

Penerapan algoritma naïve bayes classifier untuk klasifikasi judul skripsi dan tugas akhir berdasarkan Kelompok Bidang Keahlian

Utomo Pujiyanto¹, Triyanna Widiyaningtyas², Didik Dwi Prasetya³, Bintang Romadhon⁴

1. Universitas Negeri Malang, Indonesia | utomo.pujiyanto.ft@um.ac.id
2. Universitas Negeri Malang, Indonesia | triyanaaw@um.ac.id
3. Universitas Negeri Malang, Indonesia | didikdwi@um.ac.id
4. Universitas Negeri Malang, Indonesia | bintang.arema@gmail.com

Abstrak

SISINTA UM adalah sistem informasi berbasis web yang menampilkan data berkaitan dengan skripsi dan tugas akhir di Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Malang. Berdasarkan hasil observasi terhadap fitur SISINTA, diketahui bahwa judul yang ada belum terklasifikasi berdasarkan tema penelitiannya. Tujuan penelitian ini adalah untuk menerapkan dan menguji performa algoritma Naive Bayes Classifier dalam mengklasifikasi judul skripsi dan tugas akhir, berdasarkan tema penelitian dalam KBK di Jurusan Teknik Elektro FT UM. Model pengembangan yang digunakan penelitian ini adalah model waterfall. Model ini terdiri dari lima tahapan, yaitu: (1) analisis kebutuhan sistem, (2) desain sistem dan perangkat lunak, (3) implementasi dan pengujian unit, (4) integrasi dan pengujian sistem, (5) penerapan dan perawatan produk. Pengujian performa algoritma klasifikasi yang digunakan adalah teknik K-fold Cross Validation. Berdasarkan pengujian performa penerapan algoritma Naïve Bayes Classifiers menggunakan teknik K-Fold Cross Validation, dengan jumlah tahap pengujian sebanyak 10 kali, terhadap 1103 judul skripsi dan tugas akhir, didapatkan hasil rata-rata akurasi 94%, presisi 80%, dan recall 69%.

Kata Kunci

KBK, Klasifikasi, Naive Bayes, Skripsi dan Tugas Akhir

1. Pendahuluan

SISINTA UM adalah sistem informasi berbasis web yang berfungsi untuk menampilkan data yang berkaitan dengan skripsi dan tugas akhir milik mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Malang. Halaman web SISINTA menyediakan informasi berupa daftar judul, penulis, pembimbing, dan status seminar dari skripsi dan tugas akhir milik mahasiswa. Pengguna web SISINTA dapat menjadikan daftar judul yang ada, sebagai bahan referensi untuk pemilihan tema penelitian maupun pemilihan judul skripsi dan tugas akhir.

Daftar judul pada SISINTA disajikan dalam bentuk list yang terdiri dari sepuluh judul di setiap halaman, sehingga dalam pencarian referensi judul skripsi, pengguna harus membaca daftar judul di masing-masing halaman. Disediakan pula fitur pencarian berdasarkan kata kunci, dimana sistem hanya akan menampilkan judul yang berkaitan dengan kata kunci yang dimasukkan. Namun fitur yang ada ini masih belum bisa memenuhi kebutuhan pengguna yang ingin mencari referensi judul berdasarkan tema penelitiannya. Pencarian berdasarkan tema penelitian dapat memberikan referensi judul yang lebih luas tentang tema penelitian tertentu, dibandingkan dengan pencarian yang hanya menggunakan kata kunci.

Pencarian berdasarkan tema penelitian dapat dilakukan jika judul telah dikelompokkan terlebih dahulu, untuk itu diperlukan klasifikasi judul berdasarkan kategori penelitiannya. Berdasarkan Kelompok Bidang Keahlian (KBK) yang ada di Jurusan Teknik Elektro FT UM, lingkup penelitian yang ada secara umum terbagi menjadi: lingkup pendidikan vokasi, lingkup teknik elektro, dan lingkup teknik informatika. Setiap lingkup penelitian terdiri dari tiga program payung, sehingga ada sembilan program payung penelitian, yang terdiri dari: (1) manajemen vokasi dan pendidikan kejuruan, (2) kurikulum dan pembelajaran vokasi dan pendidikan kejuruan, (3) evaluasi pembelajaran vokasi dan pendidikan kejuruan, (4) kontrol, sistem cerdas, dan robotik, (5) sistem daya dan energi, (6) teknik elektronika, pengolahan sinyal, dan telekomunikasi, (7) sistem komputer, (8) rekayasa perangkat lunak, dan (9) teknologi informasi. Klasifikasi judul dapat dilakukan dengan dua cara, yakni secara manual (dilakukan oleh manusia) atau secara otomatis (dengan bantuan komputer). Kedua cara tersebut dapat diaplikasikan dengan mempertimbangkan keperluan dan situasi yang ada di lapangan. Dokumen berbentuk fisik memerlukan tenaga manusia untuk menata dan mengelompokkannya. Sedangkan pengelompokan dokumen digital dapat dilakukan dengan secara manual dan otomatis, namun akan lebih efisien jika dikelompokkan secara otomatis.

Klasifikasi dokumen secara otomatis, dapat dibagi menjadi tiga jenis, yaitu: (1) supervised document classification (klasifikasi dokumen dengan pengawasan) dimana dalam penerapannya masih memerlukan pendampingan dari manusia, (2) unsupervised document classification (klasifikasi dokumen tanpa pengawasan) dimana penerapannya dilakukan tanpa bantuan manusia (disebut juga dengan clustering), (3) semi-supervised document classification (klasifikasi dokumen dengan semi-pengawasan) dimana pengklasifikasian masih memerlukan kontribusi manusia di beberapa bagian tertentu.

Naive Bayes Classifier (NBC) adalah salah satu dari algoritma supervised document classification yang sederhana namun efisien. Model probabilistik dari algoritma ini didasarkan pada teori Bayes. Naive Bayes Classifier telah diterapkan di berbagai bidang, seperti: kedokteran, perbankan, perpustakaan, instansi perkantoran, dll. Permasalahan klasifikasi yang sering dipecahkan dengan NBC antara lain seperti: diagnosa penyakit dan pembuatan keputusan pada prosedur perawatannya, penyaringan spam pada email clients, analisis sentimen, prediksi kelayakan calon nasabah asuransi, dll.

