SELEKSI FITUR MENGGUNAKAN *FAST CORRELATION BASED FILTER* PADA ALGORITMA *VOTING FEATURE INTERVALS* 5

Hida Nur Firqiani, Aziz Kustiyo , Endang Purnama Giri

ABSTRAK

Seleksi fitur adalah salah satu tahapan praproses klasifikasi. Seleksi fitur dilakukan dengan cara memilih fitur-fitur yang relevan yang mempengaruhi hasil klasifikasi. Seleksi fitur digunakan untuk mengurangi dimensi data dan fitur-fitur yang tidak relevan. Seleksi fitur digunakan untuk meningkatkan efektifitas dan efisiensi kinerja dari algoritma klasifikasi. Pada penelitian ini dilakukan seleksi fitur menggunakan Fast Correlation Based Filter pada klasifikasi data menggunakan algoritma Voting Feature Intervals 5. Penelitian ini bertujuan menganalisis kinerja seleksi fitur pada klasifikasi data menggunakan algoritma Voting Feature Intervals 5. Penelitian ini akan membandingkan tingkat akurasi data pada algoritma klasifikasi Voting Feature Intervals 5 jika sebelumnya dilakukan seleksi fitur dan tanpa dilakukan seleksi fitur. Data yang digunakan pada penelitian ini memiliki dimensi yang beragam. Hasil dari penelitian ini berupa perbandingan nilai akurasi data pada klasifikasi menggunakan Voting Feature Intervals 5 jika sebelumnya dilakukan seleksi fitur dan tidak dilakukan seleksi fitur. Hasil yang diperoleh menunjukkan bahwa nilai akurasi klasifikasi dengan seleksi fitur lebih baik daripada tanpa seleksi fitur. Dari keempat data yang digunakan, tingkat akurasi data mengalami peningkatan jika menggunakan seleksi fitur. Rata-rata hasil akurasi data tanpa seleksi fitur yaitu 81.66% sedangkan menggunakan seleksi fitur yaitu 85.51% .

Kata Kunci: *seleksi fitur, voting feature intervals, fast correlation based filter, klasifikasi.*

PENDAHULUAN

Latar Belakang

Klasifikasi adalah proses menemukan sekumpulan model yang menggambarkan serta membedakan kelas-kelas data. Tujuan dari klasifikasi adalah agar model yang dihasilkan dapat digunakan untuk memprediksi kelas dari suatu data yang tidak mempunyai label kelas. Jika diberikan sekumpulan data yang terdiri dari beberapa fitur dan kelas, maka klasifikasi adalah menemukan model dari kelas tersebut sebagai fungsi dari fitur-fitur yang lain.

Pada umumnya algoritma klasifikasi menggunakan semua fitur yang terdapat pada data untuk membangun sebuah model, padahal tidak semua fitur tersebut relevan terhadap hasil klasifikasi. Apabila hal tersebut terjadi pada data yang memiliki ukuran dan dimensi yang sangat besar, maka membuat kinerja algoritma menjadi tidak efektif dan efisien, misalnya saja waktu pemrosesan menjadi lebih lama akibat banyak fitur yang harus diproses.

Salah satu solusi yang digunakan untuk mengatasi masalah tersebut adalah dengan menggunakan seleksi fitur. Seleksi fitur adalah salah satu tahap praproses pada klasifikasi. Seleksi fitur dilakukan dengan cara memilih fitur-fitur yang relevan terhadap data yang mempengaruhi hasil klasifikasi. Seleksi fitur digunakan untuk mengurangi dimensi data dan fitur yang tidak relevan, serta untuk meningkatkan efektifitas dan efisiensi kinerja dari algoritma klasifikasi.

Algoritma *Fast Correlation Based Filter* adalah salah satu algoritma seleksi fitur yang dikembangkan oleh Yu dan Huan (2003). Konsep utama dari algoritma ini adalah menghilangkan fitur-fitur yang tidak relevan serta menyaring fitur-fitur yang *redundant* terhadap fitur-fitur yang lain. Berdasarkan penelitian yang dilakukan Yu dan Huan (2003) diperoleh hasil bahwa *Fast Correlation Based Filter* sangat efisien dalam melakukan seleksi fitur serta memberikan performa yang baik bagi kinerja algoritma klasifikasi, baik dari segi waktu maupun akurasi hasil klasifikasi. Penelitian Yu dan Huan (2003)

menggunakan sepuluh *data sets* dan dievaluasi hasilnya dengan menggunakan algoritma klasifikasi C4.5 dan NBC.

Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan metode seleksi fitur *Fast Correlation Based Filter* pada klasifikasi data menggunakan Algoritma *Voting Feature Intervals* 5.

Ruang Lingkup

Ruang lingkup penelitian ini yaitu penerapan seleksi fitur menggunakan algoritma *Fast Correlation Based Filter* pada klasifikasi data menggunakan algoritma *Voting Feature Intervals* 5 dengan bobot setiap fitur pada semua data seragam.

Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan informasi, pengetahuan serta kontribusi terutama untuk memperbaiki kinerja algoritma klasifikasi *Voting Feature Intervals* 5 menggunakan seleksi fitur sehingga menjadi lebih efektif dan efisien.

TINJAUAN PUSTAKA

Seleksi Fitur

Seleksi fitur adalah salah satu tahapan praproses yang berguna terutama dalam mengurangi dimensi data, menghilangkan data yang tidak relevan, serta meningkatkan hasil akurasi (Yu dan Liu 2003). Jain dan Zongker (1997) mendefinisikan masalah seleksi fitur sebagai berikut: diberikan sekumpulan fitur lalu dipilih beberapa fitur yang mampu memberikan hasil yang terbaik pada klasifikasi. Ada dua titik berat seleksi fitur dengan pendekatan *machine learning* menurut Portiale (2002) yaitu memilih fitur yang akan digunakan dan menjelaskan secara konsep bagaimana mengkombinasikan fitur-fitur tersebut untuk menghasilkan konsep induksi yang benar atau hasil yang sesuai.

