

## Implementasi Seleksi Fitur dengan *Backward Elimination* untuk Klasifikasi Prediksi Perceraian

Muhammad Alif Raihan<sup>1</sup>, Pratama Haryandi<sup>2</sup>, Raihan Adyatma Subagja<sup>3</sup>, Riduwan Purnaminyan<sup>4</sup>, Nurul Chamidah<sup>5</sup>  
Program Studi Informatika / Fakultas Ilmu Komputer  
Universitas Pembangunan Nasional Veteran Jakarta  
Jl. RS Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia  
muhammadalif@upnvj.ac.id<sup>1</sup>, pratamah@upnvj.ac.id<sup>2</sup>, raihanaas@upnvj.ac.id<sup>3</sup>, riduwanp@upnvj.ac.id<sup>4</sup>,  
nurul.chamidah@upnvj.ac.id<sup>5</sup>

**Abstrak.** Berdasarkan studi mengenai prediksi perceraian yang memanfaatkan “Divorce Predictors Scale” yang berdasar pada terapi pasangan Gottman, performa model untuk memprediksi perceraian dapat ditingkatkan dengan menerapkan metode seleksi fitur *correlation-based feature selection*. Meskipun demikian, apabila data tidak dilakukan prapemrosesan seperti menghapus data yang duplikat, maka prediksi model dapat membias ke kelas data yang duplikat yang tentunya berdampak negatif bagi model. Sehingga dalam ruang lingkup penelitian ini, sebelum melatih model prediksi perceraian dengan data “Divorce Predictors Scale”, dilakukan prapemrosesan data dengan menghapus data duplikat dan melakukan normalisasi fitur dengan *min-max*. Selanjutnya metode seleksi fitur *backward elimination* diterapkan ke empat algoritma klasifikasi. Fitur yang terseleksi adalah 6 fitur yang menghasilkan performa terbaik bagi masing-masing algoritma dengan pembagian data *hold-out*. Keenam fitur tersebut divalidasi ke setiap algoritma klasifikasi dengan *stratified 10-fold cross-validation*. Hasilnya, algoritma *naive Bayes* menghasilkan performa terbaik dengan akurasi 97.34%, presisi 100%, dan *f-score* 96.88%.

**Kata Kunci:** Seleksi fitur, *backward elimination*, klasifikasi, perceraian.

### 1. Pendahuluan

Penelitian yang dilakukan oleh Yöntem *et al.* pada [1] adalah penelitian tentang faktor-faktor yang dapat memprediksi terjadinya perceraian bagi pasangan yang menikah di masa mendatang. Penelitiannya tersebut menggunakan *Divorce Predictors Scale* (DPS) yang telah dikembangkan pada penelitiannya sebelumnya [2] berdasarkan penelitian Gottman [3] pada pasangan yang baru menikah tentang terapi pasangan dan faktor-faktor yang dapat mengestimasi pasangan yang mungkin akan bercerai di masa mendatang untuk mengestimasi sampel di negara Turki.

Yöntem *et al.* pada [1] menggunakan *correlation-based feature selection* untuk seleksi fitur, yang kemudian 6 fitur dengan korelasi tertinggi yang akan digunakan untuk pemodelan. Performa dari 3 model prediksi dibandingkan antara model yang menggunakan seluruh fitur sebelum seleksi dengan model yang menggunakan hanya 6 fitur yang telah diseleksi. Hasilnya adalah algoritma ANN (*artificial neural network*) dengan 6 fitur yang diseleksi menghasilkan performa terbaik, dengan akurasi 98.82%. Meskipun demikian, data yang digunakan tidak melalui tahapan praproses dan pembagian data untuk memvalidasi model tidak disebutkan.

Pada penelitian Wah *et al.* mengenai perbandingan beberapa metode seleksi fitur [4] yang membandingkan metode *filter* dan metode *wrapper* untuk memaksimalkan tingkat akurasi klasifikasi. Metode *filter* yang digunakan adalah *correlation based feature selection* dan *information gain*, sedangkan metode *wrapper* yang digunakan adalah *sequential forward* dan *sequential backward elimination*. Algoritma klasifikasi yang digunakan adalah regresi logistik. Penelitiannya menggunakan 3 dataset nyata yang didapatkan dari UCI Machine Learning Repository, yaitu Pima Indians Diabetes, Breast Cancer Wisconsin, dan Spambase. Performa metode seleksi fitur didasarkan pada akurasi klasifikasi, *Akaike information criterion* (AIC), *Bayesian information criterion* (BIC), *area under receiver operating characteristic curve* (AUC), serta sensitivitas dan spesifisitas klasifikasi. Hasilnya adalah metode *wrapper* dapat memilih fitur yang lebih signifikan dibandingkan dengan metode *filter*.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengembangkan penelitian Yöntem *et al.*, dengan menggunakan data yang sama, dan dilakukan beberapa perbaikan atau pengembangan yang tidak dilakukan pada penelitian tersebut seperti dengan melakukan pembuangan data duplikat, normalisasi data, seleksi fitur dengan *backward elimination* (metode *wrapper*), pembagian data yang digunakan, dan algoritma yang digunakan. Pada

