JURNAL PINTER
VOL 3. NO.1 JUNI 2019
e-ISSN: 2597- 4475
http://doi.org/10.21009/pinter.3.1.9

Bayes Classifier dan Support Vector Machine dalam Klasifikasi Judul Karya Akhir Mahasiswa Program Studi PTIK UNJ

Razi Aziz Syahputro[1], Widodo[2], Hamidillah Ajie[3]

[1] Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ
[2,3] Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ
[1]razisyahputro@gmail.com, [2] widodo03@yahoo.com, [3] hamidillah@yahoo.com

Abstrak

Penelitian ini dilatarbelakangi dengan dibutuhkannya sistem pengklasifikasian untuk memudahkan pihak Jurusan Teknik Elektro khususnya Program Studi PTIK untuk mengklasifikasikan judul skripsi berdasarkan peminatan. Sebelum sistem dibuat diperlukan pertimbangan dari beberapa algoritma klasifikasi yang ada, maka dari itu penelitian ini memilih 3 algoritma dari 10 algoritma terbaik menurut ICDM tahun 2006. Klasifikasi terhadap dokumen teks pendek seperti judul skripsi mahasiswa memiliki kesulitan tersendiri daripada dokumen teks panjang karena semakin sedikit kata semakin sulit diklasifikasi. Sehingga tujuan dari penelitian ini adalah untuk mengetahui algoritma yang paling efektif untuk mengklasifikasi judul skripsi. Penelitian ini terdiri dari beberapa tahap yaitu pengumpulan data, pengelompokan data melalui angket oleh dosen ahli, *pre-processing text*, pembobotan kata menggunakan *vector space model* dan *tf-idf*, evaluasi dengan *k-fold cross validation*, klasifikasi menggunakan *k-nearest neighbor, naïve bayes classifier*, *dan support vector machine*, dan analisis dengan *confusion matrix*. Percobaan dilakukan dengan menggunakan 266 data judul skripsi mahasiswa PTIK UNJ dari angkatan 2010-2013, dengan data terakhir berasal dari sidang skripsi pada semester 105(semester ganjil 2016/2017). Hasil dari klasifikasi menggunakan algoritma tersebut didapatkan algoritma yang paling efisien yaitu *support vector machine* dengan akurasi 82% dari 10 kali percobaan.

Kata kunci : Klasifikasi, *K-Nearest Neighbor, Naïve Bayes Classifier, Support Vector Machine*, judul, akurasi

1. Pendahuluan

Perkembangan teknologi saat ini berlangsung dengan sangat pesat disegala bidang kehidupan. Pesatnya teknologi menghasilkan banyak sekali data yang dihasilkan oleh teknologi yang canggih ini terutama penerapan teknologi informasi dalam dunia pendidikan yang menghasilkan data yang berlimpah mengenai mahasiswa khususnya di Program Studi Pendidikan Teknik Informatika dan Komputer Univeritas Negeri Jakarta dalam proses pembelajaran yang dihasilkan. Salah satunya adalah data karya akhir mahasiswa yang dihasilkan setiap semester selama menempuh proses kegiatan belajar mengajar.

Informasi yang berlimpah tersebut, apabila ada kebutuhan untuk proses pencarian secara manual akan menghabiskan waktu dan tenaga, maka manfaat dari informasi yang diperoleh berkurang. Jika data mahasiswa dan karya akhirnya dapat diolah, maka pengolahan data dapat dilakukan untuk menghasilkan informasi penting berupa pengetahuan baru. Pengetahuan baru tesebut dapat membantu pihak universitas untuk melakukan klasifikasi mengenai judul karya akhir mahasiswa berdasarkan peminatan guna untuk memudahkah proses pencarian.

Format data teks yang tidak terstruktur membuatnya sulit untuk diolah oleh komputer dalam proses pengklasifikasian karena banyak sekali mengandung noise yang membuat proses pengklasifikasian lebih sulit dilakukan. Text mining adalah salah satu teknik yang dapat digunakan untuk melakukan klasifikasi dokumen yang berurusan dengan data tidak terstruktur yang merupakan pembeda utama dengan data mining yang menggunakan data terstruktur. Klasifikasi teks bertujuan untuk menentukan kelas kelas atau kategori dari suatu teks. Ada 10 algoritma terbaik dalam data mining, yaitu C4.5, K-Means, SVM, Apriori, EM, PageRank, AdaBoost, KNN, Naive Bayes, dan CART (ICDM: 2006). Dengan banyaknya algoritma klasifikasi yang dapat digunakan dalam klasifikasi dokumen, maka diperlukan suatu perbandingan yang tujuannya untuk mengetahui algoritma mana yang dapat menghasilkan performa lebih baik dari algoritma yang digunakan tersebut. Efisiensi algoritma klasifikasi dapat diukur berdasarkan beberapa parameter, seperti kecepatan, proses, keakuratan, dan kesalahan.

Pada penelitan ini algoritma KNN, NBC, dan SVM akan digunakan untuk melakukan klasifikasi pada judul karya akhir mahasiswa Universitas Negeri Jakarta prodi Pendidikan Teknik Informatika dan Komputer berdasarkan peminatan. Klasifikasi berdasarkan judul pada penelitian ini menggunakan data dari Teknik Elektro Universitas Negeri Jakarta. Judul karya akhir memiliki karakteristik yang berbeda dengan data tweet, karana judul karya akhir

merupakan serangkaian kata yang disusun secara sistematis sehingga meminimalisir terjadinya ambiguitas dan lebih sedikit mengandung noise daripada data tweet yang cenderung informal. Sehingga akhirnya akan diketahui algoritma manakah yang lebih cepat, lebih akurat, atau yang lebih banyak melakukan kesalahan. Setelah mengetahui algoritma yang lebih akurat dan tepat untuk klasifikasi judul karya akhir diharapkan dapat membantu pembuatan sistem klasifikasi karya akhir mahasiswa berdasarkan judul.

