



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Machine Learning

Kombinasi *K-Nearest Neighbor* (KNN) dan *Relief-F* Untuk Meningkatkan Akurasi Pada Klasifikasi Data

Rahmad Nurhadi Yusra ¹, Opim Salim Sitompul ², Sawaluddin ³

^{1,2}Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Medan, Indonesia

³Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan, Indonesia

KEYWORDS

Klasifikasi, K-Nearest Neighbor, *Relief-F*, Seleksi Fitur, Peningkatan Akurasi

CORRESPONDENCE

Phone: 081360331775

E-mail: hadi2106yusra@gmail.com

ABSTRACT

Dalam penelitian ini, penulis mengusulkan proses peningkatan akurasi pada *K-Nearest Neighbor* (KNN) dengan kombinasi seleksi fitur menggunakan metode *Relief-F*. Adapun penyebab kurang maksimalnya akurasi pada *K-Nearest Neighbor* dibandingkan dengan metode klasifikasi lainnya disebabkan oleh pengaruh atribut yang kurang signifikan dan persentase pengaruh yang cenderung rendah dari suatu data dalam menentukan kelas pada data baru. Metode *Relief-F* digunakan untuk melakukan seleksi pada atribut yang korelasinya kurang baik dari data yang diujikan. Pengujian dari metode yang diusulkan yaitu membandingkan akurasi yang diperoleh dari metode KNN tanpa menggunakan seleksi fitur dengan KNN menggunakan seleksi fitur *Relief-F*. Hasil pengujian yang diperoleh yaitu metode yang diusulkan mampu meningkatkan akurasi klasifikasi dari KNN dengan peningkatan yang diperoleh yaitu sebesar 10.32% setelah dibandingkan dengan pengujian KNN tanpa seleksi fitur.

PENDAHULUAN

Dalam kajian *Data Mining* ataupun juga *Machine Learning*, suatu teknik yang digunakan dalam memilih ataupun menghilangkan fitur yang dianggap kurang penting dengan menggunakan kriteria tertentu dan termasuk juga salah satu dari tahapan *pre-processing* pada data, dan teknik tersebut dikenal dengan sebutan Seleksi Fitur (*Feature Selection*) [1].

Feature Selection merupakan cara yang efektif untuk melakukan reduksi data dan menjadi langkah penting yang perlu dilakukan agar dapat memperoleh kinerja dan hasil yang baik pada *data mining* [2].

Salah satu metode yang sering digunakan pada *Data Mining* ataupun *Machine Learning* yaitu *K-Nearest Neighbor* (KNN) yang merupakan metode klasifikasi yang paling sederhana dan paling sering digunakan untuk klasifikasi dengan menggunakan pendekatan berdasarkan jarak tetangga terdekat dari data pembelajaran ke data pengujian dalam menetapkan sebuah kelas pada data berdasarkan suara terbanyak [3].

Dari beberapa penelitian yang telah ada, akurasi yang diperoleh KNN kurang maksimal dibandingkan dari metode klasifikasi

lainnya. Seperti pada penelitian dari [4], yang memperoleh perbandingan akurasi antara SVM dan KNN dengan akurasi SVM sebesar 82.54% dan KNN sebesar 79,22%.

Kemudian pada penelitian [5], hasil perbandingan antara *Naive Bayes* yang memiliki akurasi terbaik sebesar 73,7%, *Decision Tree* yaitu 58,9% dan KNN sebesar 56,7%.

Dari penjabaran beberapa penelitian tersebut dapat ditarik kesimpulan bahwa akurasi yang diperoleh KNN cenderung lebih rendah yang disebabkan pengaruh atribut yang digunakan terhadap proses klasifikasi.

Pada penelitian [6] mengusulkan dengan melakukan seleksi fitur untuk membuang atribut-atribut yang kurang relevan sebelum data diklasifikasikan dengan K-NN. Dan berdasarkan hasil penelitian dari penelitian tersebut seleksi fitur dapat menghasilkan akurasi yang lebih besar pada KNN setelah dilakukannya seleksi fitur.