Penelitian ini membahas penerapan Naive Bayes Classifier untuk mengklasifikasi judul skripsi dan tugas akhir berdasarkan KBK di Jurusan Teknik Elektro FT UM. Pemilihan kategori tersebut didasarkan pada kebutuhan untuk mendapatkan referensi judul sesuai tema penelitian yang diinginkan pengguna. Dengan adanya sistem klasifikasi judul ini, diharapkan dapat mempermudah pengelompokan judul yang ada, serta meningkatkan efisiensi waktu ketika dilakukan pencarian judul skripsi berdasarkan tema penelitiannya.

2. Metode

Prosedur penelitian dan pengembangan ini mengadaptasi model pengembangan *waterfall*, menurut Sommerville (2011) model pengembangan *waterfall* terdiri dari langkah-langkah berikut:

1. Definisi Kebutuhan Perangkat Lunak

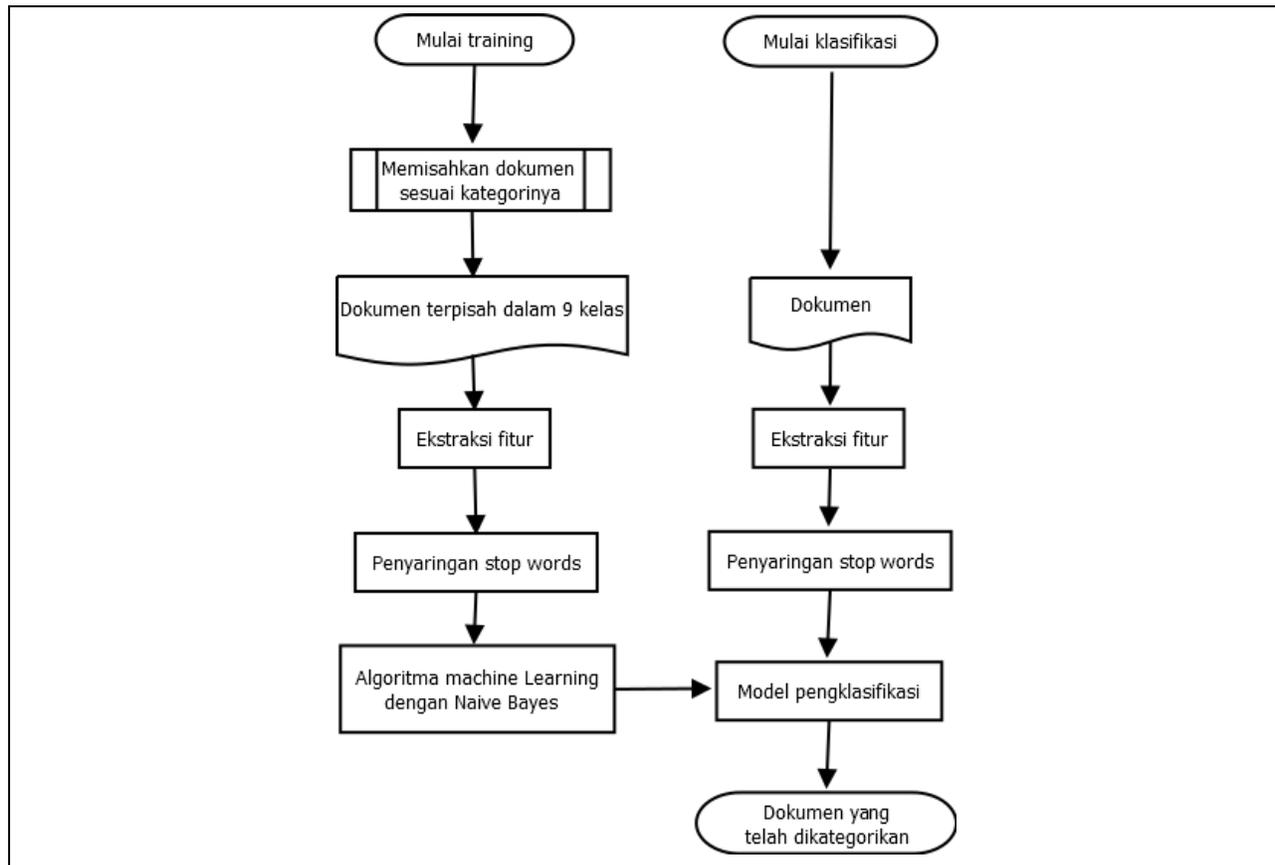
Berdasarkan observasi yang telah dilakukan pada fungsionalitas web SISINTA, dapat diketahui bahwa data berupa judul skripsi dan tugas akhir yang ada, belum dikelompokkan berdasarkan tema penelitiannya. Maka disimpulkan bahwa perlu ditambahkan fitur pengkategorian pada judul berdasarkan tema penelitiannya. Diharapkan dengan adanya klasifikasi judul, dapat memudahkan pengguna dalam mencari referensi judul skripsi atau tugas akhir berdasarkan tema yang diinginkan.

Pengembangan ini akan berfokus pada penerapan algoritma *Naive Bayes Classifier* untuk mengelompokkan judul skripsi/tugas akhir pada web SISINTA TE FT UM. Judul akan dikelompokkan ke dalam sembilan kategori berdasarkan tema penelitian di KBK di Jurusan Teknik Elektro FT UM.

Dalam tahapan *data mining* yang akan dilakukan, dibutuhkan *data training* untuk pembentukan *classifier*. *Data training* diambil dari data skripsi/tugas akhir milik Jurusan TE FT UM terdahulu. Jumlah *data training* untuk tiap kelas/kategori adalah sekitar 100 judul. Pelabelan judul yang dijadikan *data training*, ditentukan berdasarkan deskripsi tema penelitian yang tersaji pada berkas KBK Jurusan TE FT UM tahun 2015.

