1. Tahap pertama yaitu *preprocessing* terdiri dari integrasi, reduksi, *cleaning* dan transformasi. Tahap kedua membangun model *classifier* terdiri dari inialisasi jumlah k untuk metode k -fold (1), proses memperoleh bobot *item* berdasarkan *data training* dengan algoritma HITS (2), proses *mining* aturan dengan algoritma WIT-tree (3) dan klasifikasi (4). Metode k -fold digunakan untuk mengukur akurasi model yang dihasilkan.



Gambar 1 Gambaran umum model klasifikasi tingkat keberhasilan pemberdayaan keluarga miskin

2.3. Preprocessing

Preprocessing meliputi proses integrasi data, reduksi data, data cleaning dan transformasi data

1. Integrasi data

Integrasi dilakukan untuk menggabungkan file data keluarga miskin dari BKKPPKB dan file data evaluasi pemberdayaan dari SKPD Pemdes ke dalam satu basisdata. Data keluarga miskin sejumlah 1086 *record*. Data evaluasi pemberdayaan sejumlah 831, yang terbagi dalam dua level yaitu “Berkembang” sebanyak 576 dan “Belum Berkembang” sebanyak 255.

2. Reduksi data

Reduksi data dilakukan dengan memilih data penduduk miskin dengan kriteria sebagai berikut:

- Data keluarga miskin dengan status sebagai KK (kepala keluarga).
- KK penerima bantuan tahun 2011
- KK dengan evaluasi Berkembang atau Belum Berkembang.
- Tidak semua atribut digunakan dalam penelitian ini sehingga hanya atribut NIK, status KK (dalam data asli adalah kolom *stkel*), 11 aspek indikator kemiskinan dan evaluasi yang dipilih dari database keluarga miskin dan pemberdayaan. Sebelas aspek indikator kemiskinan tersebut adalah, aspek pangan (1), aspek sandang (2), aspek papan (3), aspek penghasilan (4), aspek kesehatan (5), aspek pendidikan (6), aspek kekayaan harta (7), aspek kepemilikan rumah (8), aspek air bersih (9), aspek listrik (10), dan aspek jumlah jiwa/ tanggungan (11)

Data hasil integrasi dan reduksi sebanyak 831 *record*.

3. Data cleaning

Proses *data cleaning* bertujuan mengambil data yang tidak mengandung *noise*, data tidak lengkap dan format data yang salah. Beberapa aturan validasi yang dilakukan dalam proses *data cleaning* adalah sebagai berikut:

- Validasi nomor induk kependudukan (NIK). Nomor NIK harus dalam format string dengan panjang 16 karakter tanpa titik dan spasi.
- Jika NIK kosong atau tidak memenuhi 16 karakter maka data keluarga miskin tidak akan diikuti dalam *data training*.
- *Cleaning* data 11 aspek tersebut dikenakan aturan yaitu jika atribut tidak berisi data akan diisi dengan nilai 0. Jika atribut berisi nilai bukan 0 atau angka lain tetapi nilai skor tidak sesuai pedoman pada Tabel 1 maka akan diisi dengan nilai skor yang ditetapkan sesuai Tabel 1. Misalnya aspek penghasilan tidak berisi data maka akan diisi angka 0 sebaliknya jika berisi data namun bukan 35 akan diisi dengan 35 karena skor aspek penghasilan seharusnya 35. Alasan aturan ini dikenakan karena bisa jadi terjadi kesalahan petugas input data akibat *tool* yang digunakan tidak mampu mengatasi kesalahan tersebut. Kesalahan yang umum ditemukan yaitu antara aspek pendidikan dan aspek kesehatan.