Pada penelitian [7], yang melakukan penelitian dalam meningkatkan akurasi untuk memprediksi jenis kelamin (*Gender*) dengan seleksi fitur *Relief-F* dengan hasil yang diperoleh yaitu akurasi sebesar 93.35 % setelah menghapus fitur yang kurang relevan pada data yang diujikan.

Kemudian pada penelitian [8] yang melakukan kombinasi SVM dengan *Relief-F* dalam klasifikasi data yang menghasilkan peningkatan akurasi setelah SVM dikombinasikan dengan *Relief-F* yaitu peningkatan akurasi sebesar 24.37 %.

Maka pada penelitian ini penulis mengusulkan untuk mengkombinasikan metode KNN dengan seleksi fitur *Relief-F* dengan harapan dapat meningkatkan perolehan akurasi dari proses klasifikasi data pada metode KNN dan dapat memaksimalkan proses penentuan kelas pada suatu data.

TINJAUAN PUSTAKA

Penelitian Terdahulu

Penelitian dari [4], membandingkan akurasi antara *Support Vector Machine* (SVM) dan KNN dan terlihat bahwa kinerja SVM lebih baik daripada KNN, dengan akurasi SVM sebesar 82.54% dan KNN sebesar 79,22%.

Penelitian [5], membandingkan performa dari *Naïve Bayes*, *Decision Tree* dan *K-Nearest Neighbor* (KNN). *Naïve Bayes* memiliki akurasi terbaik dengan akurasi rata-rata 73,7%, *Decision Tree* yaitu 58,9% dan KNN sebesar 56,7%.

Penelitian [7], peningkatan akurasi dalam memprediksi jenis kelamin (*Gender*) dengan seleksi fitur menggunakan *Relief-F* dengan hasil yang diperoleh menggunakan *Relief-F* yaitu sebesar 93.35 % setelah menghapus fitur yang kurang relevan dari kumpulan fitur yang ada pada data yang diujikan.

Penelitian [8] yang membahas tentang kombinasi metode *Support Vector Machine* (SVM) dengan *Relief-F* dalam proses klasifikasi. Dan terjadinya peningkatan akurasi setelah *Support Vector Machine* (SVM) dikombinasikan dengan *Relief-F* yaitu sebesar 24.37 %.

Klasifikasi

Klasifikasi umumnya dipergunakan untuk memprediksi atau mengelompokkan suatu kelas dari suatu data berdasarkan atribut. Adapun *discription* dari setiap kelas data telah ditetapkan [9].

K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (KNN) merupakan model klasifikasi yang paling sederhana dengan menggunakan pendekatan dari k tetangga terdekat dan menetapkan sebuah kelas berdasarkan suara terbanyak [10].

Langkah-langkah dalam proses klasifikasi pada *K-Nearest Neighbor* (KNN) sebagai berikut:

1. Menentukan parameter k (jumlah tetangga paling dekat).
2. Menghitung kedekatan berdasarkan model jarak *Euclidean* terhadap data latih yang diberikan, dengan persamaan (1) berikut.

$$D(x, y) = \|x - y\|_2 = \sqrt{\sum_{j=1}^N |x_j - y_j|^2} \quad (1)$$

3. Mengurutkan hasil jarak yang didapatkan secara *ascending* (berurutan dari nilai tertinggi ke terendah).
4. Menghitung jumlah setiap kelas berdasarkan k tetangga terdekat.

5. Kemudian, kelas dengan nilai terbesar dijadikan sebagai kelas untuk data uji.

Seleksi Fitur

Proses yang dilakukan untuk menentukan fitur-fitur yang signifikan dalam dataset yang sesuai untuk permasalahan yang akan dipecahkan disebut dengan Seleksi Fitur. Jika semakin baik fitur yang dipilih, maka dapat meningkatkan nilai *accuracy* dari metode yang diuji. Seleksi fitur juga bermanfaat dalam mereduksi dimensi dari dataset dengan cara membuang fitur-fitur yang tidak memiliki pengaruh terhadap penentuan kelas/label [1].