2. Desain Sistem dan Perangkat Lunak

Desain sistem pada penelitian ini terdiri dari pembuatan *flowchart* (alur proses), *data flow diagram* (DFD), dan *mockup* (perancangan desain antarmuka). Gambar 1 adalah *flowchart* yang menggambarkan urutan proses klasifikasi teks, yang secara umum terbagi menjadi dua tahapan, yaitu: tahap *training* dan tahap klasifikasi.



Gambar 1. Tahapan Klasifikasi Dokumen

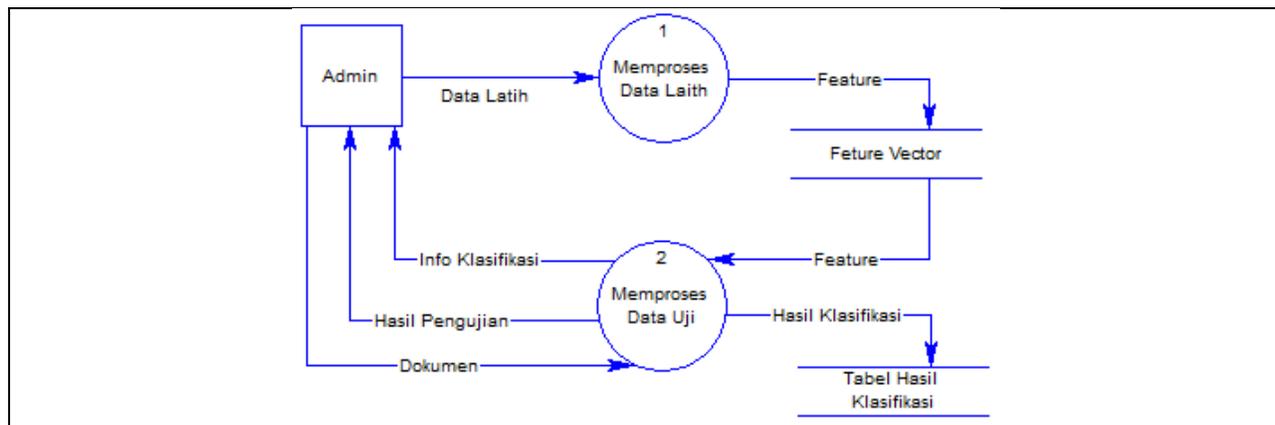
Tahapan klasifikasi dokumen pada sistem digambarkan oleh Gambar 1. Pada tahap *training*, judul skripsi dan tugas akhir dikelompokkan ke dalam sembilan kategori yang telah ditentukan, kemudian dianalisis guna membentuk aturan klasifikasi (*classifier*). Pada tahap klasifikasi, sebuah dokumen baru yang belum dikenali dikelompokkan ke dalam kategori tertentu menggunakan *classifier*.

Data flow diagram dari sistem klasifikasi judul skripsi dan tugas akhir ditampilkan pada Gambar 2.

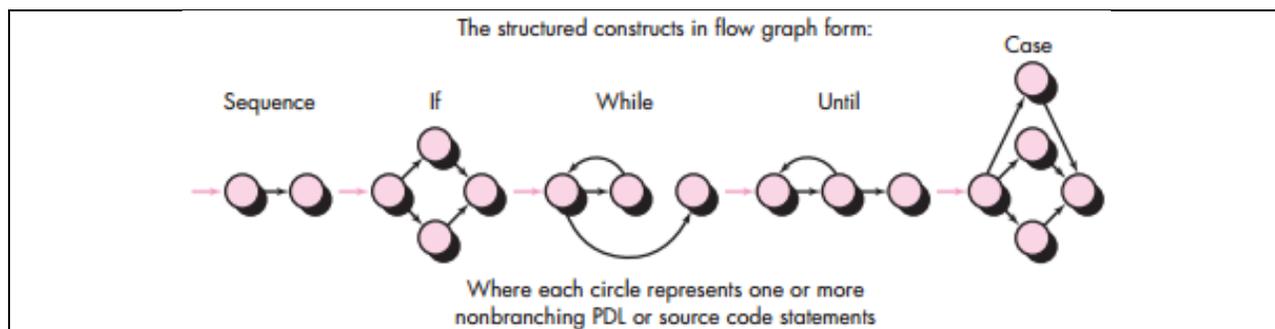
3. Implementasi dan Pengujian Unit

Pada tahap ini, dilakukan pembangunan sistem berdasarkan definisi kebutuhan dan rancangan sistem. Sistem ini dibangun dengan berbasis web, menggunakan bahasa pemrograman PHP, pada sisi tampilan menggunakan HTML5 dan CSS3. Pada sisi penyimpanan menggunakan MySQL. Sistem yang dibangun kemudian diuji sebagai unit-unit program yang terpisah. Metode pengujian unit memakai metode *basis path*, yaitu salah satu metode pengujian *white box*. Metode *basis path* memungkinkan pengujian untuk

mendapatkan ukuran kompleksitas logis dari desain prosedural dan menggunakannya untuk menentukan *basis set* dari aliran eksekusi. Pada metode ini, *control flow* dilambangkan dengan notasi *flow graph* (disebut juga dengan *program graph*) yang ditampilkan pada Gambar 3.