4. Transformasi data

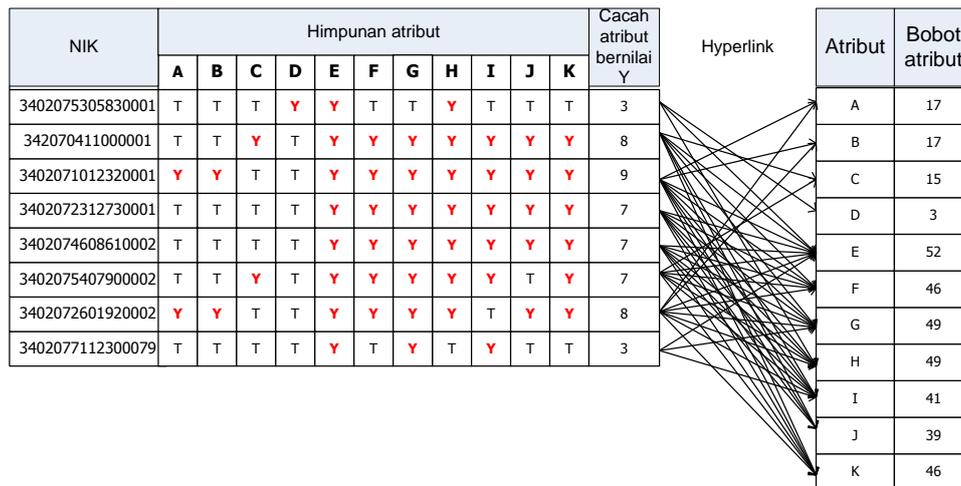
Transformasi dilakukan pada data 11 (sebelas) indikator kemiskinan. Tujuan transformasi adalah mengubah nilai atribut kedalam bentuk biner yaitu Y (Ya) dan T (tidak).

2.4. Pembobotan Item dengan HITS

Metode HITS merepresentasikan hubungan antara *item* dan himpunan atribut sebagai sebuah *bipartite graph* [13][14][15][16]. Misalnya, terdapat database pemberdayaan keluarga miskin pada Tabel 2 dengan 11 atribut aspek kemiskinan. Representasi *bipartite graph* -nya ditunjukkan oleh Gambar 2.

Tabel 2 Contoh *database* keluarga miskin

NIK	Atribut indikator											Pemberdayaan
	A	B	C	D	E	F	G	H	I	J	K	
3402075305830001	T	T	T	Y	Y	T	T	Y	T	T	T	Berkembang
3402070411000001	T	T	Y	T	Y	Y	Y	Y	Y	Y	Y	Berkembang
3402071012320001	Y	Y	T	T	Y	Y	Y	Y	Y	Y	Y	Berkembang
3402072312730001	T	T	T	T	Y	Y	Y	Y	Y	Y	Y	Berkembang
3402074608610002	T	T	T	T	Y	Y	Y	Y	Y	Y	Y	Berkembang
3402075407900002	T	T	Y	T	Y	Y	Y	Y	Y	T	Y	Berkembang
3402072601920002	Y	Y	T	T	Y	Y	Y	Y	T	Y	Y	Berkembang
3402077112300079	T	T	T	T	Y	T	Y	T	Y	T	T	Berkembang
3402077112410026	T	T	T	T	T	T	T	T	Y	T	T	Belum berkembang
3402075305830001	T	T	T	T	T	T	Y	T	T	T	Y	Belum berkembang



Gambar 2 Perhitungan bobot atribut

Gambar 2 menunjukkan cacah atribut diisi dengan jumlah *hyperlink* yang terjadi antara setiap *record* dan atribut. Bobot atribut merupakan akumulasi total cacah *hyperlink* yang dimiliki oleh setiap *record* dalam *database*. Contohnya pada Gambar 2, bobot atribut A adalah 17 diperoleh dengan menjumlahkan cacah *hyperlink* NIK 3402071012320001 yaitu 9 dan cacah *hyperlink* NIK 3402077112410026 yaitu 8, karena dalam NIK tersebut terdapat atribut A. Atribut yang mempunyai banyak *hyperlink* dengan *record* dalam *database* akan mempunyai bobot tinggi dan atribut yang bobotnya tinggi terdapat pada banyak *record* data yang penting [13][14]. Bobot atribut tersebut merefleksikan tingkat kepentingannya. Untuk itu diperlukan normalisasi bobot atribut, caranya membagi setiap nilai bobot atribut dengan nilai bobot atribut tertinggi. Sehingga diperoleh nilai bobot atribut antara 0 sampai dengan 1. Hasil normalisasi bobot atribut pada Gambar 2 ditunjukkan pada Tabel 3.