Relief-F

Relief-F merupakan metode untuk seleksi fitur yang dikembangkan oleh Kononenko pada tahun 1994 dan merupakan pengembangan dari *Relief*. *Relief-F* memiliki efisiensi tinggi dan tidak membatasi karakteristik tipe data. Langkah-langkah untuk mencari nilai bobot fitur dengan *Relief-F* yaitu sebagai berikut [11].

1. Inisialisasi nilai awal seluruh bobot fitur = 0 dan menentukan jumlah iterasi.
2. Memilih sebuah data yang akan dijadikan sebagai titik acak atau titik pusat.
3. Mencari *miss* dan *hit* terdekat dengan cara menghitung jarak antara titik pusat dengan data yang memiliki kelas yang sama. Jarak terdekat antara titik pusat dan data pada kelas positif disebut *hit*. Sedangkan, jarak terdekat antara titik pusat dengan data yang pada kelas negatif disebut *miss*.
4. Melakukan *update* bobot untuk setiap fitur. Fitur dengan data kategori dihitung menggunakan Persamaan (2) berikut.

$$diff(A, Ri, HM) = \begin{cases} 0; & \text{value}(A, Ri) = \text{value}(A, H, M) \\ 1; & \text{otherwise} \end{cases} \quad (2)$$

5. Sedangkan, fitur dengan data numerik dihitung menggunakan persamaan (3) berikut.

$$diff(A, Ri, HM) = \frac{|value(A, Ri) - value(A, H, M)|}{\max(A) - \min(A)} \quad (3)$$

6. Sehingga rumus perbaruan bobot dihitung menggunakan persamaan (4) berikut.

$$W[A] = W[A] - \frac{diff(A, Ri, H)}{m} + \frac{diff(A, Ri, M)}{m} \quad (4)$$

7. Selanjutnya, dilanjutkan dengan iterasi selanjutnya yang dimulai dari langkah 1 hingga bobot fitur yang baru telah didapat.

Fold Cross Validation

Fold Cross Validation digunakan untuk mengevaluasi dan membandingkan algoritma pembelajaran dengan membagi data menjadi dua segmen yaitu model data pembelajaran (*training*) dan model data pengujian (*testing*) [12]. Tujuan utama dari *fold cross validation* yaitu untuk mengukur kinerja pada model pembelajaran dari data yang tersedia dalam satu metode. Selain itu dapat juga untuk membandingkan kinerja dari dua atau lebih variasi model parameter.

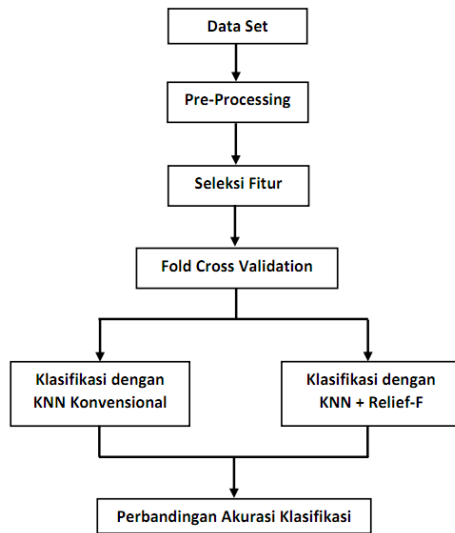
Pada penelitian ini menggunakan metode evaluasi *10-fold cross validation* dengan membentuk 10 *subsets* dari *data set* yang digunakan. 9 *subsets* digunakan sebagai *training* dan 1

subset digunakan sebagai testing. Hasil pengukuran akhir yaitu rata-rata dari 10 kali pengujian.

METODOLOGI PENELITIAN

Tahapan Penelitian

Adapun untuk menguji dari metode yang diusulkan pada penelitian ini dilakukan dengan tahapan penelitian pada Gambar 1 berikut.