Gambar 2. DFD Level 1



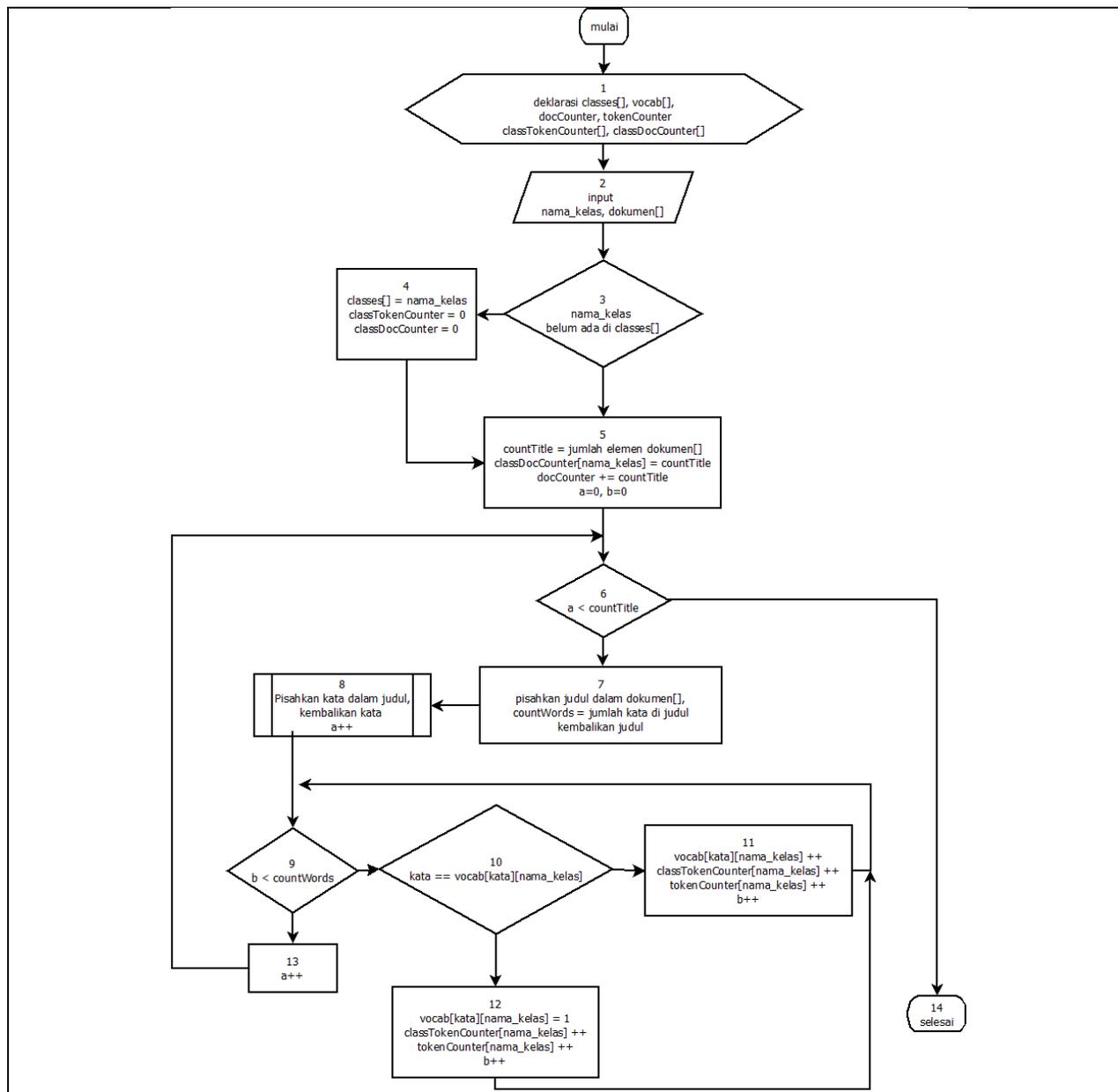
Gambar 3. Notasi Flow Graph (Sumber: Pressman 2010)

Pengujian *white box* yang dilakukan, difokuskan pada prosedur pembentukan *classifier* dan prosedur klasifikasi dokumen. Tahapan pengujian kedua prosedur tersebut masing-masing adalah sebagai berikut:

a) Menentukan diagram alir masing-masing prosedur.

Langkah pertama dalam proses klasifikasi dokumen adalah membentuk *classifier model* atau disebut juga dengan proses *training*. Dokumen yang dijadikan data *training* harus dikelompokkan dahulu secara manual, barulah diinputkan ke sistem untuk diproses. Proses *training* terdiri dari tokenisasi (pemecahan kalimat menjadi satuan kata), penghilangan *stop words* (menghilangkan kata yang tidak penting), dan pembentukan *vocabulary* (vektor atribut yang berisi daftar kata dan frekuensi kemunculannya). *Vocabulary* yang terbentuk, nantinya akan dijadikan acuan dalam perhitungan probabilitas pada proses klasifikasi

nantinya. Gambar 4 menunjukkan *flowchart* dari prosedur pembentukan *classifier model* atau *vocabulary*.



Gambar 4. Flowchart Pembentukan Classifier

Langkah selanjutnya adalah proses klasifikasi dokumen, yaitu pengelompokan dokumen berdasarkan perhitungan probabilitas menggunakan algoritma Naïve Bayes. Proses

klasifikasi dokumen (judul skripsi dan tugas akhir) terdiri dari beberapa langkah, yaitu: tokenisasi, perhitungan *prior probability*, *class probability*, dan *posterior probability*. Persamaan untuk menghitung *prior probability* ditunjukkan oleh Pers. (1).

$$\hat{P}(\omega_j) = \frac{N_{\omega_j}}{N_c} \quad \text{Pers. (1)}$$

Keterangan:

- N_{ω_j} : Total sampel pada kelas ω_j .
- N_c : Total seluruh sampel.