penelitian ini akan digunakan perbandingan hasil evaluasi dari 4 algoritma klasifikasi yaitu *decision tree*, *random forest*, *Gaussian naive Bayes* dan *multilayer perceptron* dengan parameter yang telah ditentukan untuk mencari model terbaik yang dapat dijadikan sebagai alternatif dari model penelitian yang dilakukan oleh Yöntem *et al.*

## 2. Tinjauan Pustaka

### 2.1 Penskalaan Fitur Min-Max

*Min-max normalization* atau *min-max rescaling* merupakan sebuah algoritma untuk menormalisasi skala jarak antar setiap data ke dalam skala jarak tertentu. Normalisasi dilakukan untuk menghindari terjadinya bias yang dapat disebabkan oleh skala jarak antardata dikarenakan data dengan nilai berskala besar cenderung mendominasi data dengan nilai berskala kecil [5]. *Min-max normalization* dapat dilakukan dengan rumus berikut.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Keterangan formula di atas sebagai berikut :

$x'$  = nilai data terbaru yang sudah di normalisasi  
 $x$  = nilai data awal  
 $\min(x)$  = nilai minimum  
 $\max(x)$  = nilai maksimum

### 2.2 Seleksi Fitur Backward Elimination

*Backward elimination* (atau BFS, *backward feature selection*) merupakan sebuah algoritma untuk menyeleksi fitur menggunakan kumpulan kombinasi fitur demi mencari kombinasi terbaik secara rekursif. BFS bekerja dengan cara melakukan pengujian terhadap seluruh fitur terlebih dahulu kemudian secara bertahap mengurangi fitur yang tidak signifikan berdasarkan perbandingan evaluasi hasil uji yang didapatkan dari setiap kombinasi fitur tersebut.

### 2.3 Decision Tree

*Decision tree* adalah algoritma metode klasifikasi dan prediksi yang umum digunakan dalam pengambilan keputusan. *Decision tree* akan mencari solusi dari permasalahan dengan suatu konsep menyederhanakan data menjadi sebuah pohon keputusan. Setiap pohon memiliki cabang dan setiap cabang mewakili atribut yang harus dipenuhi sampai cabang selanjutnya hingga tidak terdapat cabang lagi. Langkah yang diperlukan untuk membuat pohon keputusan adalah mencari nilai *entropy* [6]. Berikut ini adalah rumus untuk mencari nilai *entropy*.

$$Entropy(S) = -\sum_{i=1}^n p_i * \log_2 p_i \quad (2)$$

Di mana :

$S$  = himpunan kasus  
 $n$  = jumlah partisi  $S$   
 $p_i$  = proporsi dari  $S_i$  terhadap  $S$

Setelah menghitung nilai *entropy* dari setiap kasus, maka digunakan informasi *gain* untuk melakukan pemisahan objek. Berikut ini adalah rumus untuk mencari *gain*.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Entropy(S_i) \quad (3)$$

Di mana :

$S$  = himpunan kasus  
 $A$  = atribut  
 $n$  = jumlah partisi  $S$   
 $|S_i|$  = jumlah kasus pada partisi ke-1

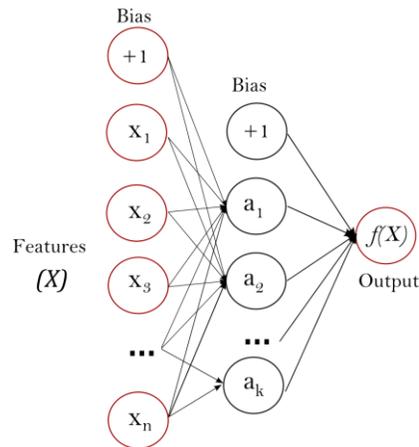
$|S|$  = jumlah kasus dalam  $S$

Setelah semua atribut dihitung dengan rumus pencarian *gain*, maka atribut yang memiliki nilai *gain* tertinggi akan dijadikan *node* (akar). *Node* memiliki *instance*, sehingga *instance* dijadikan cabang dari *node*. Nilai *instance* kemudian diklasifikasikan lagi sehingga menjadi lebih sederhana. Ulangi proses tersebut untuk setiap cabangnya sehingga membentuk sebuah pohon keputusan.