Gambar 1. Tahapan Penelitian

Adapun keterangan tahapan penelitian pada Gambar 1 menggambarkan langkah-langkah yang diterapkan pada penelitian ini yaitu sebagai berikut:

Dataset

Adapun data yang digunakan pada penelitian yaitu *User Knowledge Modeling* yang merupakan data set berisi tentang status pengetahuan pelajar tentang subjek *Electrical DC Machines*. Adapun Data set *User Knowledge Modeling* diperoleh dari *UCI Machine Learning Repository*. Karakteristik dari data set tersebut yaitu Multivariate, dengan karakteristik atribut yaitu Integer. Jumlah data pada data set tersebut yaitu 403 data dengan banyak atribut sebanyak 5 atribut dan 1 kelas atribut yang terdiri dari 4 kelas yaitu *Very Low*, *Low*, *Middle*, dan *High*.

Adapun rincian informasi atribut yang berhubungan pada *User Knowledge Modeling Dataset* yaitu pada Tabel 1 berikut.

Tabel 1. Atribut *User Knowledge Modeling Dataset*

No.	Atribut	Nilai	Penanda
1.	STG	0 – 0.99	X1
2.	SCG	0 – 0.9	X2
3.	STR	0 – 0.95	X3
4.	LPR	0 – 0.99	X4
5.	PEG	0 – 0.99	X5
6.	UNS (Class)	Very Low, Low, Middle, High	-

Adapun informasi rincian dari data yang digunakan yaitu pada Tabel 2 berikut:

Tabel 2. Rincian Data *User Knowledge Modeling*

No.	X1	X2	X3	X4	X5	Class
1	0	0	0	0	0	very_low
2	0.08	0.08	0.1	0.24	0.9	High
3	0.06	0.06	0.05	0.25	0.33	Low
4	0.1	0.1	0.15	0.65	0.3	Middle
5	0.08	0.08	0.08	0.98	0.24	Low
6	0.09	0.15	0.4	0.1	0.66	Middle
7	0.1	0.1	0.43	0.29	0.56	Middle
8	0.15	0.02	0.34	0.4	0.01	very_low
9	0.2	0.14	0.35	0.72	0.25	Low
10	0	0	0.5	0.2	0.85	High
11	0.18	0.18	0.55	0.3	0.81	High
12	0.06	0.06	0.51	0.41	0.3	Low
13	0.1	0.1	0.52	0.78	0.34	Middle
14	0.1	0.1	0.7	0.15	0.9	High
15	0.2	0.2	0.7	0.3	0.6	Middle
16	0.12	0.12	0.75	0.35	0.8	High
17	0.05	0.07	0.7	0.01	0.05	very_low
18	0.1	0.25	0.1	0.08	0.33	Low
19	0.15	0.32	0.05	0.27	0.29	Low
20	0.2	0.29	0.25	0.49	0.56	Middle
21	0.12	0.28	0.2	0.78	0.2	Low
22	0.18	0.3	0.37	0.12	0.66	Middle
23	0.1	0.27	0.31	0.29	0.65	Middle
24	0.18	0.31	0.32	0.42	0.28	Low
25	0.06	0.29	0.35	0.76	0.25	Low
26	0.09	0.3	0.68	0.18	0.85	High
27	0.04	0.28	0.55	0.25	0.1	very_low
28	0.09	0.255	0.6	0.45	0.25	Low
29	0.08	0.325	0.62	0.94	0.56	High
30	0.15	0.275	0.8	0.21	0.81	High
...
391	0.78	0.47	0.29	0.98	0.59	High
392	0.58	0.4	0.32	0.22	0.24	Low
393	0.68	0.43	0.6	0.47	0.55	Middle
394	0.57	0.37	0.75	0.27	0.32	Low
395	0.62	0.56	0.11	0.24	0.22	Low
396	0.64	0.58	0.14	0.32	0.21	Low
397	0.64	0.59	0.12	0.58	0.24	Low
398	0.68	0.61	0.34	0.31	0.23	Low
399	0.9	0.78	0.62	0.32	0.89	High
400	0.85	0.82	0.66	0.83	0.83	High
401	0.56	0.6	0.77	0.13	0.32	Low
402	0.66	0.68	0.81	0.57	0.57	Middle
403	0.68	0.64	0.79	0.97	0.24	Middle