Rumus *class probability* adalah sbb:

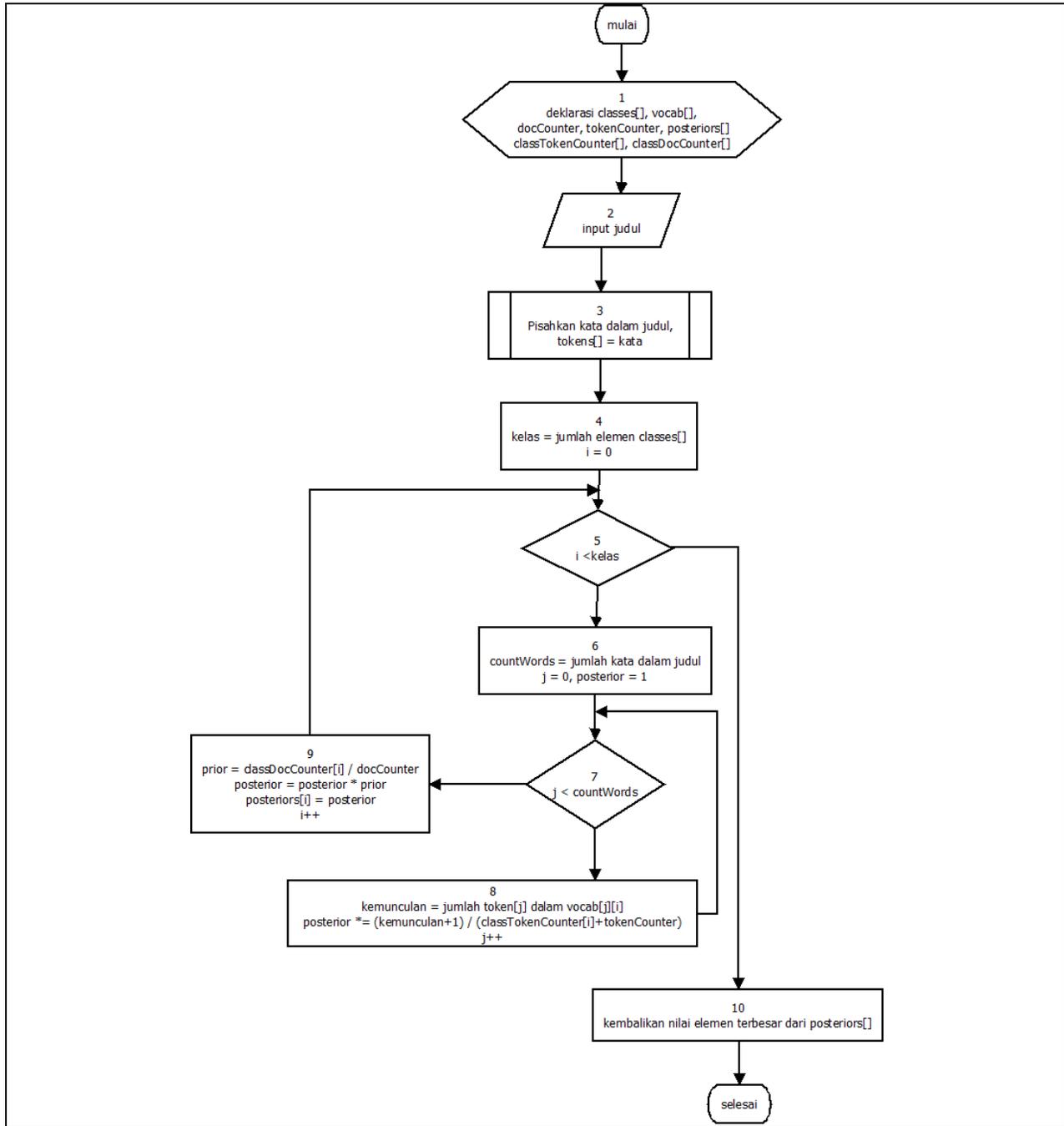
$$\hat{P}(x_i | \omega_j) = \frac{\sum tf(x_i, d \in \omega_j) + \alpha}{\sum N_{d \in \omega_j} + \alpha \cdot V} \quad \text{Pers. (2)}$$

Keterangan:

- x_i : kata dari sampel dalam *vocabulary*
- $\sum tf(x_i, d \in \omega_j)$: kemunculan kata x_i dalam kelas ω_j .
- $\sum N_{d \in \omega_j}$: Total seluruh kata pada kelas ω_j .
- α : Parameter *smoothing* tambahan ($\alpha = 1$ untuk *Laplace smoothing*)
- V : Ukuran dari *vocabulary* (jumlah kata unik dalam *vocabulary*).

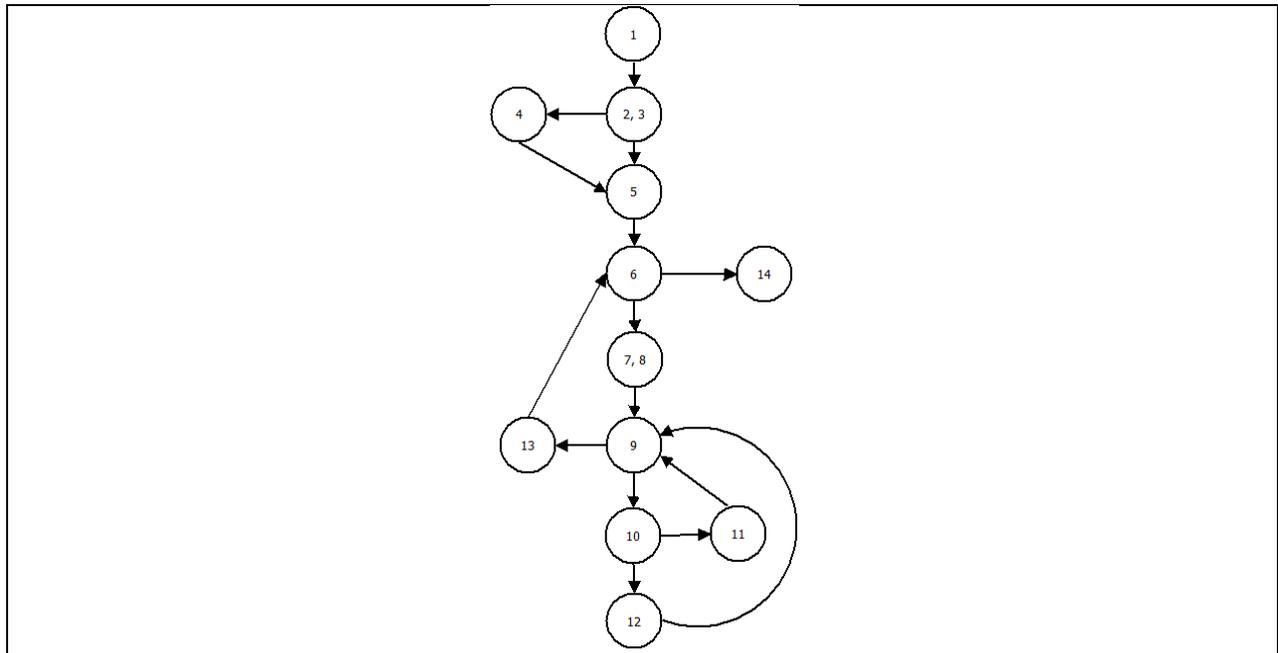
Pada rumus *class probability* di atas telah ditambahkan sebuah parameter *smoothing*, lebih tepatnya *Laplace smoothing* ($\alpha = 1$). Hal ini dilakukan untuk menghindari adanya *zero probabilities*. Contoh kasus *zero probabilities* ialah ketika ada kata dari sampel yang ternyata tidak ditemukan dalam *vocabulary*, maka *class-conditional probability*-nya akan bernilai 0. Jika *class-conditional probability* bernilai nol, maka *posterior probability*-nya juga akan bernilai 0, mengingat *posterior probability* adalah hasil kali antara *class-conditional probability* dengan *prior probability*. Penambahan parameter *smoothing* akan menjadikan suatu kata dianggap muncul minimal satu kali.