Seleksi fitur digunakan memberikan karakterisik dari data. Seleksi fitur merupakan salah satu penelitian yang banyak dilakukan di berbagai bidang seperti *pattern recognition, process identification,* dan *time series modelling*.

***Fast Correlation Based Filter* (FCBF)**

Algoritma *Fast Correlation Based Filter* adalah algoritma seleksi fitur yang dikembangkan oleh Yu dan Liu (2003). Algoritma ini didasarkan pada pemikiran bahwa suatu fitur yang baik adalah fitur-fitur yang relevan terhadap kelas tapi tidak *redundant* terhadap fitur-fitur relevan yang lain. Oleh karena itu, Lei Yu dan Huan Liu melakukan dua pendekatan dengan mengukur korelasi antara dua variabel acak yaitu berdasar pada *classical linear correlation/ linear correlation coefficient* dan berdasar pada teori informasi.

Pendekatan *linear correlation coefficient* untuk setiap variabel (X, Y) dirumuskan sebagai berikut

$$r = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2}\sqrt{\sum_i (y_i - y_i)^2}}$$

$\overline{x_i}$ adalah rata-rata dari X dan $\overline{y_i}$ adalah rata-rata dari Y serta rentang nilai r berada antara –1 dan 1. Jika X dan Y memiliki korelasi maka nilai r adalah 1 atau -1. Jika tidak berkorelasi maka nilai r adalah 0. Ada beberapa keuntungan menggunakan pendekatan ini yaitu mudah untuk menghilangkan fitur-fitur yang tidak relevan dengan memilih fitur yang nilai korelasinya 0 dan membantu mengurangi *redundant* pada fitur-fitur yang sudah dipilih. Namun pendekatan ini juga memiliki keterbatasan yaitu membutuhkan fitur-fitur yang memiliki nilai-nilai numerik.

Untuk mengatasi hal ini dilakukan pendekatan yang kedua yaitu pendekatan berdasar pada *information-theorical concept of entropy* (mengukur ketidakpastian pada random variabel). *Entrophy* dari variabel X didefinisikan sebagai berikut:

$$H(x) = -\sum_i P(x_i)\log_2(P(x_i))$$

Entrophy dari variabel X jika diketahui variabel Y didefinisikan sebagai berikut:

$$H(X \mid Y) = -\sum_j P(y_j)\sum_i P(x_i \mid y_j)\log_2(P(x_i \mid y_j))$$

$P(x_i)$ adalah *prior probabilities* untuk semua nilai X dan $P(x_i \mid y_i)$ adalah *posterior probabilities* dari X jika diketahui nilai Y. Dari *entrophy* tersebut dapat diperoleh *Information Gain* sebagai berikut:

$$IG(X \mid Y) = H(X) - H(X \mid Y)$$

Untuk mengukur korelasi antar fitur, maka digunakan *symmetrical uncertainty*. Nilai *symmetrical uncertainty* berkisar pada rentang 0 sampai dengan 1. *Symmetrical uncertainty* dirumuskan sebagai berikut:

$$SU(X,Y) = 2\left[\frac{IG(X \mid Y)}{H(X)+H(Y)}\right]$$

Ada beberapa konsep yang digunakan dalam algoritma *Fast Correlation Based Filter* yaitu *predominant correlation* untuk menentukan korelasi antar fitur dengan kelas, dan *heuristic-heuristic* yang digunakan untuk mengasumsikan jika ada dua fitur saling *redundant* dan salah satunya harus dihilangkan. *Predominant correlation* didefinisikan pada Gambar 1, sedangkan *heuristic-heuristic* yang digunakan pada algoritma *Fast Correlation Based Filter* ditampilkan pada Gambar 2.

Definition 1 (Predominant Correlation). *The correlation between a feature* $F_i (F_i \in S)$ *and the class C is predominant iff* $SU_{i,c} \geq \delta, \text{and} \forall F_j \in S'(j \neq i)$, *there exists no* F_j *such that* $SU_{j,i} \geq SU_{i,c}$

if there exists such F_j *to a* fitur F_i, *we call it a redundant peer to* F_i *and use* S_{Pi} *to denote the set of all redundant peers to for* F_i. *Given* $F_i \in S'$ *and* $S_{P_i} (S_{P_i} \neq \varnothing)$, *we divide* S_{pi} *into two parts* S_{Pi}^+ *and* S_{Pi}^-, *where*

$$S_{P_i}^+ = \left\{F_j \mid F_j \in S_{P_i}, SU_{j,c} > SU_{i,c}\right\}$$

and

$$S_{P_i}^- = \left\{F_j \mid F_j \in S_{P_i}, SU_{j,c} \leq SU_{i,c}\right\}.$$

Definition 2 (Predominant Feature) *A feature is predominant to the class iff its correlation to the class is predominant or can become predominant after removing its redundant peers.*

Gambar 1 *Predominant correlation* (Yu 2003)

Heuristic 1 *(if* $S_{P_i}^+ = \varnothing$*). Treat* F_i *as a predominant feature, remove all* fitur *in* $S_{P_i}^-$*, and skip identifying redundant peers of them.*

Heuristic 2 *(if* $S_{P_i}^+ \neq \varnothing$*). Process all features in* $S_{P_i}^+$ *, before making a decision on* F_i*. If none of them become predominant, follow Heuristic 1; otherwise only remove* F_i *and decide whether or not to remove features in* $S_{P_i}^-$ *based on other* fitur *in S'.*

Heuristic 3 *(starting point). The feature with the largest* $SU_{i,c}$ *value is always predominant feature and can be a starting point to remove other feature.*

Gambar 2 *Heuristic* pada FCBF (Yu 2003)