Tabel 3. Jumlah Data Per Kelas

No.	Class Attribute	Jumlah Data
1.	Very Low	50
2.	Low	129
3.	Middle	122
4.	High	102
	Jumlah	403

Preprocessing

Preprocessing dilakukan untuk mempersiapkan data yang benar-benar valid sebelum diproses pada tahap berikutnya. Pada tahap ini dilakukan *cleansing*, transformasi, seleksi atribut. Data yang didapat akan diolah untuk mendapatkan atribut yang relevan dan sesuai. Pada penelitian ini,

normalisasi data dilakukan dengan menggunakan metode *Min-Max* dengan rumus berikut.

$$Newdata = \frac{(Data - Min) * (NewMax - NewMin)}{(Max - Min)} + NewMin \quad (5)$$

Keterangan:

- Newdata* : Data hasil normalisasi
Min : Nilai minimum dari data perkolom
Max : Nilai maksimum dari data perkolom
Newmin : Batas nilai minimum yang kita berikan
Newmax : Batas nilai maksimum yang kita berikan

Klasifikasi K-Nearest Neighbor

Setelah hasil normalisasi data diperoleh, untuk selanjutnya dalam proses klasifikasi terhadap data set yang diujikan terlebih dahulu dilakukan pembagian data set menjadi dua bagian yaitu *Data Training* dan *Data Testing* dengan menggunakan metode evaluasi *10Fold Cross Validation* dengan pembagiannya yaitu 9 subset untuk *training* dan 1 subset untuk *testing*.

Kemudian klasifikasi dilakukan dengan dua tahap yaitu pertama dengan KNN secara konvensional atau tanpa menggunakan seleksi fitur dan yang kedua yaitu KNN dengan seleksi fitur yang diperoleh dari proses perhitungan *Relief-F*.

Perbandingan Akurasi Klasifikasi

Evaluasi bertujuan untuk mengetahui tingkat akurasi hasil klasifikasi dataset terhadap data uji. Evaluasi dilakukan untuk menemukan solusi optimal yang dihasilkan dari berbagai metode klasifikasi yang kompleks dan dilakukan berulang-ulang [13].

Oleh karena itu, pada penelitian ini akan menggunakan *Accuracy* karena dianggap memadai. *Accuracy* dapat dirumuskan menggunakan persamaan sebagai berikut:

$$Accuracy = \frac{Jumlah\ Nilai\ Benar}{Jumlah\ Data\ Keseluruhan} \times 100\% \quad (6)$$

HASIL DAN PEMBAHASAN

Hasil Penelitian

Untuk mempermudah pengujian dari metode yang diusulkan pada penelitian ini, penulis dibantu dengan menggunakan *Tools* yaitu *Jupyter Notebook* dan *Rapid Miner Studio* yang berguna dalam mempersingkat waktu pengujian terhadap *User Knowledge Modeling Dataset* untuk melihat hasil perolehan akurasi dari perbandingan KNN secara konvensional dengan kombinasi KNN menggunakan seleksi fitur *Relief-F*.

Kemudian langkah pertama yang dilakukan yaitu melakukan *Preprocessing* untuk menormalisasi data set agar diperoleh data yang valid sebelum dilakukan klasifikasi. Normalisasi data dilakukan menggunakan perhitungan normalisasi *min-max* sesuai dengan persamaan (5) yang telah dijelaskan pada bagian sebelumnya Adapun hasil normalisasi *min-max* pada *User Knowledge Modeling Dataset* yaitu pada Tabel 4 berikut.