2. Dasar Teori

2.1. Data Mining

Data mining juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah *data mining* kadang disebut juga *Knowledge Discovery Data* (KDD). Pengelompokan data juga bisa dilakukan, tujuannya adalah agar kita dapat mengetahui pola universal data-data yang ada.

2.2. Text Mining

Text mining merupakan bagian integral dari data mining yang bertujuan mengekstrak pengetahuan secara otomatis dari data tekstual tidak terstruktur. Tugas utama text mining meliputi klasifikasi teks, ringkasan teks, dokumen dan/atau kata clustering, selain itu tugasnya memproses bahasa alami klasik seperti mesin terjemahan dan tanya-jawab (Berry dan Kogan, 2010:183).

Menurut Langgeni, dkk. (2010: 2), diacu dalam Reynaldo (2016: 5) langkah-langkah dalam text preprocessing adalah:

a. Case Folding

Tahap mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karena selain huruf dihilangkan dianggap delimeter.

b. Tokenizing

Tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Secara garis besar tokenisasi adalah tahap memecah sekumpulan karakter dalam suatu teks kedalam satuan kata. Sekumpulan karakter tersebut dapat berupa karakter whitspace, seperti enter, tabulasi, spasi. Namun untuk karakter petik tunggal ('), titik (.), semikolon (;), titik dua (:) atau lainnya juga dapat memiliki peran yang cukup banyak sebagai pemisah kata.

c. Filtering

Tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata-kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist* atau *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang. Contoh stopwords adalah "yang", "dan", "di", "dari", dan seterusnya.

d. Stemming

Menurut Tala (2003), diacu dalam Manalu (2014: 19) setelah melalui proses *stopword removal* tindakan selanjutnya adalah yaitu proses *stemming*. *Stemming* adalah proses pemetaan dan penguraian berbagai bentuk (varian) dari suatu kata menjadi bentuk kata dasarnya (*stem*). Tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun infiks yang ada pada setiap kata.

2.3. TF-IDF(*Terms Frequency-Inverse Document Frequency*)

Menurut Robertson (2005) diacu dalam Intan dan Defeng (2006: 3) Model TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut.

2.4. Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (1) pembangunan model sebagai prototipe untuk disimpan sebagai memori dan (2) penggunaan model tersebut untuk melakukan klasifikasi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya (Prasetyo, 2012: 45). Dalam pembanguan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algortima pelatihan. Berikut algoritma pelatihan yang penulis teliti:

a. K-Nearest Neighbor

Menurut Prasetyo (2012: 48) *K-Nearest Neighbor* merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain. Salah satu masalah yang dihadapi K-NN adalah pemilihan *K* yang tepat. Cara voting mayoritas dari *K-nearest* untuk nilai *K* yang besar bisa mengakibatkan distorsi data yang besar, untuk menangani masalah voting mayoritas tersebut biasanya ditambahkan penggunaan bobot untuk menghitung kandidat kelas. Bobot dari setiap *nearest neighbor* dihitung dengan formula:

$$y' = \text{argmax} \sum_{(x_i,y_i)\in D_z} w_i \; x \; I\,(v = y_i) \quad (1)$$

b. Naïve Bayes Classifier

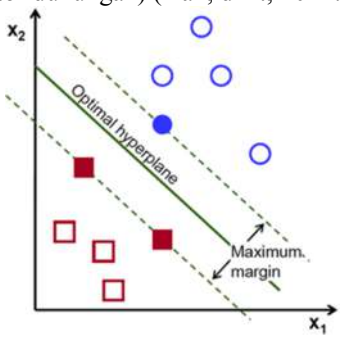
Bayes (terutama Naïve Bayes) merupakan teknik prediksi berbasis probabilistik sederhana yang berdasarkan pada penerapan teorema Bayes (atau aturan Bayes) dengan asumsi ketidaktergantungan yang kuat. Dengan kata lain, dalam Naïve Bayes, model yang digunakan

adalah "model fitur independen". Maksud indepedensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama (Prasetyo, 2012: 59). Klasifikasi *Naïve Bayes* dengan formula umum sebagai berikut:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{q} P(X_i|Y)}{P(X)} \quad (2)$$

c. Support Vector Machine

SVM adalah sebuah metode untuk klasifikasi bagi data *linear* maupun *nonlinear*. SVM merupakan algoritma yang bekerja dengan menggunakan pemetaan *nonlinear* untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dalam dimensi baru ini, akan mencari *linear* yang optimal untuk memisahkan *hyperplane* (yaitu, "batas keputusan" yang memisahkan *tupel* dari satu kelas ke kelas lainnya). Dengan pemetaan *nonlinear* yang tepat untuk dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan dengan *hyperplane*. SVM menentukan *hyperplane* menggunakan vektor dukungan (*tupel* pelatihan "penting") dan *margin* (didefinisikan oleh vektor dukungan) (Han, dkk., 2012: 408).