Tabel 3 Contoh bobot atribut hasil metode HITS

Kode	Atribut	Bobot HITS
A	AsPangan	0,33
B	AsSandang	0,33
C	AsPapan	0,29
D	AsPenghasilan	0,06
E	AsKesehatan	1,00
F	AsPendidikan	0,88
G	AsPerumahan 1	0,94
H	AsPerumahan 2	0,94
I	As Air Bersih	0,79
J	AsListrik	0,75
K	AsJumlahJiwa	0,88

2.5. Struktur Data WIT-Tree (Weighted Itemsets Tidset-Tree)

Proses *Weighted Association Rule Mining* menggunakan algoritma WIT-tree untuk menghasilkan *frequent itemset* selanjutnya membentuk aturan klasifikasi. Pembentukan *frequent itemset* dilakukan terpisah pada *data training* dengan kelas “Berkembang” dan “Belum Berkembang”. Algoritma WIT-tree akan membaca database transaksi sekali saja, karena konsep dasarnya adalah menemukan titik potong antar *TidSets* (himpunan id *record*) untuk menghitung bobot *support* (ws) pada langkah berikutnya [18]. Hal ini menghemat waktu pembacaan database transaksi yang besar. Sebelum membangun WIT-tree, terlebih dahulu dicari nilai bobot *item* atau atribut dalam transaksi. Sebagai contoh, D adalah database transaksi terdiri atas beberapa transaksi $D = \{t_1, t_2, t_3, \dots, t_k\}$. Setiap transaksi atau *itemset* terdiri dari *item-item* atau atribut, dituliskan $I = \{i_1, i_2, i_3, \dots, i_j\}$ dan himpunan bobot $W = \{w_1, w_2, w_3, \dots, w_j\}$ berkorespondensi dengan setiap *item* dalam I . Bobot *record* atau bobot transaksi (tw) setiap transaksi ke- k (t_k) dihitung dengan persamaan (1) dimana $|t_k|$ adalah cacah *item* transaksi ke- k dan w adalah bobot setiap *item*.

$$tw(t_k) = \frac{\sum_{j=1}^{|t_k|} w(i_j)}{|t_k|} \quad (1)$$

Klasifikasi berbasis aturan asosiasi maka aturan asosiasi bukan bentuk $x \rightarrow y$ dimana y adalah *subset* dari aturan tersebut tetapi y adalah label kelas. Bobot *support* (ws) aturan $x \rightarrow \text{kelas label}$, dimana x adalah himpunan *item* [9]. Bobot *support* dihitung dengan persamaan (2).

$$ws(x \rightarrow \text{label kelas}) = \frac{\sum_{k=1}^{|x|} tw(t_k)}{\sum_{k=1}^n tw(t_k)} \quad (2)$$

dimana n adalah total *record* dalam database.

Ada dua hal yang diperhatikan dalam membangun WIT-tree yaitu *vertex* dan *arc* [13].

- **Vertex**

Vertex adalah node–node dalam membangun WIT-tree. Vertex terdiri dari tiga hal yaitu: X adalah *itemset* atau himpunan atribut dalam *record*, $t(X)$ adalah transaksi yang mengandung *itemset* X , dan $ws(X)$ adalah bobot *support* daripada *itemset* X . Vertex disimbolkan sebagai

$$\text{berikut: } \begin{matrix} X \times t(X) \\ ws(X) \end{matrix}$$

- **Arc**

Arc adalah garis hubungan antara vertex pada level ke k^{th} (disebut X) dengan vertex pada level $(k+1)^{\text{th}}$ (disebut Y). Prefiks node level satu adalah himpunan $\{\}$. Setiap *item* node level satu akan menjadi prefiks pada level berikutnya. Setiap node *item* pada level satu join dengan *item* sebelah kanannya sehingga membentuk level dibawahnya. Node dengan prefiks sama pada level sebelumnya disebut kelas ekuivalen. Misalnya, node $\{A\}$ secara berturut-turut akan join dengan node $\{B\}, \{C\}, \{E\}$ untuk membentuk kelas ekuivalen baru sehingga $[A] = \{\{AB\}, \{AC\}, \{AD\}, \{AE\}\}$. Node $[AB]$ akan menjadi kelas ekuivalen baru setelah di join-kan secara berturut-turut dengan node $\{AC\}, \{AD\}, \{AE\}$.