Tabel 4. Hasil Normalisasi *User Knowledge Modeling*

No.	X1	X2	X3	X4	X5	Class
1	0	0	0	0	0	very_low
2	0.0808	0.0889	0.1053	0.2424	0.9091	High
3	0.0606	0.0667	0.0526	0.2525	0.3333	Low
4	0.101	0.1111	0.1579	0.6566	0.303	Middle
5	0.0808	0.0889	0.0842	0.9899	0.2424	Low
6	0.0909	0.1667	0.4211	0.101	0.6667	Middle
7	0.101	0.1111	0.4526	0.2929	0.5657	Middle
8	0.1515	0.0222	0.3579	0.404	0.0101	very_low
9	0.202	0.1556	0.3684	0.7273	0.2525	Low
10	0	0	0.5263	0.202	0.8586	High
11	0.1818	0.2	0.5789	0.303	0.8182	High
12	0.0606	0.0667	0.5368	0.4141	0.303	Low
13	0.101	0.1111	0.5474	0.7879	0.3434	Middle
14	0.101	0.1111	0.7368	0.1515	0.9091	High
15	0.202	0.2222	0.7368	0.303	0.6061	Middle
16	0.1212	0.1333	0.7895	0.3535	0.8081	High
17	0.0505	0.0778	0.7368	0.0101	0.0505	very_low
18	0.101	0.2778	0.1053	0.0808	0.3333	Low
19	0.1515	0.3556	0.0526	0.2727	0.2929	Low
20	0.202	0.3222	0.2632	0.4949	0.5657	Middle
21	0.1212	0.3111	0.2105	0.7879	0.202	Low
22	0.1818	0.3333	0.3895	0.1212	0.6667	Middle
23	0.101	0.3	0.3263	0.2929	0.6566	Middle
24	0.1818	0.3444	0.3368	0.4242	0.2828	Low
25	0.0606	0.3222	0.3684	0.7677	0.2525	Low
26	0.0909	0.3333	0.7158	0.1818	0.8586	High
27	0.0404	0.3111	0.5789	0.2525	0.101	very_low
28	0.0909	0.2833	0.6316	0.4545	0.2525	Low
29	0.0808	0.3611	0.6526	0.9495	0.5657	High
30	0.1515	0.3056	0.8421	0.2121	0.8182	High
...
391	0.7879	0.5222	0.3053	0.9899	0.596	High
392	0.5859	0.4444	0.3368	0.2222	0.2424	Low
393	0.6869	0.4778	0.6316	0.4747	0.5556	Middle
394	0.5758	0.4111	0.7895	0.2727	0.3232	Low
395	0.6263	0.6222	0.1158	0.2424	0.2222	Low
396	0.6465	0.6444	0.1474	0.3232	0.2121	Low
397	0.6465	0.6556	0.1263	0.5859	0.2424	Low
398	0.6869	0.6778	0.3579	0.3131	0.2323	Low
399	0.9091	0.8667	0.6526	0.3232	0.899	High
400	0.8586	0.9111	0.6947	0.8384	0.8384	High
401	0.5657	0.6667	0.8105	0.1313	0.3232	Low
402	0.6667	0.7556	0.8526	0.5758	0.5758	Middle
403	0.6869	0.7111	0.8316	0.9798	0.2424	Middle

Kemudian melakukan tahapan dalam seleksi fitur pada *User Knowledge Modeling Dataset* menggunakan seleksi fitur *Relief-F*. Dalam melakukan seleksi fitur dengan *Relief-F*, pada penelitian ini dibantu dengan menggunakan *software Rapid Miner Studio* untuk memperoleh nilai bobot dari masing-masing atribut pada *User Knowledge Modeling Dataset*. Dan menghitung persentase dari bobot data set dilakukan dengan menghitung nilai normalisasi *min-max* dengan rentang persentase dari 0 % sampai 100 %.

Hasil perolehan bobot atribut akan diurutkan berdasarkan urutan nilai bobot terbesar hingga urutan nilai bobot terendah yang bertujuan untuk melihat seberapa besar tingkat pengaruh dari masing-masing atribut terhadap data berdasarkan persentase dari masing-masing atribut.

Tabel 5. Nilai Bobot *Relief-F* Data Set *User Knowledge Modeling*

No.	Atribut	Nilai Bobot	Persentase
1.	X5	0.302	100 %
2.	X4	0.037	10.7 %
3.	X1	0.016	3.8 %
4.	X2	0.007	0.6 %
5.	X3	0.005	0 %

Berdasarkan pada Tabel 4 bobot yang diperoleh dari 5 atribut *User Knowledge Modeling Dataset*. Kemudian dilakukan seleksi fitur untuk membuang fitur dengan nilai persentase terendah. Atribut yang dihapus yaitu X1, X2, X3 karena memiliki persentase rendah yang menunjukkan bahwa atribut tersebut memiliki pengaruh yang kurang signifikan pada data set.