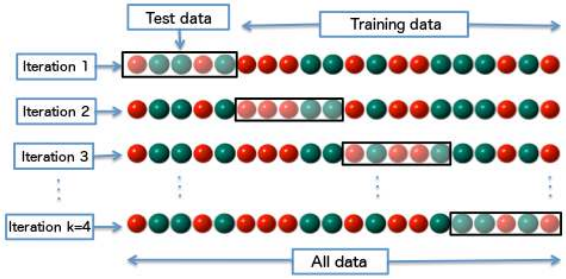
Gambar 2.1. Margin Hyperlane

2.5. Dokumen Teks Pendek

Menurut Timonen, et al., (2013), diacu dalam Dini (2015: 4) mendefinisikan dokumen pendek sebagai dokumen yang berisi tidak lebih dari 100 kata, yang sama dengan abstrak ilmiah yang sangat singkat. Menurut Sistiani (2015:5-6), diacu dalam Afendi (2017: 24) jenis-jenis dokumen menurut sifatnya ada dua yaitu dokumen tekstual dan dokumen non-tekstual. Dokumen tekstual adalah dokumen yang menyajikan informasi dalam bentuk tulisan, contohnya jurnal, majalah, buku, dan sebagainya. Dokumen teks pendek termasuk jenis dokumen tekstual.

2.6. K-Fold Cross Validation

Bentuk umum pendekatan ini disebut dengan *K-fold Cross Validation*, yang memecah set data menjadi *k* bagian set data dengan ukuran yang sama. Setiap kali berjalan, satu pecahan berperan sebagai set data uji sedangkan pecahan lainnya menjadi set data latih. Prosedur tersebut dilakukan sebanyak *k* kali sehingga setiap data berkesempatan menjadi data uji tepat satu kali dan menjadi data latih sebanyak *k*-1 kali. Total *error* didapatkan dengan menjumlah semua *error* yang didapatkan dari *k* kali proses (Prasetyo, 2014: 264, diacu dalam Fakhriyani, 2016: 20).



Gambar 2.2 Ilustrasi Tahapan K-Fold Cross Validation

2.7. Confusion Matrix

Menurut Han, dkk. (2012: 365-366) *Confusion Matrix* adalah alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. TP dan TN memberikan informasi ketika *classifier* benar, sedangkan FP dan FN memberitahukan ketika *classifier* salah. Contoh *Confusion Matrix* ditunjukkan pada Tabel 1.

Tabel 2.1. Confusion Matrix

		Predicted Class		
		Yes	No	Total
Actual Class	Yes	TP	FN	P
	No	FP	TN	N
	Total	P'	N''	P+N

3. Metodologi

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Multimedia Program Studi Pendidikan Teknik Informatika dan Komputer Fakultas Teknik Universitas Negeri Jakarta yang berlokasi di Jl. Rawamangun Muka Jakarta. Penelitian ini dilaksanakan pada semester genap (106) selama 4 bulan terhitung mulai Maret 2017 hingga Juni 2017.

3.2. Alat dan Bahan Penelitian

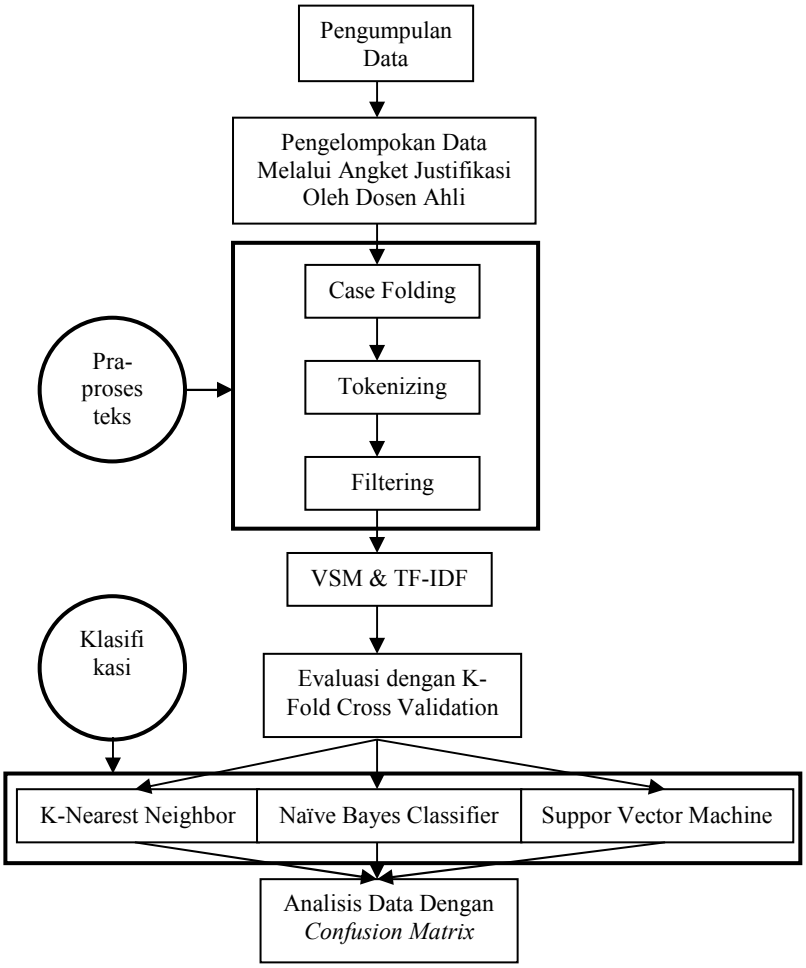
1. Perangkat keras yang dibutuhkan:
   a. *Processor* Intel® Core™ i3-5010U CPU @ 2.10Ghz
   b. Memory RAM 6GB DDR3
   c. Layar 14"