Pseudocode dari algoritma *Fast Correlation Based Filter* ditampilkan pada Gambar 3. Jika diberikan data dengan N fitur dan kelas C, maka algoritma *Fast Correlation Based Filter* menentukan *predominant* fitur S_{best} untuk setiap kelas. Tahapan ini terdiri dari dua bagian. Tahapan pertama, menghitung nilai SU untuk setiap fitur, memilih fitur-fitur yang relevan lalu dimasukkan ke S'_{list} berdasarkan nilai *threshold* δ, dan mengurutkannya sesuai dengan nilai SU. Tahapan kedua, menghilangkan fitur-fitur yang *redundant*. Berdasarkan *Heuristic* 1, fitur F_p yang sudah ditentukan sebagai *predominant* fitur dapat digunakan untuk mem-*filter* fitur- fitur yang lain yang berada pada urutan di bawahnya. Tahapan kedua ini dimulai dari elemen pertama (*Heuristic* 3) pada S'_{list} sampai tidak ada lagi fitur yang dihilangkan dari S'_{list}. Jika F_p menemukan bahwa F_q adalah *redundant* maka F_q akan dihilangkan dari S'_{list} (*Heuristic* 2).

```
input  : S(F1, F2,…, FN,C)      // a training data set
         δ                      // a predefined threshold
output : Sbest                  // an optimal subset

begin
    for i = 1 to N do begin
        calculate SUi,c for Fi;
    if ( SUi,c ≥ δ )
        append Fi to S'list;
    end;
    order S'list in descending SUi,c value;
    Fp = getFirstElement(S'list);
    do begin
        Fq = getNextElement(S'list, Fp);
        if ( Fq < > NULL)
            do begin
                F'q = Fq;
                if ( SUp,q ≥ SUq,c )
                    remove Fq from S'list;
                    Fq = getNextElement(S'list, F'q');
                else Fq = getNextElement(S'list, Fq);
            end until (Fq = NULL);
        Fp = getNextElement(S'list, Fp);
    end until (Fp==NULL)
Sbest=S'list;
end;
```

Gambar 3 Algoritma FCBF (Yu 2003)

***Voting Feature Intervals 5* (VFI5)**

Voting Feature Intervals 5 adalah salah satu algoritma klasifikasi yang merepresentasikan deskripsi sebuah konsep oleh sekumpulan interval nilai-nilai fitur (Guvenir 1998). Klasifikasi menggunakan algoritma ini didasarkan pada *vote* dari nilai-nilai pada fitur. Algoritma ini disebut *non incremental classification algorithm* karena semua data *training* hanya diproses satu kali. Algoritma ini dikembangkan oleh Guvenir dan Demiroz (1998).

Cara kerja algoritma ini yaitu membuat interval dari setiap fitur menggunakan *instance-instance* yang terdapat pada fitur tersebut. Interval yang dibuat dapat berupa *range interval* atau *point interval*. *Point interva*l terdiri dari seluruh *end point* semua fitur secara berurut sedangkan *range interval* terdiri dari nilai-nilai antara dua *end point* yang berdekatan namun tidak termasuk kedua *end point* tersebut. Nilai *vote* setiap kelas akan disimpan pada setiap interval. Dengan demikian, sebuah interval dapat mempresentasikan beberapa kelas dengan menyimpan *vote* dari kelas-kelas tersebut, sehingga algoritma ini dapat dikatakan sebagai *multi-class feature projection based algorithm*.

Algoritma *Voting Feature Intervals* 5 terdiri dari dua tahap yaitu tahap pelatihan dan klasifikasi

1 Pelatihan

Langkah pertama yang dilakukan pada tahap ini adalah menentukan *end point* dari fitur f pada kelas data c. *End point* untuk fitur linier adalah fitur yang nilainya

$$feature\_vote[f,c] = \frac{interval\_class\_count[f,i,c]}{class\_count[c]}$$

memiliki urutan dan bisa dibandingkan tingkatannya yaitu berupa nilai minimum dan nilai maksimun kelas untuk setiap fitur. *End point* untuk fitur nominal adalah fitur yang nilainya tidak memiliki urutan dan tidak bisa dibandingkan tingkatannya yaitu semua nilai yang berbeda yang ada pada fitur kelas yang sedang diamati.

Setelah didapatkan nilai-nilai *end point*, langkah selanjutnya adalah mengurutkan nilai-nilai *end point* menjadi suatu interval dari fitur f. Untuk setiap kelas c dengan fitur f, dihitung jumlah *instance training* yang direpresentasikan sebagai *interval_class_count [f,i,c]*. Nilai yang dihasilkan dimasukkan ke dalam interval i sesuai dengan nilai fitur f dari *instance training* e (e_f) tersebut.

Jika interval i merupakan *point interval* dan nilai e_f sama dengan nilai pada batas bawah atau batas atas maka jumlah kelas data tersebut (e_f) pada interval i ditambah 1. Jika interval i merupakan *range interval* dan nilai e_f jatuh pada interval tersebut maka jumlah kelas data e_f pada interval i ditambah 1. Hasil dari proses tersebut merupakan jumlah *vote* kelas c pada interval i.

Untuk menghilangkan efek perbedaan distribusi setiap kelas, maka jumlah *vote* kelas c untuk fitur f pada interval i dibagi dengan *class_count[c]*, yaitu jumlah data pada kelas c. Hasil normalisasi ini direpresentasikan dalam *interval_class_vote[f,i,c]*. Nilai-nilai pada *interval_class_vote[f,i,c]* dinormalisasi sehingga jumlah *vote* dari beberapa kelas pada setiap fitur sama dengan 1. Normalisasi ini bertujuan agar setiap fitur memiliki kekuatan *voting* yang sama pada proses klasifikasi yang tidak dipengaruhi ukurannya. *Pseudocode* untuk tahapan pelatihan ditampilkan pada Lampiran 1.