Atribut X4 dan X5 memiliki persentase yang terbesar sehingga dapat diketahui bahwa atribut tersebut memiliki pengaruh yang besar pada data set dan sebagai atribut yang terpilih untuk proses perhitungan klasifikasi.

 Tabel 6. Hasil Seleksi Fitur *User Knowledge Modeling Dataset*

No.	Atribut	Persentase	Keterangan
1.	X5	100 %	Terpilih
2.	X4	10.7%	Terpilih

Langkah selanjutnya yaitu menghitung klasifikasi KNN dengan dua tahap yang pertama yaitu menghitung KNN secara konvensional atau tanpa seleksi fitur, dan yang kedua yaitu KNN dengan seleksi fitur dari *ReliefF*. Adapun hasil akurasi dari KNN secara konvensional dan KNN dengan seleksi fitur dari *ReliefF* dapat dilihat pada Tabel 6 dan Tabel 7 sebagai berikut.

Tabel 7. Hasil Akurasi K-NN Secara Konvensional

<i>k</i>	Jumlah Data Benar	Akurasi (%)
1	338	83.87
2	340	84.37
3	354	87.84
4	340	84.37
5	349	86.60
6	342	84.86
7	347	86.10
8	345	85.61
9	342	84.86
10	341	84.62
Rata-Rata		85.31

 Tabel 8. Hasil Akurasi K-NN dengan *Feature Selection Relief-F*

<i>k</i>	Jumlah Data Benar	Akurasi (%)
1	381	94.54
2	381	94.54
3	383	95.04
4	384	95.29
5	386	95.78
6	388	96.28

7	388	96.28
8	388	96.28
9	388	96.28
10	387	96.03
Rata-Rata		95.63

Pembahasan

Pada bagian ini melakukan pembahasan mengenai hasil pengujian yang diperoleh untuk membuktikan kinerja dari metode yang diusulkan. Untuk lebih jelas perbandingan rata-rata akurasi dari seluruh *k* yang diperoleh KNN konvensional dan KNN + *ReliefF* pada seluruh data set yang digunakan dalam pengujian dapat dilihat pada Tabel 9 berikut.

Tabel 9. Perbandingan Hasil Akurasi

<i>k</i>	K-NN Konvensional (%)	K-NN + ReliefF (%)	Selisih Akurasi (%)
1	83.87	94.54	10.67
2	84.37	94.54	10.17
3	87.84	95.04	7.2
4	84.37	95.29	10.92
5	86.60	95.78	9.18
6	84.86	96.28	11.42
7	86.10	96.28	10.18
8	85.61	96.28	10.67
9	84.86	96.28	11.42
10	84.62	96.03	11.41
Rata-Rata	85.31	95.63	10.32

Dari Tabel 9 dapat dilihat bahwa hasil akurasi setelah melakukan seleksi atribut pada KNN dengan *Relief-F* diperoleh peningkatan nilai akurasi dibandingkan nilai akurasi yang dihasilkan oleh KNN konvensional dengan selisih rata-rata akurasi sebesar 10.32 %. Pada KNN + *ReliefF* akurasi tertinggi diperoleh saat *k* bernilai 6, 7, 8 dan 9 yaitu sebesar 96.28 %, sedangkan nilai akurasi terendah diperoleh pada saat *k* bernilai 1 dan 2 yaitu sebesar 94.54 %. Pada KNN Konvensional akurasi tertinggi diperoleh saat *k* bernilai 3 yaitu sebesar 87.84 %, sedangkan nilai akurasi terendah diperoleh pada saat *k* bernilai 1 sebesar 83.87 %. Rata-rata akurasi dari seluruh *k* Pada KNN + *Relief-F* yaitu sebesar 95.63 % dan pada KNN konvensional sebesar 85.31 %. Berdasarkan metode yang diusulkan terlihat bahwa KNN menggunakan *Relief-F* mampu meningkatkan akurasi yang lebih baik dibandingkan akurasi dari KNN tanpa menggunakan seleksi fitur.