2. Perangkat Lunak yang dibutuhkan:
   a. Sistem operasi Windows 8.1 Professional 64-bit
   b. Jetbrains Pycharm Community Edition 2017.1 untuk memproses algoritma *K-*

Nearest Neighbor, Naive Bayes, dan Support Vector Machine

c. Anaconda versi 4.3.1 sebagai Depedency Manager dari aplikasi Pycharm
d. Menggunakan modul klasifikasi, TfI-df, K-fold, metrics, numpy, dan lainnnya dari fungsi yang ada pada www.scikit-learn.org
e. Notepad untuk mengolah data teks

3.3. Diagram Alir Penelitian



Gambar 3.1. Diagram Alir Penelitian

Berdasarkan data yang telah dikumpulkan, peneliti membuat angket pengkategorian manual kepada bapak Widodo sebagai Ketua Kelompok Bidang Ilmu Rekayasa Perangkat Lunak pada Program Studi Pendidikan Teknik Informatika dan Komputer yang berisi seluruh judul skripsi untuk dikategorikan secara manual apakah masuk ke kelompok Pendidikan, RPL, TKJ, atau MM.

Setelah pengelompokan data selesai, tahap selanjutnya adalah pra-proses data. Pra-proses dalam penelitian ini meliputi tiga tahap, yaitu:

1. *Case folding* mengubah semua huruf yang ada di dalam file menjadi huruf kecil semua.
2. *Tokenizing* merupakan tahap memecah sekumpulan karakter kedalam satuan kata
3. *Filtering* merupakan tahap membuang kata-kata penting dari hasil tokenizing. *Stopwords* (kata-kata tidak penting) yang digunakan adalah stopword Tala (https://github.com/masdevid/ID-Stopwords). Berisi 758 kata untuk dijadikan dasar membuang kata yang tak penting.

Setelah itu akan terbentuk *bag of words* yang memuat dokumen yang telah bersih dari kata-kata tidak penting. Dokumen-dokumen yang sudah bersih kemudian di ubah kedalam ruang vektor, yang peneliti gunakan menggunakan pembobotan Tf-Idf (*Term Frequency – Inverse Document Frequency*). Setelah melalui tahap-tahap diatas maka data siap diklasifikasi, pada penelitian ini data yang diklasifikasi menggunakan algoritma *K-Nearest Neighbor*, *Naïve Bayes Classifier*, dan *Support Vector Machine* dan dievaluasi dengan teknik *K-Fold Cross Validation*, dengan nilai K pada penelitian ini adalah 10 yang artinya akan ada 10 kali *learning testing*. Berikut pembagian skema dalam *K-Fold Cross Validation*:

1. K1 = data 1 – 27
2. K2 = data 28 – 54
3. K3 = data 55 – 81
4. K4 = data 82 – 108
5. K5 = data 109 – 135
6. K6 = data 136 – 162
7. K7 = data 163 – 188
8. K8 = data 189 – 214
9. K9 = data 215 – 240
10. K10 =data 241 – 266

Tabel 3.1. K-Fold Cross Validation

Eksperimen	Data Train	Data Test
1	K2, K3, K4, K5, K6, K7. K8, K9, K10	K1
2	K1, K3, K4, K5, K6, K7. K8, K9, K10	K2
3	K1, K2, K4, K5, K6, K7. K8, K9, K10	K3
4	K1, K2, K3, K5, K6, K7. K8, K9, K10	K4
5	K1, K2, K3, K4, K6, K7. K8, K9, K10	K5
6	K1, K2, K3, K4, K5, K7. K8, K9, K10	K6
7	K1, K2, K3, K4, K5, K6. K8, K9, K10	K7
8	K1, K2, K3, K4, K5, K6. K7, K9, K10	K8
9	K1, K2, K3, K4, K5, K6, K7, K8, K10	K9
10	K1, K2, K3, K4, K5, K6, K7, K8, K9	K10

Data yang telah terklasifikasi oleh ketiga algoritma tersebut akan dianalisis dengan *confusion matrix*. *Confusion Matrix* adalah merupakan alat untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. Analisis dilakukan dengan menggunakan rumus *accuracy*, *precision*, *recall* dan *F1-Score.* dari *confusion matrix*. *Accuracy* adalah persentase test set tuple

yang dikelompokkan dengan benar oleh *classifier*, untuk menghitungnya digunakan rumus berikut:

$$Accuracy = \frac{TP+TN}{P+N} \quad (3)$$

Precision merupakan ukuran ketepatan (berapa persen dari tuple diidentifikasi sebagai positif dengan benar) dengan rumus:

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

sedangkan *recall* adalah ukuran dari kelangkapan (berapa persen tuple positif diberi label seperti itu) dengan rumus:

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{P} \quad (5)$$

Perhitungan *precision* dan *recall* dapat menghasilkan *F1-Score* yang merupakan rata-rata *weighted*, dengan rumus:

$$F = \frac{2\ x\ precision\ x\ recall}{precision + recall} \quad (6)$$

4. Hasil dan Analisis

4.1. Deskripsi Hasil Penelitian

Data yang diperoleh dalam penelitian ini diambil dari administrasi Program Studi PTIK. Data yang diambil adalah judul skripsi dari hasil proses penyelesaian perbaikan sidang skripsi. Data untuk penelitian ini adalah 266 data judul skripsi yang berasal dari angkatan 2010-2013, dengan data terakhir berasal dari sidang skripsi pada semester 105 (semester ganjil 2016/2017).