2 Prediksi (Klasifikasi)

Tahapan ini dimulai dengan inisialisasi awal nilai *vote* masing-masing kelas dengan nilai 0. Untuk setiap fitur f, dicari nilai interval i dimana e_f jatuh. Nilai e_f adalah nilai fitur dari *instance test* e. Jika e_f tidak diketahui maka fitur tersebut tidak diikutsertakan dalam voting (memberi nilai *vote* nol untuk masing-masing kelas) sehingga fitur tersebut diabaikan. Jika e_f diketahui maka interval tersebut dapat ditemukan. Fitur tersebut akan memberi nilai *vote* untuk masing-masing kelas dengan rumus:

interval_class_count[f,i,c] merupakan *vote* fitur f yang diberikan untuk kelas c. Setiap fitur f mengumpulkan nilai *vote*nya kemudian dijumlahkan untuk memperoleh total *vote*. Kemudian kelas c yang memiliki nilai *vote* tertinggi diprediksi sebagai kelas dari data *test* e. *Pseudocode* algoritma klasifikasi *Voting Feature Intervals* 5 ditampilkan pada Lampiran 1.

K-Fold Cross Validation

K-Fold Cross Validation (Stone 1974 diacu dalam Fu 1994) adalah sebuah metode yang membagi himpunan contoh secara acak menjadi k himpunan bagian (*subset*). Pada metode ini dilakukan pengulangan sebanyak k kali untuk data pelatihan dan pengujian. Pada setiap pengulangan, satu *subset* digunakan untuk pengujian sedangkan *subset* sisanya digunakan untuk pelatihan.

Data awal dibagi menjadi k *subset* secara acak dengan ukuran *subset* yang hampir sama dengan mempertahankan perbandingan antar kelas. Pada iterasi pertama, *subset* satu menjadi data pengujian sedangkan *subset* lainnya menjadi data pelatihan. Pada iterasi kedua, *subset* kedua digunakan sebagai data pengujian dan *subset* lainnya sebagai data pelatihan, dan seterusnya hingga seluruh *subset* digunakan sebagai data pengujian.

METODE PENELITIAN

Penelitian ini dilakukan dalam beberapa tahap. Tahapan-tahapan yang dilakukan ditampilkan pada Gambar 4.

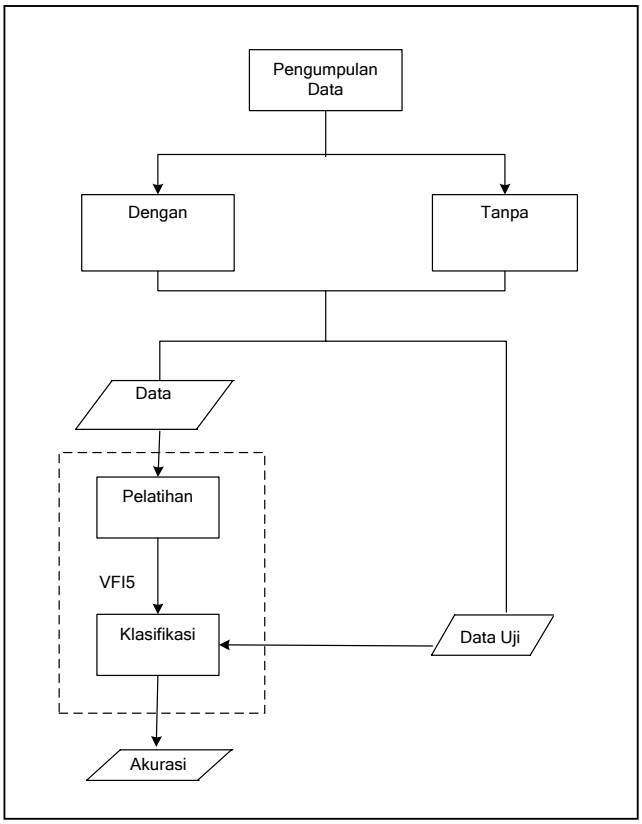
Pengumpulan Data

Data yang digunakan pada penelitian ini diambil dari *UCI repository of machine learning database* (http://www.ics.uci.edu/~mlearn/MLRespository). Penelitian ini menggunakan empat data yang memiliki ukuran yang berbeda. Spesifikasi data yang digunakan disajikan pada Tabel 1.

Tabel 1 Spesifikasi data

Nama data	Jumlah Fitur	Jumlah Kelas	Jumlah *Instance*
Dermatology	34	6	366
Lung Cancer	54	3	32
Promoters	57	2	106
Splice	61	3	3190

Data yang digunakan pada penelitian ini adalah data yang memiliki dimensi yang beragam. Hal ini dimaksudkan agar perbedaan hasil akurasi klasifikasinya dapat terlihat ketika data diolah tanpa seleksi fitur maupun menggunakan seleksi fitur. Data yang digunakan dalam penelitian ini dibatasi hanya untuk data yang memiliki fitur-fitur



Gambar 4 Tahapan penelitian

nominal dan fitur-fitur linier yang nilainya sudah didiskretisasi. Hal ini merupakan salah satu syarat yang harus dipenuhi dalam penerapan algoritma *Fast Correlation Based Filter*.

Praproses Data

Tahapan praproses data merupakan tahapan yang paling utama. Pada tahapan ini data diolah menggunakan seleksi fitur dan tanpa seleksi fitur.

Langkah pertama yang dilakukan pada tahapan ini yaitu menghilangkan fitur-fitur yang memiliki data tidak lengkap seperti fitur-fitur yang memiliki data kosong dan menghilangkan fitur-fitur linier yang nilainya berupa rentang. Fitur-fitur yang memiliki nilai kosong mampu mempengaruhi hasil klasifikasi sehingga harus dihilangkan. Selanjutnya, tahapan ini dibagi menjadi dua bagian yaitu pengolahan data menggunakan seleksi fitur dan tanpa menggunakan seleksi fitur. Data yang diolah tanpa menggunakan seleksi fitur akan langsung diklasifikasi menggunakan algoritma *Voting Feature Intervals* 5 lalu dihitung akurasinya.