## 2.4 Multilayer Perceptron

*Multilayer perceptron* adalah algoritma *supervised learning* yang mempelajari fungsi dengan melatih pada dataset. Pada algoritma *multilayer perceptron* digunakan fungsi standar *sigmoid* di mana jumlah pembobotan dari sejumlah input dan bias dimasukkan ke *activation level* melalui fungsi transfer untuk menghasilkan *output*, yang selanjutnya unit-unit akan diatur dalam lapisan topologi *feed-forward* yang disebut *feed-forward neural network* (FFNN) [7].

Ketika ada lebih dari satu lapisan yang tersembunyi (*hidden layer*), maka *output* dari *hidden layer* diteruskan ke *hidden layer* berikutnya dan bobot terpisah digunakan untuk penjumlahan ke setiap lapisan berikutnya. *Multilayer perceptron* terdiri dari *input layer*, *hidden layer*, dan *output layer*. Contoh sederhana arsitektur *multilayer perceptron* dapat dilihat pada Gambar 1 berikut.



**Gambar 1.** *Multilayer perceptron* dengan 1 *hidden layer* dan 1 *node output*

Arsitektur sederhana *multilayer perceptron* pada Gambar 1 di atas adalah pada lapisan awal (yaitu lapisan input) terdapat  $n$  jumlah fitur (dinotasikan dengan  $x$ , dari  $x_1$  sampai  $x_n$ ), yang kemudian pada *hidden layer* adalah nilai aktivasi dari hasil penjumlahan dari nilai input fitur dikalikan dengan bobot, dan pada lapisan akhir terdapat satu nilai luaran yang berasal dari nilai aktivasi dari hasil penjumlahan dari nilai *node* pada *hidden layer* dikalikan dengan bobot.

## 2.5 Gaussian Naive Bayes

*Gaussian naive Bayes* merupakan salah satu varian dari algoritma klasifikasi *naive Bayes* dimana setiap kelas data diasumsikan terdistribusi menggunakan *normal distribution* atau *Gaussian distribution*. Cara kerja klasifikasi pada *Gaussian naive Bayes* sama seperti *naive Bayes* dimana klasifikasi didasarkan pada teorema Bayes namun pada perhitungan nilai *likelihood*-nya digunakan rumus *Gaussian distribution*. Berikut merupakan persamaan yang diperlukan pada proses klasifikasi *Gaussian naive Bayes*.

Persamaan pada metode *naive Bayes* adalah sebagai berikut :

$$p(c | X) = \frac{p(X|c)p(c)}{p(x)} \quad (4)$$

Keterangan rumus di atas adalah sebagai berikut :

- $x$  = data dengan kelas yang belum diketahui  
 $c$  = hipotesis data merupakan suatu class spesifik  
 $p(c|X)$  = probabilitas berdasarkan kondisi (*Posterior Probability*)  
 $p(x|c)$  = probabilitas berdasarkan kondisi pada hipotesis (*likelihood*)  
 $p(c)$  = probabilitas hipotesis (*Prior Probability*)  
 $p(x)$  = probabilitas X

Berikut ini adalah rumus pencarian nilai *likelihood* menggunakan *Gaussian distribution* :

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right). \quad (5)$$

Di mana :

- $x_i$  = data ke i  
 $y$  = kelas yang dicari  
 $\mu_y$  = rata-rata dari seluruh data  
 $\sigma_y^2$  = varian dari seluruh data

## 2.6 Random Forest

*Random forest* merupakan sebuah algoritma klasifikasi yang bekerja secara *ensemble* atau kombinasi dari beberapa algoritma klasifikasi lainnya. *Random forest* didasari oleh algoritma klasifikasi *decision tree*. Pada kasus klasifikasi, pengambilan keputusan pada *random forest* bekerja dengan cara mengambil suara terbanyak yang dihasilkan dari sekumpulan *decision tree* [8]. Pada kasus regresi, pengambilan keputusan pada *random forest* bekerja dengan cara mengambil nilai rata-rata dari hasil keputusan sekumpulan *decision tree* [8]. Perhitungan pengambilan keputusan pada *random forest* dapat dilakukan dengan menggunakan rumus *bagging* atau *bootstrap aggregating* yang dijabarkan sebagai berikut.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x'). \quad (6)$$

Di mana :