4.2. Hasil Penelitian

Berikut Fold pertama dari perbandingan 3 Algoritma:

```
Banyak Data Latih:  239
Banyak Data Uji:  27

Data Latih:
 [ 27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116
 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134
 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152
 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188
 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206
 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224
 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242
 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
 261 262 263 264 265]

Data Uji:
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26]
```

Gambar 4.1. Pembagian data uji dan Latih Iterasi ke 1

```
Hasil Prediksi:
 [1 1 1 2 1 1 1 1 0 2 3 3 1 1 1 1 2 2 3 1 3 3 1 2 1 3 2]

Confusion Matrix:
 [[1 0 0 1]
 [0 9 0 0]
 [0 2 6 1]
 [0 3 0 4]]

Akurasi:  0.740740740741

              precision    recall  f1-score   support

 Pendidikan       1.00      0.50      0.67         2
        RPL       0.64      1.00      0.78         9
        TKJ       1.00      0.67      0.80         9
         MM       0.67      0.57      0.62         7

avg / total       0.79      0.74      0.74        27
```

Gambar 4.2. Hasil Klasifikasi Algoritma KNN ke-1

```
Hasil Prediksi:
 [1 1 1 1 1 1 1 0 1 3 3 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1]

Confusion Matrix:
 [[1 0 0 1]
 [0 9 0 0]
 [0 9 0 0]
 [0 4 0 3]]

Akurasi:  0.481481481481

              precision    recall  f1-score   support

 Pendidikan       1.00      0.50      0.67         2
        RPL       0.41      1.00      0.58         9
        TKJ       0.00      0.00      0.00         9
         MM       0.75      0.43      0.55         7

avg / total       0.40      0.48      0.38        27
```

Gambar 4.3. Hasil Klasifikasi Algoritma NBC ke-1

```
Hasil Prediksi:
 [1 1 1 2 1 1 1 1 0 2 3 3 1 1 1 1 2 1 3 1 3 1 1 2 1 2 1]

Confusion Matrix:
 [[1 0 0 1]
 [0 9 0 0]
 [0 4 5 0]
 [0 4 0 3]]

Akurasi:  0.666666666667

              precision    recall  f1-score   support

 Pendidikan       1.00      0.50      0.67         2
        RPL       0.53      1.00      0.69         9
        TKJ       1.00      0.56      0.71         9
         MM       0.75      0.43      0.55         7

avg / total       0.78      0.67      0.66        27
```

Gambar 4.4. Hasil Klasifikasi Algoritma SVM ke-1

Tabel 4.1. Confusion Matrix Iterasi Ke-1

		Predicted Class				Total
		Pendidikan	RPL	TKJ	MM	
Actual Class	Pendidikan	1 1 1	0 0 0	0 0 0	1 1 1	2
	RPL	0 0 0	9 9 9	0 0 0	0 0 0	9
	TKJ	0 0 0	2 9 4	6 0 5	1 0 0	9
	MM	0 0 0	3 4 4	0 0 0	4 3 3	7
Total		1 1 1	14 22 17	6 0 5	6 4 4	***27***

Algoritma KNN Algoritma NBC Algoritma SVM

Pada iterasi pertama dengan menggunakan 3 algoritma klasifikasi, kelas Pendidikan diprediksi dengan benar sebanyak 1 data dan 1 data lainnya salah prediksi pada masing-masing algoritma. Pada kelas RPL diprediksi dengan benar sebanyak 9 data pada masing-masing algoritma. Pada kelas TKJ dengan menggunakan algoritma KNN diprediksi dengan benar sebanyak 6 data, 2 data sebagai kelas RPL dan 1 data sebagai kelas MM diprediksi salah, sedangkan dengan menggunakan algoritma NBC 9 data masuk ke kelas RPL(diprediksi salah), sedangkan dengan menggunakan algoritma SVM diprediksi dengan benar 5 data, 4 data sebagai kelas RPL diprediksi salah. Pada kelas MM diprediksi dengan menggunakan algoritma KNN diprediksi dengan benar sebanyak 4 data dan 3 data sebagai kelas RPL diprediksi salah, sedangkan dengan menggunakan algortima NBC dan SVM diprediksi dengan benar sebanyak 3 data dan 4 data sebagai kelas RPL diprediksi salah.

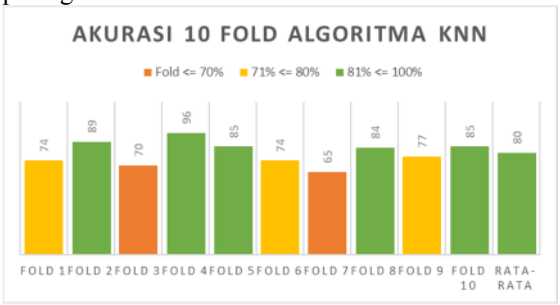
Berdasarkan tabel *confusion matrix* Tabel 4.2 maka dapat dihitung akurasi, *precision, recall*, dan *f1-Score*, berikut perbandingan 3 algoritma rata-rata akurasi, *precision, recall,* dan *f1-Score*, yang bisa dilihat pada tabel dibawah ini :