Sedangkan untuk menghitung nilai *posterior probability* adalah dengan mengalikan *prior probability* dengan *class probability*. Perhitungan dilakukan sesuai dengan jumlah kategori, kemudian diakhir proses nilai posterior akan dibandingkan dan hasil terbesar akan dijadikan acuan penentuan kategori dari dokumen yang diuji. Gambar 5 menunjukkan *flowchart* dari prosedur klasifikasi dokumen.



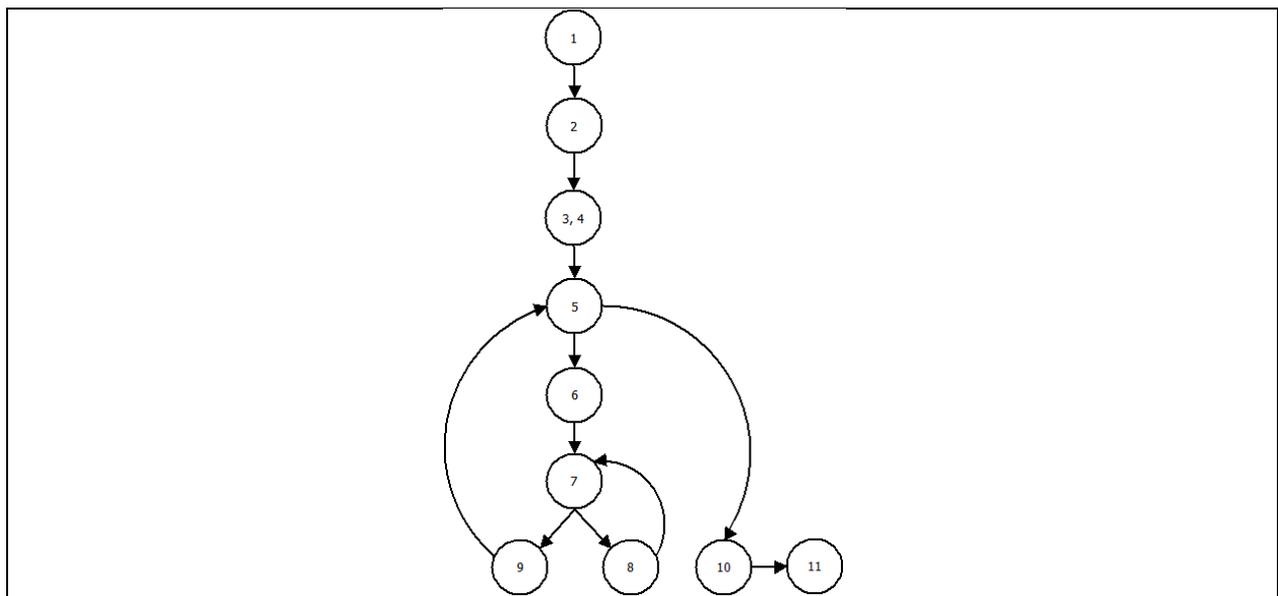
Gambar 5. Flowchart Klasifikasi Dokumen

- b) Menentukan *flow graph* berdasarkan *flow chart* yang telah dibuat.
 Gambar 6 menunjukkan *flow graph* yang terbentuk dari *flow chart* pada Gambar 4.



Gambar 6. *Flow Graph* dari Prosedur Pembentukan *Classifier*.

Gambar 7 menunjukkan *flow graph* yang terbentuk dari *flow chart* pada Gambar 5.



Gambar 7. *Flow Graph* dari Prosedur Klasifikasi Dokumen.

c) Menentukan *cyclomatic complexity* dan *independent path*.

Berdasarkan *flow graph* pada Gambar 6, maka *cyclomatic complexity* dari prosedur pembentukan *classifier* adalah berjumlah 5. Jumlah tersebut dihitung berdasarkan jumlah *region*, atau area yang dikelilingi oleh *edge*. Kemudian *independent path* yang akan diuji adalah sebagai berikut:

- *Path 1* = 1-2-3-4-5
- *Path 2* = 1-2-3-5-6-7-8-9-10-11
- *Path 3* = 1-2-3-5-6-7-8-9-10-12
- *Path 4* = 1-2-3-5-6-7-8-9-10-12-13
- *Path 5* = 1-2-3-5-6-7-8-9-10-12-13-14

Berdasarkan jumlah *region* pada *flow graph* yang ditunjukkan oleh Gambar 7, maka *cyclomatic complexity* dari prosedur pembentukan klasifikasi dokumen adalah berjumlah 3. *Independent path* yang akan diuji adalah sebagai berikut:

- *Path 1* = 1-2-3-4-5-6-7-8
- *Path 2* = 1-2-3-4-5-6-7-8-9
- *Path 3* = 1-2-3-4-5-6-7-8-9-5-10-11