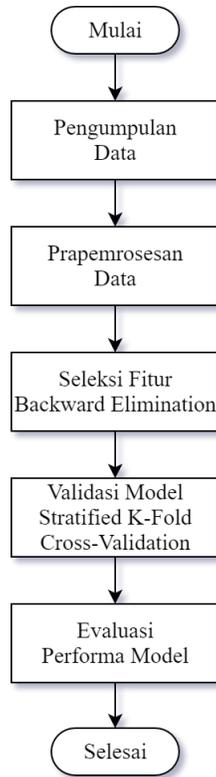
- $\hat{f}$  = nilai keputusan  
 $B$  = jumlah perulangan atau banyaknya *tree*  
 $f_b$  = nilai keputusan yang didapatkan dari *decision tree*  
 $x'$  = sampel data

## 2.7 Stratified K-Fold Cross-Validation

*Stratified k-fold cross-validation* digunakan sebagai metode validasi model. *Stratified k-fold cross-validation* bekerja dengan cara membagi data latih dan validasi menjadi k bagian dengan pembagian kelas yang sama (*stratified*), yang artinya data dibagi menurut strata atau jumlah kelas positif dan negatif (apabila kelasnya biner). Selain *stratified* adalah pembagian data berdasarkan kelas, *k-fold cross-validation* bekerja dalam k iterasi, yaitu 1 bagian akan dijadikan data validasi dan k-1 sisanya sebagai data latih yang dan setiap bagian akan mendapatkan giliran menjadi data testing.

### 3. Metodologi Penelitian

Alur penelitian yang digunakan dalam penelitian dimulai pada pengumpulan data, prapemrosesan data, seleksi fitur, validasi model, dan evaluasi performa model, sebagaimana dapat dilihat pada Gambar 2 berikut.



**Gambar 2.** Diagram Alur Penelitian

#### 3.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah data yang digunakan oleh penelitian Yöntem *et al.* [1] yang dapat diakses secara publik di UCI Machine Learning Repository [9]. Partisipan pada data penelitian tersebut berasal dari tujuh daerah yang berbeda di Turki, namun sebagian besar berasal dari daerah Laut Hitam (n=79). Terdapat dua instrumen pengumpulan data, yaitu dengan form informasi pribadi dan dengan *Divorce Predictors Scale* (DPS). Namun pada dataset yang dipublikasi hanya data dari DPS saja dan tidak mempublikasikan informasi pribadi partisipan. DPS yang dikembangkan di penelitian Yöntem *et al.* sebelumnya [2] terdiri dari 54 soal atau pertanyaan yang masing-masingnya berupa nilai dengan 5 subskala. Partisipan yang bercerai menjawab soal skala dengan mempertimbangkan pernikahan mereka. Dan, dari partisipan yang sudah menikah, hanya mereka yang memiliki pernikahan bahagia, tanpa ada pemikiran untuk bercerai, yang diikutsertakan dalam penelitian ini. Dari 170 pasangan partisipan, 86 (51%) di antaranya menikah, dan 84 (49%) lainnya telah bercerai. Dari 170 baris data dengan masing-masingnya terdapat 54 kolom fitur (skala 0-4, bilangan bulat) dan 1 kolom kelas (Boolean), tidak terdapat *missing value*, namun ada beberapa baris data yang duplikat.

### 3.2 Prapemrosesan Data

Karena terdapat baris data yang duplikat, maka dilakukan penghapusan baris data sehingga masing-masing entri adalah unik satu dengan lainnya. Selain menghapus data yang duplikat, dilakukan normalisasi fitur dengan *min-max*, dari yang sebelumnya bernilai 0-4 bilangan bulat menjadi 0-1 bilangan riil.

### 3.3 Seleksi Fitur Backward Elimination

Pada tahap seleksi fitur digunakan *backward elimination* untuk mereduksi dimensi fitur yang digunakan dengan harapan menjaga tingkat akurasi atau bahkan meningkatkan akurasi. Untuk pembagian data latih dan uji yang diperlukan khusus untuk *backward elimination*, digunakan pembagian *hold-out* pada dataset dengan 80% untuk data latih dan 20% untuk data uji. Tahap seleksi fitur ini diterapkan ke setiap algoritma klasifikasi, sehingga hasil fitur yang telah diseleksi bisa jadi berbeda bagi masing-masing algoritma klasifikasi meskipun pada hasil evaluasi yang sama (dengan pembagian *hold-out*). Algoritma klasifikasi yang digunakan adalah *decision tree*, *Gaussian naive Bayes*, *random forest*, serta *multilayer perceptron*.