Tabel 4.2. Rata-Rata Akurasi, *Precision, Recall*, dan *F1-Score*

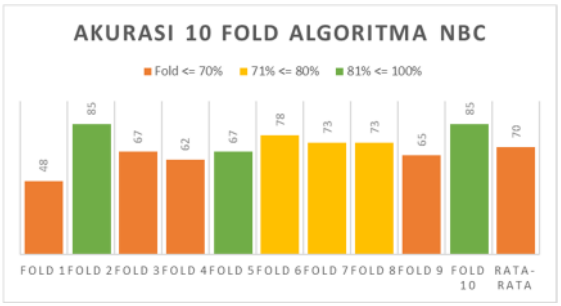
	K-Nearest Neighbor	Naïve Bayes Classifier	Support Vector Machine
Akurasi	0.74074074 0741	0.48148148 1481	0.66666666 6667
Precision	0.79	0.40	0.78
Recall	0.74	0.48	0.67
F1-Score	0.74	0.38	0.66

Hasil perbandingan dari 3 algoritma klasifikasi yang paling tinggi nilainya dapat terlihat bahwa pada iterasi pertama akurasi, *precision, recall*, dan *f1-score* berturut-turut yang dipersentasekan adalah 74%, 79%, 74%, dan 74% yang merupakan algortima KNN.

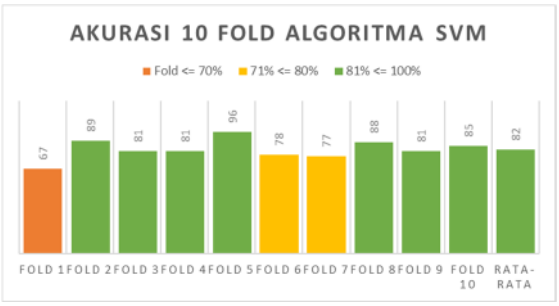
Berdasarkan hasil penelitian, dapat dilihat akurasi dari 10 kali iterasi pada masing-masing algoritma klasifikasi yang dilakukan dapat dilihat pada gambar dibawah ini:



Gambar 4.5. Akurasi Algoritma KNN 10x Iterasi



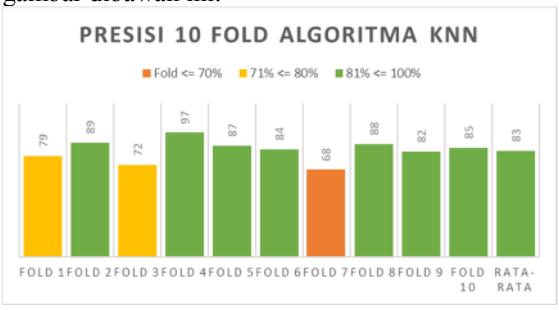
Gambar 4.6. Akurasi Algoritma NBC 10x Iterasi



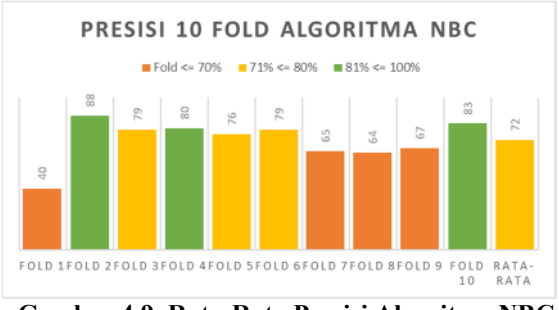
Gambar 4.7. Akurasi Algoritma SVM 10x Iterasi

Dari 10 kali iterasi, didapatkan rata-rata akurasi pada algoritma KNN sebesar 80%, algoritma NBC sebesar 70%, dan algoritma SVM sebesar 82%. Maka dapat disimpulkan bahwa algoritma *support vector machine* akurasi-nya yang paling tinggi disusul dengan algortima *k-nearest neighbor* dan *naïve bayes classifier*. Akurasi merupakan persentase test set *tuple* yang dikelompokkan dengan benar oleh *classifier*.

Presisi dari setiap klasifikasi dapat dilihat pada gambar dibawah ini:

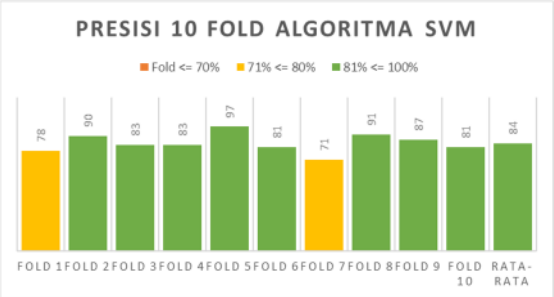


Gambar 4.8. Rata-Rata Presisi Algoritma KNN 10 Kali Iterasi



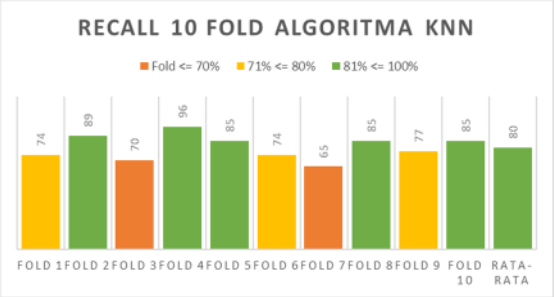
Gambar 4.9. Rata-Rata Presisi Algoritma NBC 10 Kali Iterasi