Langkah-langkah memperoleh *frequent weighted itemset* yaitu:

- Menghitung bobot *record*

Contoh database berikut nilai bobot *record*-nya (tw) ada pada Tabel 4. Kolom NIK adalah atribut nomor kependudukan. Kolom A sampai K adalah atribut 11 aspek kemiskinan, jumlah *item* adalah cacah *item* yang muncul dalam *record*. Cacah *item* yang muncul ditandai dengan huruf Y dan tidak muncul dengan huruf T.

Misalnya, NIK 3402075305830001 mempunyai atribut D, E, H, dengan merujuk nilai bobot atribut pada Tabel 3. Dengan persamaan (2) cara menghitung nilai $tw(3402075305830001)$ adalah:

$$tw(3402075305830001) = \frac{0,06+1,00+0,94}{3} = 0,67$$

Tabel 4 Contoh database dengan bobot *record*

NIK	A	B	C	D	E	F	G	H	I	J	K	Jumlah item	Total bobot item	Bobot record (Tw)
3402075305830001	T	T	T	Y	Y	T	T	Y	T	T	T	3	2	0,67
3402070411000001	T	T	Y	T	Y	Y	Y	Y	Y	Y	Y	7	6,47	0,81
3402071012320001	Y	Y	T	T	Y	Y	Y	Y	Y	Y	Y	9	6,83	0,76
3402072312730001	T	T	T	T	Y	Y	Y	Y	Y	Y	Y	7	6,18	0,88
3402074608610002	T	T	T	T	Y	Y	Y	Y	Y	Y	Y	7	5,72	0,82
3402075407900002	T	T	Y	T	Y	Y	Y	Y	Y	T	Y	7	6,04	0,76
3402072601920002	Y	Y	T	T	Y	Y	Y	Y	T	Y	Y	9	2,73	0,91
Total tw :														6,49

- Menghitung bobot *support* (ws)

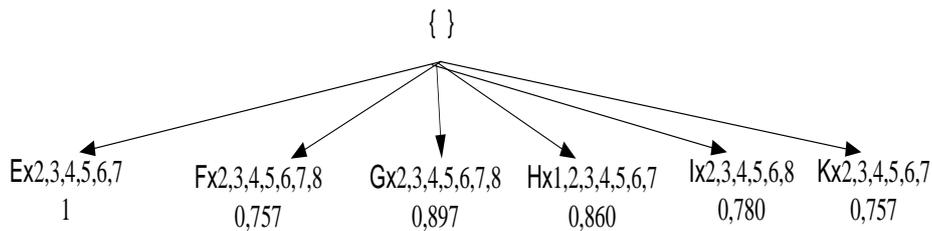
Klasifikasi berbasis *association rules mining* mempunyai aturan asosiasi $x \rightarrow y$, dimana x adalah *itemsets* dan y adalah label kelas [6]. Bobot *support* (ws) setiap kandidat *itemset* dihitung dengan persamaan (2). Bobot *support* digunakan sebagai cara *pruning* atau pemangkasan kandidat *itemset* yang tidak *frequent* [6]. Sedangkan *itemset* yang tidak memenuhi batas *minimum weight support* ($minws$) dan tidak akan diproses pada kandidat *itemset* berikutnya.

Misalnya, ditentukan $minws$ 0,7 (70%) sebagai batasan untuk menghitung bobot *support* terhadap database pada Tabel 4. *Itemset* A terdapat pada ID transaksi 3 dan 7. Berdasarkan Tabel 4, bobot *record* ID transaksi 3 adalah 0,76 dan ID transaksi 7 adalah 0,91. Dengan persamaan (2) bobot *support itemset* A adalah 0,234. Semua kandidat *itemset-1* ditunjukkan oleh Tabel 5. Struktur WIT- tree yang dihasilkan ditunjukkan oleh Gambar 3.