Data yang diolah dengan seleksi fitur akan diseleksi fitur-fiturnya menggunakan algoritma *Fast Correlation Based Filter*. Algoritma *Fast Correlation Based Filter* akan menghitung nilai *Symmetrical Uncertainty* dari masing-masing fitur data. Nilai ini akan digunakan untuk menghilangkan fitur-fitur yang tidak relevan serta *redundant* terhadap fitur-fitur yang lain. Fitur yang akan digunakan adalah fitur-fitur yang memiliki nilai korelasi terhadap kelas lebih tinggi dibanding nilai korelasi fitur tersebut terhadap fitur yang lain. Salah satu parameter yang digunakan untuk menyeleksi fitur adalah nilai *threshold*. Nilai *threshold* berada pada rentang 0 sampai dengan 1.

Hasil dari tahapan ini yaitu fitur-fitur data yang akan digunakan untuk tahapan klasifikasi. Fitur-fitur yang terpilih ini didasarkan pada nilai *threshold* yang sudah ditentukan. Hal ini berarti pada tahapan klasifikasi selanjutnya tidak semua fitur dari data digunakan. Hanya fitur-fitur tertentu

yang memenuhi syarat saja yang dapat digunakan.

Klasifikasi Menggunakan Algoritma *Voting Feature Intervals* 5

Tahapan klasifikasi *Voting Feature Intervals* 5 terdiri dari dua proses yaitu pelatihan dan klasifikasi. Dua tahapan ini berlaku baik bagi data yang sebelumnya mengalami seleksi fitur maupun tanpa seleksi fitur. Data yang digunakan pada tahapan ini juga dibagi menjadi dua bagian yaitu data pelatihan dan data pengujian.

Data Latih dan Data Uji

Penelitian ini menggunakan metode 3-*fold cross validation*. Oleh karena itu, data yang digunakan dibagi menjadi tiga *subset* secara acak yang masing-masing *subset* memiliki jumlah *instance* dan perbandingan jumlah kelas yang sama. Pembagian *subset* untuk setiap data tergantung pada jumlah *instance* dan jumlah kelas masing-masing.

Pembagian data ini digunakan pada proses iterasi klasifikasi. Iterasi dilakukan sebanyak tiga kali karena penelitian ini menggunakan metode 3-*fold cross validation*. Pada setiap iterasi, satu *subset* digunakan untuk pengujian sedangkan *subset-subset* lainnya digunakan untuk pelatihan.

Pelatihan

Subset data yang digunakan untuk pelatihan akan menjadi input bagi algoritma *Voting Feature Intervals 5*. Langkah pertama yang dilakukan pada tahapan pelatihan yaitu membuat interval dari masing-masing fitur berdasarkan nilai *end point* masing-masing fitur untuk setiap kelasnya. Setelah interval masing-masing fitur terbentuk maka dimulailah proses *voting* pada algoritma. *Voting* yang dilakukan yaitu menghitung jumlah data untuk setiap kelas pada interval tertentu. Masing-masing kelas pada rentang interval tertentu memiliki nilai *vote* yang berbeda-beda. Nilai *vote* tersebut akan dinormalisasi untuk mendapatkan nilai *vote* akhir pada masing-masing fitur.

Proses pelatihan dilakukan setiap iterasi, sehingga proses pelatihan dilakukan sebanyak tiga kali. Proses pelatihan setiap iterasi mungkin memberikan nilai *vote* yang berbeda-beda setiap fiturnya tergantung pada *subset* yang digunakan sebagai data pelatihan pada iterasi tersebut.

Pengujian

Pada tahapan pengujian atau klasifikasi setiap nilai fitur dari data pengujian akan diperiksa letaknya pada interval. *Vote-vote* setiap kelas untuk setiap fitur pada interval yang bersesuaian diambil dan kemudian dijumlahkan. Kelas dengan nilai *vote* tertinggi menjadi kelas prediksi dari data pengujian tersebut.

Proses pengujian menggunakan data uji yang telah ditentukan sebelumnya dalam proses iterasi. Data uji yang digunakan disesuaikan dengan *subset* data pelatihan yang digunakan.

Akurasi

Penghitungan tingkat akurasi diperoleh berdasarkan data pengujian. Tingkat akurasi diperoleh dengan rumus

$$\text{tingkat akurasi} = \frac{\sum \text{data uji benar diklasifikasi}}{\sum \text{total data uji}}$$

Tingkat akurasi menunjukkan tingkat kebenaran pengklasifikasian data terhadap kelas yang sebenarnya. Semakin rendah nilai akurasi maka semakin tinggi kesalahan klasifikasi. Tingkat akurasi yang baik adalah tingkat akurasi yang mendekati nilai 100%.

Tingkat akurasi dihitung, baik bagi data yang mengalami seleksi fitur maupun tanpa seleksi fitur. Tingkat akurasi inilah yang menjadi pembeda antara data yang mengalami seleksi fitur maupun tanpa seleksi fitur.

Spesifikasi Pengembangan

Aplikasi yang digunakan pada penelitian ini dibangun dengan menggunakan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

Perangkat keras berupa komputer personal:

1 Prosesor Intel Pentium IV 3.2 GHz
2 Memori 512 MB
3 *Harddisk* 120 GB
4 Monitor 17”
5 Alat input *mouse* dan *keyboard*

Perangkat lunak:

1 Microsoft® Windows XP Professional SP2
2 Microsoft® Internet Explorer 6.0
3 PHP 5.0.4
4 Apache Webserver

HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah data Lung Cancer, Dermatology, Promoters, dan Splice. Data Lung Cancer dan Dermatology merupakan data medis sedangkan Promoters dan Splice merupakan data *sequence* DNA bakteri E.Coli. Data yang digunakan dalam penelitian ini adalah data yang sudah pasti nilainya (diskret) bukan data yang kontinyu.

Tahap pertama yang dilakukan dalam penelitian ini adalah penghilangan fitur-fitur yang memiliki nilai-nilai kosong dan penghilangan fitur-fitur linier yang nilainya berupa rentang. Fitur yang memiliki nilai kosong dihilangkan, agar tidak mempengaruhi nilai akurasi klasifikasi. Jumlah fitur yang dihilangkan dari keempat data tersebut dapat dilihat pada Tabel 2.