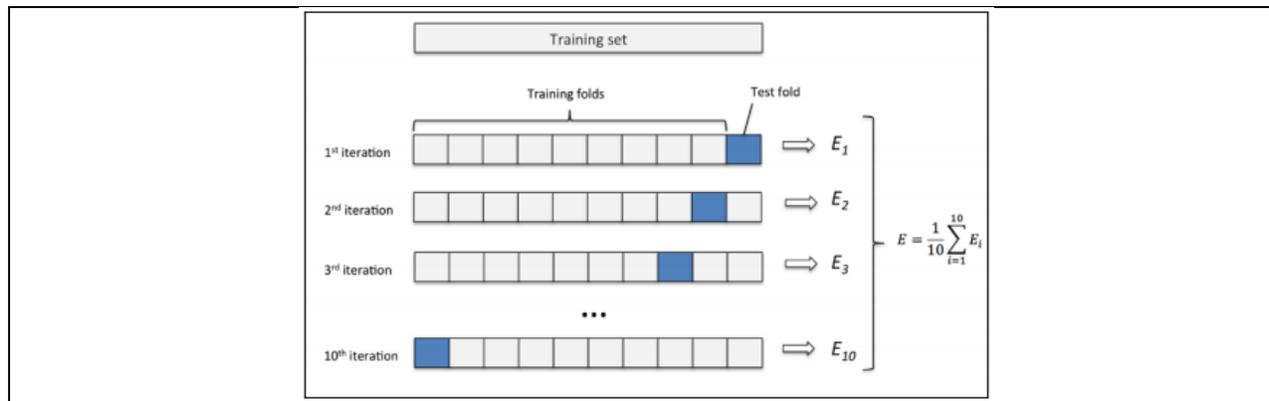
d) Menentukan *test case* berdasarkan *independent path*.

Penentuan *test case* dimaksudkan untuk menguji kebenaran logika masing-masing *independent path* minimal sekali. Biasanya dilakukan pengujian dengan memasukkan variabel yang berbeda tipe, kemudian diamati apakah terjadi *error*. Karena sistem ini hanya menginputkan variabel berjenis teks, maka untuk menguji masing-masing *path* dilakukan pencocokan hasil perhitungan dengan prosedur manual dan hasil perhitungan yang dihasilkan sistem. Kemudian sistem akan dinyatakan lulus uji jika hasil perhitungan manual sama dengan hasil perhitungan sistem.

4. Integrasi dan Pengujian Sistem

Pada tahap ini, satuan unit program diintegrasikan, kemudian diuji sebagai satu sistem yang utuh. Pengujian bertujuan untuk memastikan bahwa sistem telah memenuhi kebutuhan dan melihat kinerja dari penerapan algoritma klasifikasi.

Teknik pengujian yang dipilih untuk mengukur performa dari penerapan algoritma *Naive Bayes Classifier* adalah teknik *K-Fold Cross Validation*. Pada pengujian *K-Fold Cross-Validation*, *training data set* dibagi ke dalam lipatan sebanyak k secara acak tanpa penggantian, dimana lipatan dengan jumlah $k - 1$ digunakan sebagai *model training*, sedangkan 1 lipatan yang tersisa digunakan untuk pengujian. Prosedur ini diulang sebanyak k kali sehingga nantinya akan didapat model dan perhitungan performa sebanyak k (Raschka, 2015). Ilustrasi *k-fold cross-validation* ditunjukkan oleh Gambar 8.



Gambar 8. Ilustrasi *K-Fold Cross-Validation* (Raschka, 2015)

Jumlah *data set* yang digunakan pada percobaan ini adalah sekitar 1050, terdiri dari 100 data untuk setiap kategorinya. Pemilihan sampel yang digunakan sebagai data uji dilakukan secara acak dengan mengambil perwakilan dari setiap kategori, hal ini dilakukan agar data sampel dari masing-masing kategori tersebar merata. *Data set* diperoleh dari judul skripsi dan tugas akhir yang ada pada web SISINTA, namun tidak menutup kemungkinan diambil dari sumber lain yang relevan, mengingat beberapa kategori tema penelitian yang ada di web SISINTA masih memiliki koleksi judul yang sedikit.

Pada setiap eksperimen akan dihitung jumlah prediksi *true positive* (anggota kelas yang secara benar diprediksi sebagai anggota kelas), *true negative* (non-anggota kelas yang secara benar diprediksi sebagai non-anggota kelas), *false positive* (non-anggota kelas yang secara salah diprediksi sebagai anggota kelas), dan *false negative* (anggota kelas yang secara salah diprediksi sebagai non-anggota kelas). Perhitungan prediksi tersebut kemudian ditampung dalam *confusion matrix*. *Confusion matrix* adalah kotak berisi matriks yang menggambarkan jumlah dari prediksi *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) dari sebuah *classifier*, Gambar 9. menunjukkan ilustrasi dari *confusion matrix*.

	Diidentifikasi sebagai Kategori	Diidentifikasi sebagai Non-kategori
Kategori	True Positives	False Negatives
Non-kategori	False Positives	True Negatives

Gambar 9. Confusion Matrix

Berdasarkan data yang terekam dalam *confusion matrix*, akan dilakukan identifikasi terhadap performa penerapan algoritma klasifikasi, yakni menghitung nilai *accuracy*, *precision*, dan *recall*. *Accuracy* adalah persentase dari total dokumen yang diidentifikasi dengan benar. *Precision* adalah tingkat ketepatan jawaban yang diberikan sistem, atas informasi yang diminta oleh pengguna. Sedangkan, *recall* adalah efektifitas *classifier* dalam mengidentifikasi label positif, disebut juga dengan *True Positive Rate*.