KESIMPULAN

Berdasarkan hasil pengujian dan evaluasi terhadap metode KNN dengan seleksi atribut menggunakan *Relief-F*, maka didapatkan kesimpulan yaitu dengan melakukan seleksi atribut pada KNN menggunakan *Relief-F* terbukti dapat meningkatkan akurasi pada KNN.

Hasil rata-rata peningkatan akurasi adalah sebesar 10.32 %. Hasil pengujian seleksi atribut pada KNN dengan *Relief-F* mampu meningkatkan nilai akurasi pada *User Knowledge Modeling Dataset* dari seluruh *k* Pada KNN + *Relief-F* yaitu sebesar 95.63 % dan pada KNN konvensional sebesar 85.31 %.

REFERENSI

- [1] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature Selection: Ever Evolving Frontier in Data Mining,". In *Feature selection in data mining*, pp. 4-13. 2010.
- [2] D. A. Irawan, Z. A. Baizal, and E. G. Perdana, "Analisis dan Implementasi Algoritma Relieff untuk Feature Selection pada Klasifikasi Dataset Multiclass (Doctoral dissertation, MS thesis, Universitas Telkom, Jakarta, Indonesia,". 2011.
- [3] Y. Chen, and Y. Hao, "A Feature Weighted Support Vector Machine and K-Nearest Neighbor Algorithm for Stock Market Indices Prediction," *Expert Systems with Applications* (2017), vol. 80, pp. 340-355, 2017.
- [4] J. S. Raikwal, and K. Saxena, "Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data set," *International Journal of Computer Applications*. vol. 50, no. 14, pp. 35-39, 2012.
- [5] A. Ashari, I. Paryudi, and A. M. Tjoa, "Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool," (IJACSA) *International Journal of Advanced Computer Science and Applications*. vol. 4, no. 11, pp. 33-39, 2013.
- [6] M. Danil, S. Efendi, and R. W. Sembiring, "The Analysis of Attribution Reduction of K-Nearest Neighbor (KNN) Algorithm by Using Chi-Square," In *Journal of Physics: Conference Series*, vol. 1424, no. 1, pp. 012004, 2019.
- [7] T. R. Reddy, B. V. Vardhan, M. GopiChand, and K. Karunakar, "Gender prediction in author profiling using ReliefF feature selection algorithm," In *Intelligent Engineering Informatics*, pp. 169-176, Springer, Singapore, 2018.
- [8] J. Huang, J. Zhou, and L. Zheng, "Support Vector Machine Classification Algorithm Based on Relief-F Feature Weighting," In *2020 International Conference on Computer Engineering and Application (ICCEA)*, pp. 547-553, 2020.
- [9] J. Han, J. Pei, and M. Kamber, "Data Mining Concept and Techniques, 3rd edition," *Morgan Kaufmann-Elsevier*. vol. 2, no. 1, pp. 88-97, 2012.
- [10] A. Danades, D. Pratama, D. Anggraini, and D. Anggriani, "Comparison of Accuracy Level K-Nearest Neighbor Algorithm and Support Vector Machine Algorithm in Classification Water Quality Status," *International Conference on System Engineering and Technology*, pp. 137-141, 2016.
- [11] I. Kononenko, "Estimating Attributes: Analysis and Extensions of Relief," In *European conference on machine learning*, pp. 171-182, 1994.
- [12] P. Refaeilzadeh, L. Tang, and H. Liu, "Encyclopedia of Database Systems," In *Cross-validation*, pp. 532-538, 2009.
- [13] J. D. Novaković, A. Veljović, S. S. Ilić, Ž. Papić, and T. Milica, "Evaluation of Classification Models in Machine Learning," *Theory and Applications of Mathematics & Computer Science*, vol. 7, no. 1, pp. 39-46, 2017.