### 3.4 Validasi Model Stratified K-Fold Cross-Validation

Untuk memvalidasi fitur yang telah diseleksi pada tahap sebelumnya yang menggunakan pembagian *hold-out*, diperlukan validasi yang lebih dapat dipercaya agar hasil performa yang dihasilkan bukan kebetulan karena hanya menggunakan bagian data tertentu saja. *Stratified k-fold cross-validation* digunakan sebagai pembagian data latih dan uji, agar performa model yang dihasilkan dapat langsung disimpulkan apakah model tersebut *robust* atau tidaknya. Masing-masing algoritma klasifikasi diterapkan *stratified k-fold cross-validation* dengan fitur yang telah diseleksi pada tahap sebelumnya untuk masing-masing algoritma. Pada penelitian ini digunakan  $k=10$ .

### 3.5 Evaluasi Performa Model

Setelah model dilatih dan divalidasi dengan pembagian *stratified k-fold cross-validation*, didapatkan nilai performanya pada setiap iterasi *cross-validation* untuk setiap algoritma klasifikasi. Evaluasi performa yang digunakan di antaranya adalah akurasi, presisi, dan *f-score*. Perhitungan akurasi, presisi, dan *f-score* dapat dilihat pada rumus berikut berdasarkan matriks konfusi pada Tabel 1 di bawah.

**Tabel 1.** Matriks konfusi

Kelas	Prediksi Positif	Prediksi Negatif
Aktual Positif	<i>TRUE Positive</i> (TP)	<i>FALSE Negative</i> (FN)
Aktual Negatif	<i>FALSE Positive</i> (FP)	<i>TRUE Negative</i> (TN)

Dari Tabel 1 tersebut, dapat diturunkan beberapa rumus di antaranya perhitungan nilai akurasi, presisi, dan *f-score* :

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN) . \quad (7)$$

$$\text{Presisi} = TP / (TP + FP) . \quad (8)$$

$$F1\text{-score} = 2 * (((TP / (TP + FP)) * (TP / (TP + FN)))) / (((TP / (TP + FP)) + (TP / (TP + FN)))) . \quad (9)$$

Berdasarkan hasil evaluasi model yang telah didapatkan, segala bentuk pengambilan keputusan yang dihasilkan oleh model dapat dijadikan referensi seperti untuk pendukung keputusan prediksi akan terjadinya perceraian atau tidak.

#### 4. Hasil dan Pembahasan

Pada penelitian Yöntem *et al.* pada [1], terdapat 6 fitur yang telah terpilih menggunakan *correlation-based feature selection* (CBFS) seperti pada Tabel 2 berikut.

**Tabel 2.** Fitur terpilih dengan CBFS pada [1]

Fitur	Deskripsi	Nilai signifikansi
x2	"I know we can ignore our differences even if things get though sometimes."	0.90
x6	"We don't have a common time we spent together at home."	0.89
x11	"When I look back in the future, I think that my spouse and I had our paths harmoniously intertwined."	0.87
x18	"We have similar ideas with my spouse about how a marriage should be."	0.84
x26	"I know the basic concerns of my spouse."	0.84
x40	"We're starting to fight before I know what's going on."	0.83

Keenam fitur pada Tabel 2 tersebut digunakan di penelitiannya [1] untuk membuat model prediksi perceraian dengan algoritma klasifikasi ANN, RBF *neural network*, dan *random forest*, yang hasilnya dapat dilihat pada Tabel 3 berikut.

**Tabel 3.** Evaluasi performa model Yöntem *et al.* dengan dan tanpa CBFS pada [1]

Feature Selection	Classification	Number of Feature	Accuracy (%)
None	ANN	54	97.64
	RBF		98.23
	Random Forest		97.64
Correlation based feature selection	ANN	6	<b>98.82</b>
	RBF		97.64
	Random Forest		97.64

Pada Tabel 3 tersebut dapat dilihat bahwa model terbaik Yöntem *et al.* pada [1] adalah ANN yang menggunakan CBFS dengan akurasi 98.82%. Meskipun demikian, penelitiannya tidak mendefinisikan konfigurasi parameter yang digunakan untuk ketiga model tersebut, namun hanya menyebutkan bahwa metode-metode yang digunakan memanfaatkan perangkat lunak WEKA (Waikato Environment for Knowledge Analysis).

Sebagai penelitian yang merujuk pada penelitian tersebut dan menggunakan data yang sama, maka pada penelitian ini dilakukan beberapa perbaikan atau pengembangan yang tidak dilakukan pada penelitian tersebut seperti dengan melakukan pembuangan data duplikat, normalisasi data, seleksi fitur dengan *backward elimination*, dan pembagian data yang digunakan.