Menurut (Sokolova & Lapalme, 2009) rumus perhitungan nilai *accuracy*, *precision*, dan *recall* dalam kasus *multiclass classification* adalah sbb:

Rumus perhitungan nilai *accuracy*:

$$\frac{\sum_{i=1}^l \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \quad \text{Pers. (3)}$$

Rumus perhitungan *precision*:

$$\frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fp_i}}{l} \quad \text{Pers. (4)}$$

Rumus perhitungan *recall*:

$$\frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fn_i}}{l} \quad \text{Pers. (5)}$$

Keterangan:

- tp_i : nilai *True Positive* dari kategori ke-i
- tn_i : nilai *True Negative* kategori ke-i
- fn_i : nilai *False Negative* kategori ke-i
- fp_i : nilai *False Positive* kategori ke-i
- l : Jumlah kategori
- $\sum_{i=1}^l \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}$: Jumlah *accuracy* dari semua kategori
- $\sum_{i=1}^l \frac{tp_i}{tp_i + fp_i}$: Jumlah *precision* dari semua kategori
- $\sum_{i=1}^l \frac{tp_i}{tp_i + fn_i}$: Jumlah *recall* dari seluruh kategori.

Dapat dijelaskan bahwa nilai *accuracy*, *precision*, dan *recall* dalam kasus *multiclass classification*, diperoleh dari nilai rata-rata *accuracy*, *precision*, dan *recall* dari semua kategori.

5. Penggunaan dan Pemeliharaan

Pada tahap ini, sistem yang dibuat telah diimplementasikan atau dioperasikan secara nyata oleh pengguna. Pemeliharaan sistem dilakukan jika ditemui *error* yang belum ditemui pada tahap pengujian sistem sebelumnya.

3. Hasil

Hasil pengujian unit menghasilkan kesimpulan bahwa sistem klasifikasi judul skripsi dan tugas akhir telah berjalan sesuai dengan algoritma yang ditentukan. Hasil dari pengujian dengan metode *basis path* pada pengujian *white box* terhadap prosedur pembentukan *vocabulary* ditampilkan pada Tabel 1.

Tabel 1. Hasil Pengujian Setiap *Path* pada Prosedur Pembentukan *Vocabulary*

No	Keterangan	Status
1	Deklarasi awal variabel dan deklarasi nama kategori	Sukses
2	Mentokenisasi judul, menginkremen jumlah kata ke-i ke <i>vocabulary</i>	Sukses
3	Mentokenisasi judul, menambahkan sebuah kata baru ke <i>vocabulary</i>	Sukses
4	Kembali ke langkah tokenisasi judul	Sukses
5	Mengakhiri proses <i>training</i> , jika semua judul telah diolah oleh sistem	Sukses

Sedangkan hasil dari pengujian dengan metode *basis path* pada pengujian *white box* terhadap prosedur pembentukan klasifikasi dokumen ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Setiap *Path* pada Prosedur Klasifikasi Dokumen

No	Keterangan	Status
1	Deklarasi variabel, menghitung <i>class probability</i> masing-masing kata	Sukses
2	Menghitung <i>posterior probability</i>	Sukses
3	Menentukan kategori suatu judul berdasarkan nilai posterior terbesar	Sukses

Tabel 3. Hasil Pengujian Performa

Tahap Pangujian	Akurasi (%)	Presisi (%)	Recall (%)
1	94.29	82.50	71.74
2	94.76	85.09	73.25
3	92.80	79.60	64.18
4	93.29	76.34	66.96
5	94.92	83.65	72.53
6	94.34	83.21	70.61
7	93.50	75.05	66.90
8	94.92	84.01	73.24
9	94.23	79.52	71.29
10	93.16	76.92	66.22
Rata-rata	94.02	80.58	69.69

Data hasil pengujian performa klasifikasi oleh sistem, disajikan dalam data kuantitatif yang merepresentasikan performa penerapan algoritma Naïve Bayes. Data hasil pengujian performa diperoleh dari 10 kali tahap pengujian menggunakan teknik *K-Fold Cross Validation*. Total sampel yang dipakai dalam sesi uji performa adalah sebanyak 1052 judul skripsi dan tugas akhir. Tabel 3 menyajikan hasil pengujian performa dari salah satu sesi pengujian. Hasil yang telah didapatkan ini dapat berubah jika dilakukan pengujian ulang, hal ini terjadi karena adanya pengacakan pada data latih dan data uji, meski demikian perbedaannya tidak akan mempunyai selisih yang signifikan.

4. Kesimpulan

Berdasarkan uraian hasil penelitian, maka dapat dikemukakan kesimpulan sebagai berikut:

- Sistem klasifikasi yang dikembangkan telah berhasil mengelompokkan judul skripsi dan tugas akhir berdasarkan Kelompok Bidang Keahlian (KBK) di Jurusan Teknik Elektro FT UM.
- Pengujian performa penerapan algoritma Naïve Bayes Classifiers menggunakan teknik K-Fold Cross Validation, dengan jumlah tahap pengujian sebanyak 10 kali, terhadap 1103 judul skripsi dan tugas akhir, didapatkan hasil rata-rata akurasi 94%, presisi 80%, dan recall 69%.

Daftar Rujukan

- Pressman, R. 2010. *Software Engineering A Practitioner's Approach 7th Edition*. New York: McGraw-Hill.
- Raschka, S. 2015. *Naive Bayes and Text Classification I Introduction and Theory*. New York: Cornell University Library. Dari Cornell University Library, (Online), (<http://arxiv.org/pdf/1410.5329v3.pdf>), diakses pada 10 Desember 2016.
- Raschka, S. 2015. *Python Machine Learning*. Birmingham: Packt Publishing Ltd.
- Sokolova, M., & Lapalme, G. 2009. A Systematic Analysis of Performance Measure for Classification Tasks. *Information Processing and Management*, (Online), 1 (45): 427-437, (<http://atour.iro.umontreal.ca/rali/sites/default/files/publis/SokolovaLapalme-JIPM09.pdf>), diakses pada 19 Januari 2017.
- Sommerville, I. 2011. *Software Engineering 9th*. Boston: Pearson Education.
- Tim Penyusun. 2015. *Kelompok Bidang Keahlian Jurusan Teknik Elektro FT UM*. Malang: Jurusan Teknik Elektro FT UM.