Tabel 5 *Itemset* berikut *weight support k-1*

No	Itemset	ID Transaksi Yang disupport Itemset	Total Transaction Weight (tw)	Weight Support (tw/Sum)
1	A	3,7	1,52	0,234
2	B	3,7	1,52	0,234
3	C	2,6	1,63	0,251
4	D	1	0,06	0,009
5	E	1,2,3,4,5,6,7,8	6,49	1
6	F	2,3,4,5,6,7	4,91	0,757
7	G	2,3,4,5,6,7,8	5,82	0,897
8	H	1,2,3,4,5,6,7	5,58	0,860
9	I	2,3,4,5,6,8	5,06	0,780
10	J	2,3,4,5,7	4,09	0,631
11	K	2,3,4,5,6,7	4,91	0,757

Tabel 5 menunjukkan bahwa *itemset* A, B, C, D dan J memiliki nilai ws dibawah *minimum weight support*, maka tidak dilibatkan pada perhitungan kandidat 2 *itemset* selanjutnya.



Gambar 3 WIT- tree frequent itemset pada kandidat-1

- WIT-tree untuk kandidat 2 itemset dan seterusnya

Kandidat 2 itemset ditunjukkan oleh Tabel 6.

Tabel 6 Itemset berikut weight support untuk k-2 itemset

No	Itemset	ID Transaksi Yang disupport Itemset	Total Transaction Weight (tw)	Weight Support (tw/Sum)
1	E-F	2,3,4,5,6,7	4,91	0,75693731
2	E-G	2,3,4,5,6,7,8	5,82	0,89722508
3	E-H	1,2,3,4,5,6,7	5,58	0,85971223
4	E-I	2,3,4,5,6,8	5,06	0,78006166
5	E-K	2,3,4,5,6,7	4,91	0,75693731
6	F-G	2,3,4,5,6,7	4,91	0,75693731
7	F-H	2,3,4,5,6,7	4,91	0,75693731
8	F-I	2,3,4,5,6	4,15	0,6397739
9	F-K	2,3,4,5,6,7	4,91	0,75693731
10	G-H	2,3,4,5,6,7,8	5,82	0,89722508
11	G-I	2,3,4,5,6,8	5,06	0,78006166
12	G-K	2,3,4,5,6,7	4,91	0,75693731
13	H-I	1,2,3,4,5,6,7	5,58	0,85971223
14	H-K	2,3,4,5,6,7	4,91	0,75693731
15	I-K	2,3,4,5,6	4,15	0,6397739

Perhitungan k-2 itemset menunjukkan bahwa itemset yang pruning adalah F-I dan I-K, sehingga semua itemset selain itu terlibat pada perhitungan k-3 itemset dan seterusnya. Dengan algoritma berikut yang diadaptasi dari [18], akan dihasilkan kandidat 4-itemset seperti oleh Tabel 7 dan kandidat 5-itemset pada Tabel 8.

Input : database D dan minws
 Output : FWIs yang berisi semua frequent weighted itemsets yang memenuhi minws dari D

Method :

- (1) WIT-FWIs()
- (2) $[\phi] = \{I \in I : ws(i) \geq minws\}$
- (3) SORT($[\phi]$);
- (4) FWIs= ϕ
- (5) FWIs_EXTEND($[\phi]$)
- (6) FWIs_EXTEND([P])
- (7) For all $L_i \in [P]$ do
- (8) Add ($L_i, ws(L_i)$) to FWIs
- (9) $[P_i] = \phi$
- (10) For all $L_j \in [P]$, with $j > i$ do
- (11) $X = L_i \cup L_j$
- (12) $Y = t(L_i) \cap t(L_j)$
- (13) If $|t(L_i)| = |Y|$ then $ws(X) = ws(L_i)$
- (14) Else If $|t(L_j)| = |Y|$ then $ws(X) = ws(L_j)$
- (15) Else $ws(X) = COMPUTE-WS(Y)$
- (16) if $ws(X) \geq minws$ then
- (17) $[P_i] = [P_i] \cup \{ X \times Y \}$
- (18) $ws(X)$
- (19) FWIs_EXTEND($[P_i]$)