Avalaiable at:
http://journal.unj.ac.id/unj/index.php/pinter/article/view/16358

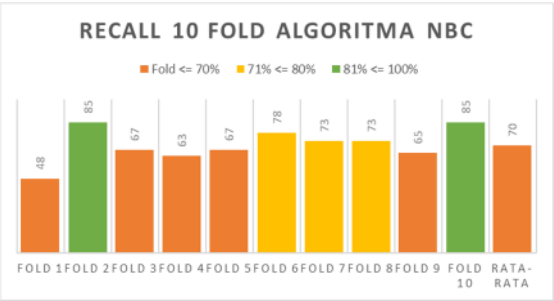


Gambar 4.10. Rata-Rata Presisi Algoritma SVM 10 Kali Iterasi

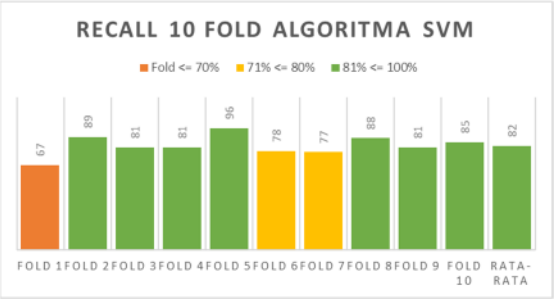
Dari 10 kali iterasi, didapatkan rata-rata presisi pada algoritma KNN sebesar 83%, algoritma NBC sebesar 72%, dan algoritma SVM sebesar 84%. Maka dapat disimpulkan bahwa algoritma *support vector machine* presisi-nya yang paling tinggi disusul dengan algortima *k-nearest neighbor* dan *naïve bayes classifier*. Presisi merupakan ketepatan prediksi algoritma dengan informasi yang diinginkan peneliti. recall dari setiap klasifikasi dapat dilihat pada gambar dibawah ini :



Gambar 4.11. Rata-Rata Recall Algoritma KNN 10 Kali Iterasi

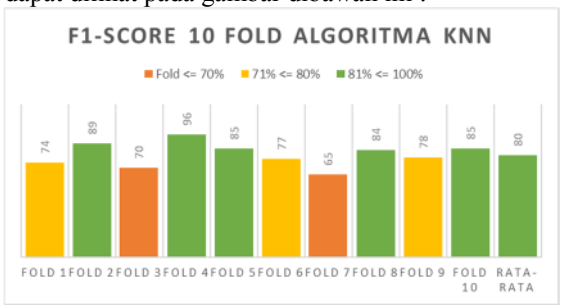


Gambar 4.12. Rata-Rata Recall Algoritma NBC 10 Kali Iterasi

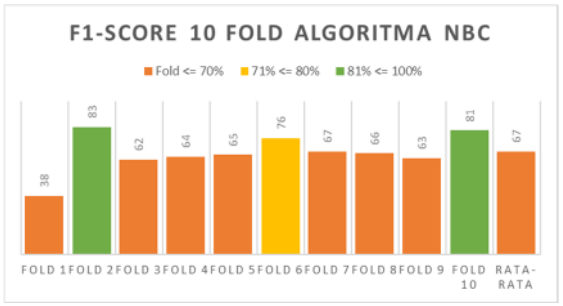


Gambar 4.13. Rata-Rata Recall Algoritma SVM 10 Kali Iterasi

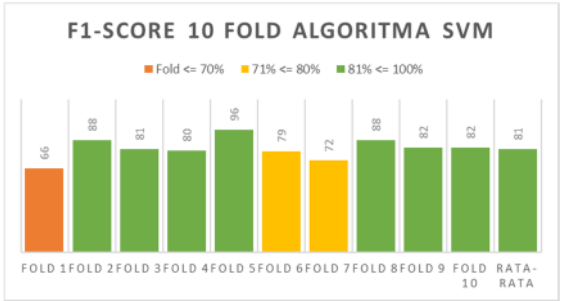
Dari 10 kali iterasi, didapatkan rata-rata *recall* pada algoritma KNN sebesar 80%, algoritma NBC sebesar 70%, dan algoritma SVM sebesar 82%. Maka dapat disimpulkan bahwa algoritma *support vector machine recall*-nya yang paling tinggi disusul dengan algortima *k-nearest neighbor* dan *naïve bayes classifier*. *Recall* merupakan keberhasilan algoritma menemukan kembali informasi. Terakhir adalah perhitungan *f1-score* dari setiap klasifikasi dapat dilihat pada gambar dibawah ini :



Gambar 4.14. Rata-Rata F1-Score Algoritma KNN 10 Kali Iterasi



Gambar 4.15. Rata-Rata F1-Score Algoritma NBC 10 Kali Iterasi



Gambar 4.16. Rata-Rata F1-Score Algoritma SVM 10 Kali Iterasi

Dari 10 kali iterasi, didapatkan rata-rata *f1-score* pada algoritma KNN sebesar 80%, algoritma NBC sebesar 67%, dan algoritma SVM sebesar 81%. Maka dapat disimpulkan bahwa algoritma *support vector machine f1-score* nya yang paling tinggi disusul dengan algortima *k-nearest neighbor* dan *naïve bayes classifier*. *F1-score* merupakan keselarasan antara presisi dan *recall*.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Data yang diperoleh didapat dari seluruh hasil studi karya akhir mahasiswa PTIK UNJ dari seluruh hasil sidang skripsi dari angkatan tahun 2010-2013, dengan data terakhir berasal dari semester 105. Berdasarkan hasil penelitian yang telah dilakukan bertahap menggunakan algoritma k-nearest neighbor, naïve bayes classifier, dan support vector machine untuk perbandingan dalam proses klasifikasi judul skripsi, dapat diambil kesimpulan sebagai berikut:

1. Dari hasil perbandingan algoritma klasifikasi dari 3 algoritma yang peneliti pilih, maka performa yang terbaik dari algoritma tersebut adalah algoritma support vector machine dengan akurasi sebesar 82%, precision sebesar 84%, recall sebesar 82%, dan f1-score 81%
2. Perbandingan dari beberapa algoritma yang terbaik untuk penelitian berdasarkan data judul skripsi PTIK UNJ dari angkatan 2010-2013 adalah support vector machine.