Tabel 2 Spesifikasi fitur

Nama data	Jumlah Fitur	Jumlah Fitur yang dibuang	Jumlah Fitur yang digunakan
Lung Cancer	56	2	54
Dermatology	34	1	33
Promoters	58	1	57
Splice	61	1	60

Keempat data tersebut selanjutnya dibagi menjadi tiga *subset*. *Subset-subset* inilah yang nantinya digunakan dalam tahapan klasifikasi sebagai data pelatihan dan data pengujian. Pembagian data menjadi *subset* tergantung dari jumlah *instance* tiap-tiap data. *Subset* yang terbentuk memiliki jumlah *instance* yang hampir sama dengan mempertahankan proporsi perbandingan antar kelas. Pembagian data secara keseluruhan dari keempat data tersebut disajikan pada Tabel 3.

Tabel 3 Pembagian data

Nama data	S1	S2	S3	Total
Lung Cancer	11	11	10	32
Dermatology	122	122	122	366
Promoters	36	35	35	106
Splice	1064	1063	1063	3190

Seleksi Fitur Menggunakan *Fast Correlation Based Filter*

Pada penelitian ini, data yang diolah menggunakan seleksi fitur akan dihitung tingkat relevansinya terhadap kelas menggunakan algoritma *Fast Correlation Based Filter*. Fitur-fitur data akan mengalami penyeleksian sehingga *output* dari tahapan ini adalah membuang fitur-fitur yang tidak memenuhi syarat.

Salah satu parameter yang digunakan untuk penyeleksian fitur adalah nilai *threshold*. Nilai *threshold* adalah nilai batas korelasi (nilai minimum dari nilai *Symmetrical Uncertainty*) yang digunakan untuk menyeleksi fitur. Nilai *threshold* berkisar pada rentang 0 sampai dengan 1. Pada penelitian ini nilai *threshold* yang digunakan yaitu 0, 0.1, 0.13, 0.2, 0.3, 0.4, dan 0.5. Jumlah fitur data yang terseleksi dengan ketujuh nilai *threshold* tersebut disajikan pada Tabel 4.

Tabel 4 Jumlah fitur yang terseleksi untuk beragam nilai *threshold*

Nilai δ	Lung Cancer	Dermatology	Promoters	Splice
0	3	14	6	22
0.1	3	13	6	6
0.13	2	13	4	5
0.2	2	10	3	3
0.3	2	4	0	0
0.4	0	3	0	0
0.5	0	0	0	0

Dari Tabel 4 dapat dilihat bahwa jumlah fitur yang terseleksi yang dapat digunakan untuk tahapan klasifikasi selanjutnya, berkurang lebih dari setengahnya dari jumlah fitur asalnya. Hal ini menunjukkan bahwa tidak semua fitur relevan dengan kelas dan tidak semua fitur juga memiliki tingkat korelasi yang tinggi terhadap kelasnya dibandingkan korelasinya terhadap fitur yang lain. Selain itu, tidak semua nilai

threshold dapat digunakan untuk menyeleksi fitur. Walaupun nilai *threshold* berada pada rentang 0 sampai dengan 1, nilai *threshold* yang dapat digunakan adalah nilai *threshold* yang bisa menghasilkan fitur-fitur yang terseleksi. Nilai *threshold* 0.5 tidak dapat digunakan karena tidak menghasilkan fitur yang terseleksi.

Klasifikasi Tanpa Menggunakan Seleksi Fitur

Setelah mengalami pengurangan fitur pada tahap praproses, data yang akan dianalisis tanpa menggunakan seleksi fitur, diklasifikasikan menggunakan algoritma *Voting Feature Intervals* 5. Klasifikasi dilakukan sebanyak tiga kali iterasi. Iterasi pertama menggunakan data pelatihan yang terdiri dari *subset* S2 dan S3, dan *subset* S1 sebagai data pengujian. Iterasi Kedua menggunakan *subset* S1 dan S3 sebagai data pelatihan dan *subset* S2 sebagai data pengujian. Iterasi ketiga menggunakan *subset* S1 dan S2 sebagai data pelatihan dan *subset* S3 sebagai data pengujian. Tiap-tiap data pengujian pada tiap iterasi akan dihitung tingkat akurasinya.

Hasil akurasi dari masing-masing iterasi dan rataan akurasi dari keempat data tersebut disajikan pada Tabel 5.

Tabel 5 Akurasi tanpa seleksi fitur (%)

Nama data	Iterasi			Rataan
	1	2	3	
Lung Cancer	45.45	63.63	60.00	56.36
Dermatology	92.62	95.08	98.36	95.35
Promoters	83.33	88.57	82.85	84.92
Splice	89.75	89.55	90.68	90.00
Rataan Akurasi				**81.66**

Dari Tabel 5 dapat dilihat bahwa akurasi terkecil terdapat pada data Lung Cancer sedangkan akurasi terbesar terdapat pada data Dermatology. Hal itu menunjukkan bahwa tingkat kesalahan klasifikasi data pengujian pada Lung Cancer lebih tinggi dari data yang lainnya. Hal itu dapat disebabkan jumlah *instance* pada data Lung Cancer jauh lebih sedikit daripada jumlah fiturnya sehingga *instance-instance* yang ada belum dapat mewakili setiap fitur untuk dapat diklasifikasikan pada kelas tertentu. Setiap fitur juga belum dapat mewakili relevansi terhadap kelas tertentu.

Dari ketiga iterasi yang dilakukan, akurasi cenderung mengalami peningkatan dari iterasi satu ke iterasi yang lainnya. Peningkatan ini membuat rataan akurasi menjadi lebih baik. Pembagian data menjadi *subset* dan dilakukan iterasi dalam pengolahannya dimaksudkan untuk mengurangi tingkat kesalahan klasifikasi sehingga akurasi dari data menjadi lebih baik.