Pada penelitian ini, seluruh algoritma klasifikasi yang digunakan diterapkan dengan menggunakan pustaka scikit-learn yang dapat digunakan di bahasa pemrograman Python [10][11]. Selain itu, konfigurasi parameter algoritma klasifikasi

yang digunakan adalah *default* kecuali pada algoritma *multilayer perceptron*. Algoritma klasifikasi multilayer perceptron diatur dengan menentukan jumlah *hidden layer*, yaitu yang *default*-nya berupa 1 *hidden layer* dengan 100 *node*, menjadi 2 *hidden layer* yang masing-masing *layer*-nya secara berurutan berisi 5 dan 2 *node*. Referensi detail terkait konfigurasi algoritma klasifikasi yang digunakan pada *decision tree* [12], *Gaussian naive Bayes* [13], *random forest* [14], dan *multilayer perceptron* [15].

*Backward elimination* diterapkan untuk masing-masing algoritma klasifikasi, sehingga fitur yang dihasilkan adalah fitur-fitur yang memberikan performa terbaik untuk masing-masing algoritma klasifikasi. Untuk mendapatkan fitur-fitur tersebut, algoritma dijalankan beberapa kali untuk dilihat seberapa sering fitur tersebut muncul sebagai prediktor yang memberikan performa terbaik bagi algoritma. Agar dapat dibandingkan dengan penelitian Yöntem *et al.* [1], diambil 6 fitur pada masing-masing algoritma klasifikasi seperti pada Tabel 4 berikut.

Pembagian data latih dan uji dilakukan dengan *stratified k-fold cross validation*. Pada setiap algoritma, didapatkan hasil 6 fitur terpilih yang berbeda setiap kali program dijalankan. Untuk menentukannya, hanya fitur-fitur yang paling sering muncul yang digunakan, dan Tabel 4 berikut adalah fitur-fitur yang terpilih.

**Tabel 4.** Fitur terpilih pada masing-masing algoritma klasifikasi

Algoritma Klasifikasi	Fitur
Decision Tree	x26, x44, x50, x52, x53, x54
Random Forest	x41, x50, x51, x52, x53, x54
Gaussian Naive Bayes	x40, x50, x51, x52, x53, x54
Multi-Layer Perceptron	x28, x45, x50, x52, x53, x54

Dari Tabel 4 tersebut, deskripsi masing-masing fitur dapat dilihat pada Tabel 5 berikut.

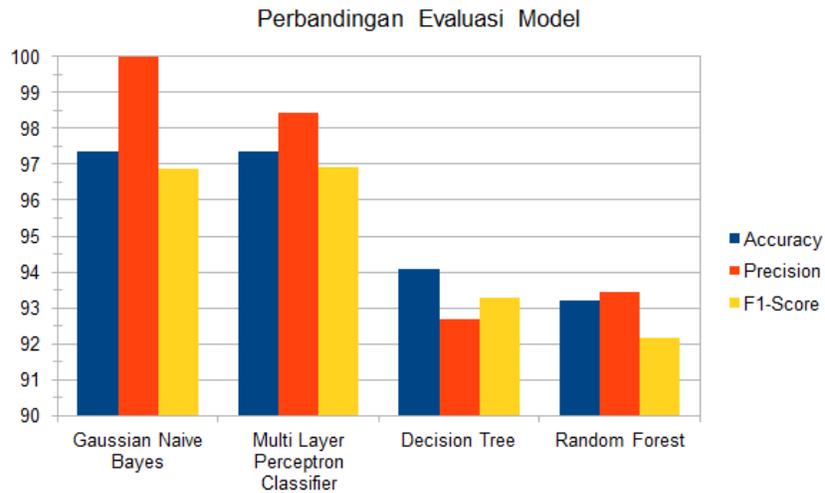
**Tabel 5.** Deskripsi masing-masing fitur pada Tabel 4

Fitur	Deskripsi
x26	"I know my spouse's basic anxieties."
x28	"I know my spouse's hopes and wishes."
x40	"We're just starting a discussion before I know what's going on."
x41	"When I talk to my spouse about something, my calm suddenly breaks."
x44	"Sometimes I think it's good for me to leave home for a while."
x45	"I'd rather stay silent than discuss with my spouse."
x50	"I'm not actually the one who's guilty about what I'm accused of."
x51	"I'm not the one who's wrong about problems at home."
x52	"I wouldn't hesitate to tell my spouse about her/his inadequacy."

x53	"When I discuss, I remind my spouse of her/his inadequacy."
x54	"I'm not afraid to tell my spouse about her/his incompetence."