Tabel 7 *Itemset* berikut *weight support* untuk *k-2 itemset*

No	<i>Itemset</i>	ID Transaksi Yang di <i>support Itemset</i>	Total <i>Transaction Weight (tw)</i>	<i>Weight Support</i> (<i>tw/Sum</i>)
1	E-F-G-H	2,3,4,5,6,7	4,91	0,757
2	E-F-G-K	2,3,4,5,6,7	4,91	0,757
3	E-F-H-K	2,3,4,5,6,7	4,91	0,757
4	E-F-I-K	2,3,4,5,6,7	4,91	0,757
5	E-G-H-K	2,3,4,5,6,7	4,91	0,757
6	F-G-H-K	2,3,4,5,6,7	4,91	0,757
7	F-G-I-K	2,3,4,5,6	4,15	0,64
8	F-H-I-K	2,3,4,5,6	4,15	0,64

Tabel 8 *Itemset* berikut *weight support* untuk *k-5 itemset*

No	<i>Itemset</i>	ID Transaksi Yang di <i>support Itemset</i>	Total <i>Transaction Weight (tw)</i>	<i>Weight Support</i> (<i>tw/Sum</i>)
1	E-F-G-H-I	2,3,4,5,6	4,15	0,64
2	E-F-G-H-K	2,3,4,5,7	4,09	0,63
5	E-G-H-I-K	2,3,4,5,6	4,15	0,64

2.6. Bobot Confidence

Weighted Association Rule Mining (WARM) adalah bentuk implikasi $X \rightarrow Y$, dimana $X \subseteq I$, I adalah *itemset* dan Y adalah kelas label. *Rule* akan dibentuk dari *frequent itemset* yang terbentuk dengan WIT-tree. Bobot *confidence* setiap $X \rightarrow Y$ akan dihitung. Nilai bobot *confidence* merupakan perbandingan antara bobot *support* $X \cup Y$ dengan bobot *itemset* X seperti persamaan (3)

$$\text{Bobot Confidence } (X \rightarrow Y) = \frac{\text{Bobot Support } (X \cup Y)}{\text{Bobot Support } (X)} \quad (3)$$

Misalnya, batas minimal bobot *confidence* yang digunakan 70%, sebagian *rule* yang dihasilkan WIT- tree untuk database pada Tabel 1, ada pada Tabel 9.

Tabel 9 *Rule* yang terbentuk

No	Anteseden (IF)	Konsekuensi (THEN)	Bobot Support	Bobot Confidence
1	E AND F AND I AND K	Berkembang	0,7569373	100%
2	E AND G AND H AND K	Berkembang	0,7569373	100%
3	F AND G AND H AND K	Berkembang	0,7569373	100%
4	E AND F AND I AND K	Berkembang	0,7569373	100%

2.7. Klasifikasi

Data baru akan diklasifikasi kedalam kelas *Berkembang* dan *Belum berkembang*. Langkah-langkah klasifikasi adalah :

1. *Rule* diurutkan berdasarkan nilai *weight confidence*, *weight support*, dan cacah anteseden.
2. Kemudian mencari prosentase kecocokan antara *itemset* data testing dengan setiap *antecedent* aturan yang dibangkitkan proses *training*.
3. Memilih aturan dengan prosentase kecocokan yang paling besar dari semua aturan yang telah dihitung tingkat kecocokannya.

Berikut contoh *data testing* yang akan diprediksi kelasnya:

NIK	A	B	C	D	E	F	G	H	I	J	K
3402121002250001	Y	Y	T	T	Y	Y	Y	Y	Y	Y	Y

Data tersebut hendak dicocokkan dengan aturan dengan anteseden dan konsekuen: **F-G-H-K → Berkembang**.