Klasifikasi Menggunakan Seleksi Fitur

Fitur-fitur yang sudah terseleksi dari setiap nilai *threshold* yang berbeda dihitung tingkat akurasinya. Klasifikasi dilakukan sebanyak tiga kali iterasi sama seperti tahapan klasifikasi tanpa menggunakan seleksi fitur. Data dari fitur-fitur tersebut dibagi menjadi tiga *subset* dengan komposisi pembagian yang sama setiap *subset*.

Data pelatihan dan data pengujian yang digunakan pada setiap iterasi mempunyai susunan yang sama seperti klasifikasi tanpa menggunakan seleksi fitur. Dengan tiga kali iterasi untuk tujuh nilai *threshold* yang berbeda maka pengulangan dilakukan sebanyak 21 kali untuk setiap data tertentu. Hasil tingkat akurasi dari setiap pengulangan dapat dilihat pada Lampiran 2.

Berdasarkan hasil akurasinya, rataan akurasi tertinggi terdapat pada nilai *threshold* 0. sedangkan terendah terdapat pada nilai *threshold* 0.2. Untuk nilai *threshold* 0.3 sampai dengan 0.5 tidak semua data memiliki nilai akurasi sehingga tidak dapat dibandingkan dengan tingkat akurasinya dengan yang lain. Secara umum tingkat akurasi data mengalami penurunan jika dinaikkan nilai *threshold*nya.

Kecenderungan nilai akurasi setiap data untuk setiap nilai *threshold* tidak dapat diprediksi apakah mengalami kenaikan atau penurunan. Data Lung Cancer mengalami penurunan jika nilai *threshold* dinaikkan, Dermatology mengalami penurunan untuk nilai *threshold* 0.1 dan 0.13 lalu naik pada nilai *threshold* 0.2, nilai akurasi data Promoters tertinggi terdapat pada nilai *threshold* 0.13, sedangkan Splice sama seperti Lung Cancer mengalami penurunan jika nilai *threshold* naik.

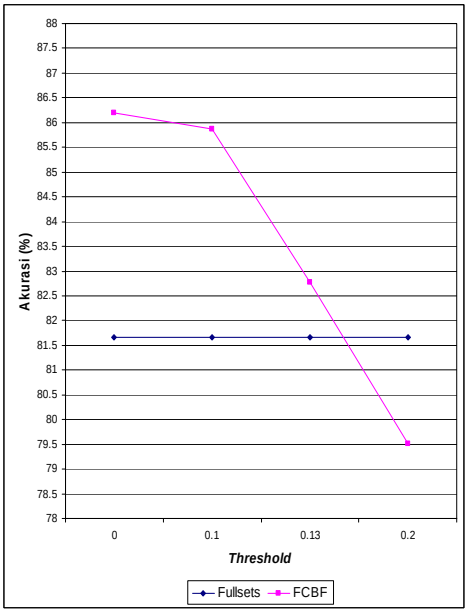
Perbandingan Antara Klasifikasi Menggunakan Seleksi Fitur dan Tanpa Menggunakan Seleksi Fitur

Rataan akurasi antara klasifikasi dengan menggunakan seleksi fitur dan tanpa menggunakan seleksi fitur (*fullsets*) untuk setiap nilai *threshold* yang berbeda dapat dilihat pada Tabel 6 dan Gambar 5.

Tabel 6 Perbandingan akurasi antara *fullsets* dengan beragam nilai *threshold*

	Rataan (%)
Fullsets	81.66
Nilai *threshold* 0	86.19
Nilai *threshold* 0.1	85.87
Nilai *threshold* 0.13	82.78
Nilai *threshold* 0.2	79.51

Pada gambar 5 dapat dilihat bahwa akurasi data menggunakan seleksi fitur lebih baik dibandingkan tanpa seleksi fitur. Walaupun pada nilai *threshold* 0.2 nilai akurasi lebih kecil dari *fullsets* tapi rataan nilai akurasi dengan seleksi fitur lebih baik. Dari gambar tersebut, nilai akurasi tertinggi dengan fitur seleksi yang membedakan dengan *fullsets* adalah nilai *threshold* 0.

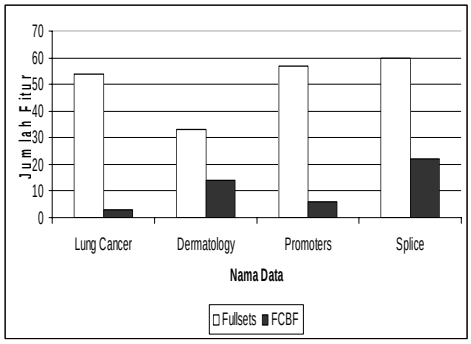


Gambar 5 Perbandingan akurasi antara *fullsets* dengan beragam nilai *threshold*

Perbandingan jumlah fitur asal dengan jumlah fitur yang sudah mengalami seleksi (untuk nilai *threshold* 0) dapat dilihat pada Tabel 7 dan Gambar 6.

Tabel 7 Jumlah fitur terseleksi

Nama data	Jumlah Fitur asal	Jumlah Fitur terseleksi
Lung Cancer	54	3
Dermatology	33	14
Promoters	57	6
Splice	60	22



Dari Gambar 6 terlihat bahwa jumlah fitur mengalami pengurangan lebih dari setengahnya dibandingkan jumlah fitur asalnya. Hal itu menunjukkan bahwa tidak semua fitur relevan terhadap hasil klasifikasi. Kesalahan klasifikasi bisa jadi disebabkan karena banyaknya fitur yang kurang relevan terhadap hasil sehingga menurunkan tingkat akurasi data.