Setelah mendapatkan masing-masing 6 fitur di Tabel 4 tersebut, dilakukan pengujian model sebanyak 10 kali untuk mendapatkan rata-rata hasil akurasi, presisi, dan *f-score* pada tiap model dengan *stratified k-fold cross-validation*, yang artinya program dijalankan total 100 kali (10 iterasi untuk setiap *stratified k-fold cross-validation*).

Perbandingan evaluasi model yang digunakan pada penelitian ini (menggunakan BFS, *backward feature selection*) dapat dilihat pada Gambar 3 dan Tabel 6 di bawah. Sedangkan perbandingan hasil akurasi pada penelitian ini dengan *correlation-based feature selection* (CBFS) milik Yöntem *et al.* [1] dapat dilihat pada Gambar 4 dan Tabel 7.

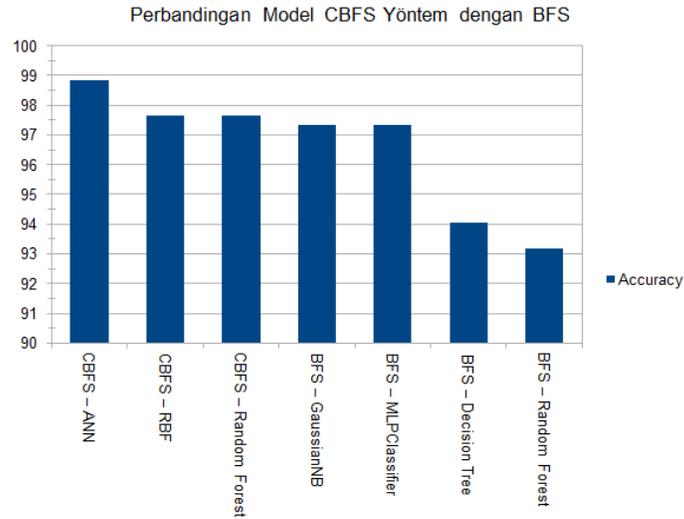


**Gambar 3.** Grafik perbandingan hasil evaluasi tiap model menggunakan BFS

**Tabel 6.** Tabel rincian evaluasi tiap model menggunakan BFS

Algoritma Klasifikasi	Akurasi (%)	Precision (%)	F1-Score (%)
Gaussian Naive Bayes	<b>97.34</b>	<b>100</b>	96.88
Multilayer Perceptron	<b>97.34</b>	98.43	<b>96.92</b>
Decision Tree	94.07	92.678	93.3
Random Forest	93.20	93.463	92.162

Pada Gambar 3 adalah perbandingan evaluasi model BFS dalam penelitian ini secara grafis, sedangkan pada Tabel 6 adalah detail evaluasinya. Untuk perbandingan evaluasi model BFS dalam penelitian ini dengan evaluasi model CBFS pada [1] dapat dilihat pada Gambar 4 dan Tabel 7 berikut.



**Gambar 4.** Perbandingan akurasi model BFS dalam penelitian ini dengan model CBFS Yöntem *et al.*

**Tabel 7.** Tabel rincian perbandingan akurasi model CBFS Yöntem dengan model BFS

Algoritma Klasifikasi	Akurasi (%)
CBFS - ANN	<b>98.82</b>
CBFS - RBF	97.64
CBFS - Random Forest	97.64
BFS - GaussianNB	97.34
BFS - MLPClassifier	97.34
BFS - Decision Tree	94.07
BFS - Random Forest	93.20

Gambar 4 adalah perbandingan model CBFS pada [1] dengan model BFS pada penelitian ini, dan Tabel 7 adalah detail dari evaluasi performa yang sama-sama digunakan, yaitu akurasi, yang diurutkan dari nilai evaluasi tertinggi.

Berdasarkan perbandingan pada grafik dan tabel tersebut, terlihat bahwa hasil yang didapatkan menggunakan *backward feature selection* (BFS) sedikit lebih rendah dari hasil yang didapatkan dengan menggunakan *correlation based feature selection* (CBFS), namun jarak antara perbedaan akurasi tersebut hanya berkisar antara 1% - 5%. Hasil akurasi yang paling mendekati dan dapat dijadikan alternatif pemodelan yang dilakukan oleh Yöntem *et al.* adalah hasil pemodelan *Gaussian naive Bayes* dan *multilayer perceptron*, yang keduanya dengan BFS yang memiliki akurasi sebesar 97.34%, untuk presisi tertinggi dimiliki oleh *Gaussian naive Bayes* sebesar 100%, sementara *f-score* sedikit lebih baik *multilayer perceptron* sebesar 96.92% dibanding *Gaussian naive Bayes* yang sebesar 96.88%.