Maka langkah pencocokannya adalah :

- 1) Tentukan *Similarity*=0
- 2) Apakah F bernilai Y, jika ya maka *similarity*=1
- 3) Apakah G bernilai Y, jika ya maka *similarity*=1+1 =2
- 4) Apakah H bernilai Y, jika ya maka *similarity*=2+1 =3
- 5) Apakah K bernilai Y, jika ya maka *similarity*=3+1 =4
- 6) Total jumlah anteseden adalah 4 maka *similarity data testing* tersebut dengan aturan *F-G-H-K → Berkembang* adalah $4/4 * 100\% = 100\%$

Maka data testing tersebut diprediksi akan *Berkembang*.

3. HASIL DAN PEMBAHASAN

Pembentukan *frequent itemset* dilakukan dengan metode WIT-tree. Langkah pertama yaitu menghitung *transaction weight (tw)*. Pada pembobotan dinamis kolom atribut dari A hingga K yang pada proses sebelumnya berisi data Y akan digantikan dengan bobot HITS. Sedangkan jika berisi data T akan diganti dengan angka 0. Pada pembobotan statis kolom atribut dari A hingga K yang pada proses sebelumnya berisi data Y akan digantikan dengan bobot statis.

Nilai *transaction weight (tw)* berperan penting dalam pembentukan WIT-tree. Suatu *itemset* yang nilai *tw* tidak mencapai *minimum weight support (minws)* tidak akan dipakai dalam proses pembentukan WIT-tree.

Pengujian akurasi model dilakukan metode *k-fold* dengan jumlah $k=4$. Jumlah *dataset* 831 *record* akan diambil dibagi secara merata secara acak dalam empat partisi dengan proporsi prosentase sama. Sehingga data uji adalah 25% dari *dataset* dan *data training* 75% dari *dataset*. Setiap perulangan *k*, satu partisi menjadi data testing dan 3 partisi lainnya menjadi *data training* secara bersamaan. Model diuji dengan *minimum weight suport* 10%, 20% dan 30% dan *minimum confidence* 50%. Hasil pengujian ada pada Tabel 10. Hasil pengujian ditunjukkan oleh Tabel 10 untuk akurasi dan Tabel 11 untuk jumlah rule yang dihasilkan.

Tabel 10 Pengujian akurasi klasifikasi

Perulangan fold	Akurasi dengan metode bobot (%)	
	HITS	BKK
1	89,00	71,95
2	88,60	66,30
3	80,00	67,30
4	88,20	59,00
Rata-rata akurasi (%)	86,45 %	66,1375%

Tabel 11 Jumlah rule yang dihasilkan

Perulangan fold	Jumlah rule	
	HITS	BKK
1	220	227
2	220	227
3	220	231
4	215	221

4. KESIMPULAN

Berdasarkan pengujian yang telah dibahas pada disimpulkan bahwa:

1. Semakin tinggi bobot *support* berpengaruh pada akurasi metode gabungan WIT-tree dan pembobotan HITS semakin tinggi. Sedangkan pada metode gabungan WIT-tree dan pembobotan BKKPPKB, semakin tinggi bobot *support* tidak berpengaruh pada akurasi yang dihasilkan.

2. Metode pembobotan dinamis dengan HITS memberikan akurasi model klasifikasi lebih baik dibanding pembobotan statis. Rata-rata akurasi model dengan metode gabungan WIT-tree dan pembobotan HITS sebesar 86,45% sedangkan dengan metode gabungan WIT-tree dan pembobotan BKKPPKB sebesar 66,13%. Angka ini diperoleh untuk *minimum bobot support* 0,1; 0,2; 0,3 dan *minimum weight confidence* minimal 0,5.
3. Jumlah aturan yang dihasilkan metode HITS lebih sedikit namun mempunyai akurasi lebih baik.

5. SARAN

Beberapa hal yang dapat dilakukan untuk perbaikan penelitian ini antara lain:

1. Penelitian selanjutnya melibatkan atribut lainnya, seperti atribut keahlian yang dimiliki, usia, pekerjaan, dan kondisi geografis.
2. Penelitian selanjutnya dapat menggunakan metode pembobotan atribut lainnya seperti *Analytic Hierarchy Process (AHP)* dan *Genetic Algorithm (GA)*.
3. Penelitian selanjutnya dapat menerapkan metode *Multiple Correspondence Analysis (MCA)* sebagai mekanisme membangkitkan association rule karena MCA dapat mengenali korespondensi antara *item-item* dan kelasnya.