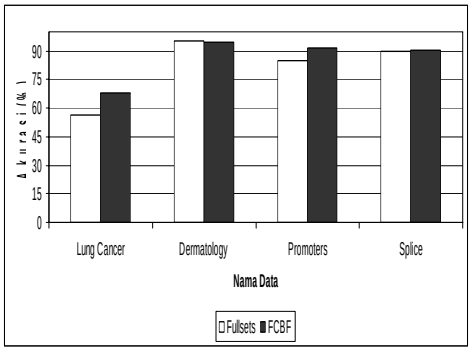
Perbandingan hasil akurasi klasifikasi tanpa seleksi fitur dan dengan seleksi fitur (untuk nilai *threshold* 0) setiap data dapat dilihat pada Tabel 8 dan Gambar 7.

Tabel 8 Perbandingan nilai akurasi

Nama Data	Akurasi *Fullsets* (%)	Akurasi dengan FCBF (%)
Lung Cancer	56.36	68.18
Dermatology	95.35	94.81
Promoters	84.92	91.48
Splice	90.00	90.28
Rataan	**81.66**	**86.19**

Berdasarkan Gambar 7, secara umum nilai akurasi data mengalami kenaikan jika dilakukan seleksi fitur sebelumnya. Kenaikan akurasi tertinggi terdapat pada data Lung Cancer sedangkan pada Dermatology akurasi data mengalami penurunan jika dilakukan seleksi fitur. Hal itu disebabkan karena fitur-fitur pada data Dermatology merupakan fitur-fitur linier sehingga mempengaruhi nilai korelasi fitur. Walaupun demikian penurunan yang terjadi

pada data Dermatology tidak terlalu signifikan dibandingkan kenaikan akurasi pada data yang lain.



Gambar 7 Perbandingan nilai akurasi

Tingkat akurasi pada klasifikasi menggunakan seleksi fitur lebih baik daripada tidak menggunakan seleksi fitur menunjukkan bahwa keberadaan fitur mempengaruhi hasil klasifikasi. Fitur-fitur yang tidak mempunyai relevansi terhadap kelas berpengaruh terhadap tingkat akurasi. Seleksi fitur memilih beberapa fitur yang mampu memberikan hasil terbaik pada klasifikasi. Selain itu, seleksi fitur juga berguna untuk mengurangi ruang penyimpanan data. Misalnya data tentang pemrosesan *text document* yang memiliki fitur-fitur yang banyak, dengan seleksi fitur dapat dipilih hanya fitur-fitur yang relevan saja terhadap hasil klasifikasi.

KESIMPULAN DAN SARAN

Kesimpulan

Seleksi fitur adalah salah satu tahapan praproses klasifikasi yang dilakukan dengan cara memilih fitur-fitur yang mampu memberikan hasil yang terbaik pada klasifikasi data. Seleksi fitur digunakan untuk mengurangi dimensi data dan meningkatkan akurasi klasifikasi.

Salah satu parameter yang digunakan untuk menyeleksian fitur pada algoritma *Fast Correlation Based Filter* adalah penentuan nilai *threshold*. Semakin tinggi nilai *threshold* maka fitur yang terseleksi akan semakin sedikit. Penentuan nilai *threshold* yang berbeda menghasilkan nilai akurasi yang berbeda pula. Nilai akurasi tertinggi pada penelitian ini terdapat pada nilai *threshold* 0.

Perbandingan hasil akurasi klasifikasi data dengan seleksi fitur jauh lebih baik daripada tanpa seleksi fitur. Dari keempat data yang digunakan, tingkat akurasi yang diperoleh masing-masing data tanpa dan dengan seleksi fitur antara lain Lung Cancer 56.4% menjadi 61.2%, Dermatology 95.35% menjadi 94.81%, Promoters 84.92% manjadi 91.48% dan Splice 90% menjadi 90.28%. Rataan dari keempat nilai akurasi tersebut meningkat yaitu 81.66% menjadi 86.2%. Hal ini menunjukkan bahwa seleksi fitur mampu meningkatkan nilai akurasi.

Jumlah fitur yang digunakan dalam proses klasifikasi berkurang hampir lebih dari setengah dari jumlah fitur asalnya jika sebelumnya dilakukan seleksi fitur. Hal ini menunjukkan bahwa seleksi fitur bermanfaat untuk mengurangi dimensi data terutama untuk data yang berukuran besar. Data yang berukuran besar seperti data DNA manusia memerlukan tempat penyimpanan yang besar, sehingga dengan seleksi fitur data yang besar dapat dikurangi dimensinya dengan cara memilih fitur-fitur yang relevan saja.

Saran

Penelitian tentang seleksi fitur masih terus berkembang. Penelitian selanjutnya dapat mencoba menerapkan seleksi fitur pada algoritma klasifikasi yang lain. Algoritma seleksi fitur yang digunakan pun bisa bermacam-macam. Agar hasil akurasi dapat terlihat perbedaanya, maka sebaiknya data yang digunakan harus memiliki ukuran dimensi yang sangat besar, misalnya data DNA manusia.

DAFTAR PUSTAKA

Blake, A., dan C Merz. 1998. *UCI respository of machine learning databases.* http://www.ics.uci.edu/ ~mlearn /MLRespository.html.

Fu, L. 1994. *Neural Network in Computers Intelligence*. Singapura: McGraw-Hill.

Guvenir, H.A. 1998. *A Classification Learning Algorithm Robust to Irrelevant Features*. http://www.cs.bilkent.edu.tr/ tech-reports/1998/BU-CEIS-9810.ps.gz.

Guvenir, H.A., dan G Demiroz. 1998. *Learning Diagnosis of Erythemato-Squamous Diseases using Voting*

Feature Intervals. Artificial Intelligence in Medicine, 13(3), 147-165.

Jain, A., dan D Zongker. 1997. *Selection Feature: Evaluation, Application, and Small Sample Performance*. IEEE Transaction on Pattern Analysis and Machine Intteligence : 153-158.

Portinale, L., dan L Saitta. 2002. *Feature Selection*. http://citeseer.ist.psu.edu.

Yu, L., dan H Liu. 2003. *Feature Selection for High Dimensional Data: A Fast Correlation-Based Filter Solution*. www.hpl.hp.com/conferences/icml2003/papers/144.pdf.