## 5. Kesimpulan

Dari keempat model yang telah divalidasi dengan *stratified k-fold cross-validation*, model *Gaussian naive Bayes* menghasilkan performa tertinggi dengan akurasi 97.34%, presisi, 100%, dan *f-score* 96.87%. Meskipun demikian,

dibandingkan dengan performa terbaik yang dicapai oleh model Yöntem *et al.* yaitu algoritma ANN dengan akurasi 98.82%, *Gaussian naive Bayes* pada penelitian ini hanya lebih buruk 1.48%. Artinya tahapan pra-proses, seleksi fitur, dan algoritma klasifikasi *Gaussian naive Bayes* dapat dipertimbangkan untuk dieksplorasi dan dijadikan model alternatif.

## 6. Saran

Saran yang dapat diberikan untuk penelitian lebih lanjut yaitu:

- a. Mereduksi dimensi dengan metode ekstraksi fitur (dibanding dengan metode seleksi fitur) seperti dengan *principal component analysis* (PCA).
- b. Prapemrosesan menggunakan normalisasi lainnya (dibanding dengan metode *min-max*) seperti dengan *z-score*.
- c. Memperbanyak sampel dengan teknik *oversampling* dan bagaimana pengaruhnya.
- d. Menggunakan algoritma klasifikasi lainnya, atau bahkan kombinasinya (*ensemble*).

## Referensi

- [1] Yöntem, M.K., Adem, K., İlhan, T. and Kılıçarslan, S., 2019. Divorce prediction using correlation based feature selection and artificial neural networks. *Neşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi*, 9(1), pp.259-273.
- [2] Yöntem, M.K. and İlhan, T. (2018). Boşanma Göstergeleri Ölçeğinin Geliştirilmesi. [Development of the Divorce Predictors Scale]. *Sosyal Polika Çalışmaları Dergisi*. 41, 339-358.
- [3] Gottman, J. M. (2014). *What Predicts Divorce? The Relationship Between Marital Processes and Marital Outcomes*. New York: Psychology Press.
- [4] Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., & Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science and Technology*, 26(1), 329–340.
- [5] Larose, D.T. and Larose, C.D., 2014. *Discovering Knowledge in Data: An Introduction to Data Mining*. 2nd ed.
- [6] Oktafianto. (2016). 'Analisis Kepuasan Mahasiswa Terhadap Pelayanan Akademik Menggunakan Metode Algoritma C4.5 (Studi Kasus: Stmik Pringsewu)', *jurnal: TIM Darmajaya Vol. 02 No. 01 Mei 2016, ISSN: 2442-5567, E-ISSN: 2443-289X*.
- [7] Setiadi, A. (2012). Penerapan Algoritma Multilayer Perceptron Untuk Deteksi Dini Penyakit Diabetes. *Paradigma-Jurnal Komputer Dan Informatika*, 14(1), 46–59.
- [8] Marsland, S., 2014. *Machine Learning: An Algorithmic Perspective*. 2nd ed.
- [9] Yöntem, M.K., Adem, K., İlhan, T. and Kılıçarslan, S., 2019. UCI Machine Learning Repository: Divorce Predictors data set Data Set. [online] Archive.ics.uci.edu. Available at: <https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set> [Accessed 30 March 2021].
- [10] Pedregosa *et al.*, 2011. Scikit-learn: Machine Learning in Python. *JMLR* 12, pp. 2825-2830.
- [11] Buitinck *et al.*, 2013. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. pp. 108–122.
- [12] Scikit-learn.org. n.d. `sklearn.tree.DecisionTreeClassifier` — scikit-learn 0.24.1 documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> [Accessed 1 April 2021].
- [13] Scikit-learn.org. n.d. `sklearn.naive_bayes.GaussianNB` — scikit-learn 0.24.1 documentation. [online] Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.naive\\_bayes.GaussianNB.html](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html) [Accessed 1 April 2021].
- [14] Scikit-learn.org. n.d. `sklearn.ensemble.RandomForestClassifier` — scikit-learn 0.24.1 documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> [Accessed 1 April 2021].
- [15] Scikit-learn.org. n.d. `sklearn.neural_network.MLPClassifier` — scikit-learn 0.24.1 documentation. [online] Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html) [Accessed 1 April 2021].