5.2. Saran

Peneliti memiliki beberapa saran untuk penelitian selanjutnya yang berhubungan dengan text mining dan algoritma support vector machine khususnya yang berhubungan dengan judul yaitu:

1. Algoritma klasifikasi masih banyak lagi dari kategori eager learner dan lazy learner, algoritma yang sudah peneliti bandingkan bisa dicoba lagi bandingkan dengan algoritma yang belum diteliti.
2. Algoritma support vector machine bisa ditingkatkan lagi akurasinya menggunakan metode extending document term dari beberapa algoritma yang ada.
3. Bisa dicoba juga menggunakan bahasa pemrograman lain yang mendukung library machine learning untuk mengetahui kecepatan dalam proses dan kesamaan akurasinya.

Daftar Pustaka

Afendi, M. (2017). Klasifikasi Dokumen Karya Akhir Mahasiswa Program Studi Pendidikan Teknik Informatika dan Komputer Universitas Negeri Jakarta berdasarkan Peminatan menggunakan Algoritma Support Vector Machine[skripsi]. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Arisputranto, I. B,. (2016). Kinerja Algoritma Naive Bayes untuk Mendeteksi Akurasi Berita Gempa Bumi melalui Twitter berbahasa Indonesia. Naskah Publikasi Jurnal, 1-10.

Berry, M. W., & Kogan, J. (2010). *Text Mining Applications and Theory First-Ed*. UK:Wiley.

Brown, Susan. (2014). *"Text Mining and Visualization for Digital Literary History." The Canadian Writing Research Collaboratory.* Canadian Foundation for Innovation. 1 Dec. 2010. Web 9 Dec. 2014

Dini, E. P,. (2015). Prediksi Topik pada Media Sosial Twitter menggunakan K-Means Clustering dan Naive Bayes Classifier. Naskah Publikasi Jurnal, 1-7.

Fakhriyani. (2016). Perbandingan Algoritma Naive Bayes dan Support Vector Machine dalam Seleksi Kelulusan Pemberkasan Beasiswa BPP-PPA Fakultas Teknik Universitas Negeri Jakarta[skripsi]. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Fakultas Teknik. (2015). *Buku Panduan Penyusunan Skripsi dan Non Skripsi*. Jakarta: FTUNJ.

Feldman, R., & Sanger, J. (2006). *The Text Mining Handbook – Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press

Fitriyani, S. P. (2017). Kinerja Algoritma Support Vector Machine dalam Mengklasifikasikan Kegiatan di Tingkat RT/RW Yang Diposting Dalam Forum Diskusi Pada Aplikasi Quue[skripsi]. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques Third-Ed* . USA: Elsevier.

Harjono, K.D. Perluasan Vektor Pada Metode *Search Vector Space. Integral Vol. 10 No.2, Juli 2005* Jurusan Ilmu Komputer, Universitas Katolik Parahyangan.

Indriyono, B. V,. (2015). Klasifikasi Jenis Buku Berdasarkan Judul dan Sinopsis menggunakan Metode Naive Bayes Classifier (Studi Kasus: STMIK Kadiri). Jurnal Sistem Informasi, 1-11.

Intan, R. & Defeng, A. (2016). Hard: Subject-Based Search Engine Menggunakan Tf-idf dan Jaccard's Coefficient. Jurnal Teknik Industri, 8(1), pp-61.

Langgeni, D. P.; Baizal Z. K. A.; Firdaus Y. (2010). *Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection*. Yogyakarta : UPN Veteran Yogyakarta.

Manalu, B. U,. (2014). Analisis Sentimen pada Twitter Menggunakan Text Mining[skripsi]. Medan: Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Martina. (2017). Performa Algoritma Support Vector Machine (SVM) dalam Analisis Sentimen Eksitensi Ahok Sebagai Gubernur DKI Jakarta pada Twitter[skripsi]. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Prasetyo, E. (2012). *Data Mining – Konsep dan Aplikasi menggunakan Matlab*. Editor oleh Nikodemus WK. Yogyakarta: ANDI

Reynaldo, P. (2016). Analisis Metode Extending Document Term dengan Algoritma K-Means untuk Meningkatkan Hasil Akurasi Algoritma Support Vector Machine pada Klasifikasi

Dokumen Teks Pendek Twitter. Naskah Publikasi Jurnal, 1-11.
Robertson, S., (2004). "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF", Journal of Documentation, Vol.60, no.5, pp.503-520
Tala, F. Z. (2003). A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia, Theses. *Institute for Logic, Languange and Computation Universiteit van Amsterdam, The Nedherlands.*
Turban, E., J.E. Aronson dan T.P. Liang. (2005). *Decision Support System and Intelligent System – 7th edition*, Terjemahan oleh Dwi Prabantini. 2005. Yogyakarta: ANDI.
Yusuf, M. (2016). Klasifikasi ketertarikan pengguna twitter di Universitas Negeri Jakarta terhadap organisasi menggunakan algoritma Naive Bayes. Naskah Publikasi Jurnal, 1-9.
Zaki, Mohammed J. & Meira, Wagner JR. (2014). *Data Mining and Analysis*. New York: Cambridge University Press