DAFTAR PUSTAKA

- [1] Syari'udin, A, [Artiani, L.E.](#), Gusaptono, H., 2011, Efektivitas Program Pengentasan Kemiskinan :Studi Kasus Kabupaten Bantul, Provinsi Daerah Istimewa Yogyakarta, *Laporan Penelitian*, LPPM [Universitas Pembangunan Nasional Veteran Yogyakarta](#), Yogyakarta.
- [2] Tan, P., Steinbach, M., dan Kumar, V., 2006, *Data Mining Concept and Technique*, Morgan Kaufman Publisher, San Francisco.
- [3] Agrawal, R. dan Srikant, R., 1994, Fast Algorithms for Mining Association Rules, *VLDB'94*, pp. 487- 499.
- [4] Liu, B., Hsu, W., & Ma, Y. (1998). Integrating Classification And Association Rule Mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining (KDD-98)*. New York.
- [5] Liu, B., Ma, Y., dan Wong, C.-K., 2001, Classification Using Association Rules:Weaknesses and Enhancements, *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishers, New York.
- [6] Yin, X. dan Han,J., 2003, CPAR: Classification Based On Predictive Association Rules, *Proceedings of the Society for Industrial and Applied Mathematics International Conference on Data Mining*, San Francisco, Calif, USA
- [7] Yang, Z., Tang, W.H., Shintemirov, A., Wu., Q.H., 2009, Association Rule Mining Based Dissolved Gas Analysis for Fault Diagnosis of Power Transformer, *IEEE*, Vol 39
- [8] Dua,S., Singh,H., Thompson, H.W. , 2009, Associative Classification of Mammograms using Weighted Rules, *Expert Systems with Applications*, Volume 36, Issue 5, Pages 9250–9259.
- [9] Soni, S, Pillai, J, Vyas, O.P, 2009, An Associative Classifier Using Weighted Association Rule, *IEEE*, 978-1-4244-5612-3/09.
- [10] Tao, F., Murtagh, F. dan Farid,M.,Weighted Association Rule Mining using Weighted Support and Significance Framework, 2003, *SIGKDD*, August 24-27, Washington, DC, USA.
- [11] Kumar, P and V.S., Ananthanarayana, 2010, Discovery of Weighted Association Rule Mining, *IEEE* , volume 5, 978-1-4244-5586-7.

-
- [12] Mary, S.A. dan Malarvizhi,M., 2014, A New Improved Weighted Association Rules Mining With Dynamic Programming Approach For Predicting A User's Next Access, *Computer Science & Information Technology*.
- [13] Wang, K. dan Su, T., 2002, Item Selection By "HubAuthority" Profit Ranking, *SIGKDD '02*, Canada, ACM 158113567X/02/0007.
- [14] Sun, K dan Bai,F., 2008, Mining Weighted Association Rules without Preassigned Weights, *IEEE Transactions On Knowledge And Data Engineering*, Vol. 20, No. 4, April 2008.
- [15] Ibrahim, S.P.S, dan Chandran, K.R., 2011, Compact Weight Class Association Rule Mining Using Information Gain, *International Journal of Data Mining & Knowledge Management Process*, Vol.1, No.6 November 2011.
- [16] Padmavalli, M., Dan Rao, Sreenivasa, 2013, An Efficient Interesting Weighted Association Rule Mining , *International Journal Of Advanced Research In Computer Science And Software Engineering*, Volume 3, Issue 10, October 2013 ISSN: 2277 128X.
- [17] Le, B., Nguyen, T.A. Cao, B. Vo, 2009, A Novel Algorithm for Mining High Utility Itemsets, *IEEE*, pp. 13 – 16.
- [18] Le, B., Nguyen, H., Vo, B., 2010, Efficient Algorithms for Mining Frequent Weighted Itemsets from Weighted Items Databases, *IEEE*, 978-1-4244-8075-3/10.