KLASIFIKASI CITRA DOKUMEN MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* DENGAN EKSTRAKSI CIRI *TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY*

Arif Munandar*), Achmad Hidayatno, and Teguh Prakoso

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

*)*E-mail: munandar140150@gmail.com*

Abstrak

Dokumen yang berisi infomasi mengenai berita atau sastra seringkali disimpan dalam bentuk citra. Informasi yang dimuat citra dokumen seperti kategori atau kata kunci dapat diambil dengan cara membaca isi citra dokumen secara manual. Namun cara ini menghabiskan waktu dan tidak efisien, terutama saat citra dokumen diperiksa dalam jumlah besar. Masalah ini dapat diatasi dengan cara merancang sistem yang dapat mengklasifikasikan citra dokumen berdasarkan konten yang dimuat. Suatu sistem pengklasifikasi citra dokumen berdasarkan konten telah dirancang pada penelitian ini. Sistem yang dirancang menggunakan *term frequency-inverse document frequency* sebagai ekstraksi ciri dan *support vector machine* sebagai pengklasifikasi. Ciri dari citra dokumen akan diambil dengan mengolah konten hasil dari *optical character recognition* menggunakan *term frequency-inverse document frequency*. Kategori dari citra dokumen didapatkan dengan mengolah ciri tersebut menggunakan metode *support vector machine*. Hasil yang diperoleh dari sistem ini berupa kategori yang sesuai untuk citra dokumen yang diuji berdasarkan konten yang dimuat pada citra. Parameter terbaik untuk pengklasifikasi *support vector machine* hasil dari validasi silang *grid search* adalah *kernel radial basis function* dengan $C = 2^3$ dan $\gamma = 2^{-3}$ dengan akurasi 99,6%. Sistem mampu mengklasifikasikan citra dokumen dengan ukuran data yang bervariasi dengan rata-rata akurasi 95,4%.

Kata kunci: Klasifikasi citra dokumen, Optical character recognition, Term frequency-inverse document frequency, Support vector machine, Validasi silang grid search.

Abstract

News or literature is often saved as a document image. Information such as category or keyword can be retrieved by reading the image content manually. However, it is time consuming and not efficient, especially when the number of saved image is too large. This problem can be solved by designing a system which can classify document images based on their contents. In this final project, a content-based document image classification system has been designed using term frequency-inverse document frequency as feature extractor and support vector machine as classifier. The document image features are obtained from the extracted text using term frequency-inverse document frequency. The features are processed by support vector machine classifier to obtain the document image category. The final result of the system is the correct category for document image based on its content. Grid search cross validation shows that the best parameter for support vector machine classifier is radial basis function kernel with $C = 2^3$ and $\gamma = 2^{-3}$. Grid search cross validation results show that the chosen kernel is able to achieve 99,6% accuracy. The system is able to achieve 95,4% accuracy for classifying document images with various data size.

Keywords: Document image classification, Optical character recognition, Term frequency-inverse document frequency, Support vector machine, Grid search cross validation.

1. Pendahuluan

Dokumen adalah sekumpulan tulisan yang memuat informasi. Dokumen tak hanya dalam bentuk cetak namun terdapat pula bentuk elektronik. Dokumen tersebut seringkali disimpan dalam bentuk citra dengan berbagai format seperti JPG, PNG, dan format lainnya. Namun seiring dengan bertambahnya jumlah citra yang disimpan maka semakin sulit pula untuk mengambil kembali informasi *(information retrieval)* pada citra tersebut. Cara yang sering dilakukan adalah dengan membaca citra satu demi satu, memberikan label, dan menyusun citra

dokumen secara manual berdasarkan informasi yang ada di dalamnya. Metode ini seringkali menghabiskan banyak waktu dan tidak efisien untuk berkas citra dalam jumlah yang besar.

Penelitian terhadap klasifikasi dokumen berdasarkan konten telah dilakukan. Hakim [1] dan Kuncoro [2] merancang sistem untuk mencari berbagai topik penting pada suatu artikel menggunakan metode *term frequency-inverse document frequency*. Sistem akan menyaring berbagai istilah pada artikel dan menentukan istilah yang paling tepat sebagai topik utama pada artikel tersebut.

Term frequency-inverse document frequency yang digunakan pada sistem tersebut mampu mengumpulkan istilah penting pada artikel, namun sistem akan menemukan kesulitan untuk menggolongkan suatu dokumen dengan konten yang beragam ke dalam suatu kategori tertentu, misalnya untuk menentukan apakah suatu dokumen berisi berita, cerita, pengumuman, atau konten yang lain. Cara yang dapat digunakan untuk mengenali berkas tersebut adalah dengan machine learning sebagai pengenal suatu berkas dokumen, salah satunya dengan *support vector machine*. Joachim [3] menggunakan machine learning tersebut untuk merancang sistem yang dapat mengklasifikasikan teks berita ke dalam kategori tertentu. Sistem akan mencari himpunan kata yang mewakili suatu kategori. Cara tersebut memungkinan sistem untuk menentukan kategori yang paling sesuai untuk suatu dokumen.

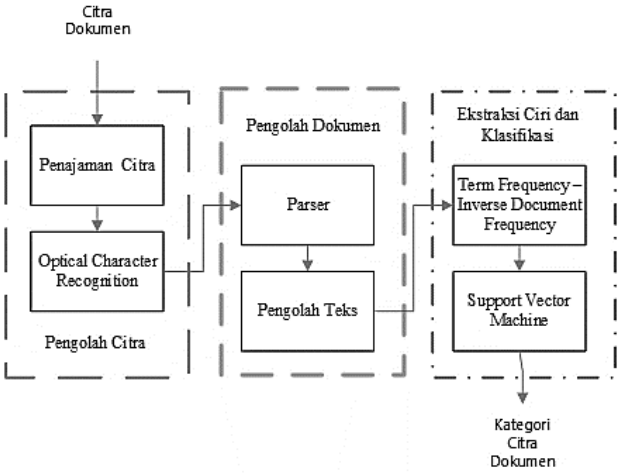
Sistem yang dirancang pada penelitian ini digunakan untuk mengklasifikasikan citra dokumen ke dalam kategori berita atau sastra berdasarkan konten yang dimuat. Masukan sistem berupa citra berisi dokumen. Berkas citra diproses dengan mesin *optical character recognition* (OCR) untuk mendapatkan konten berupa teks [4]. Metode pencirian yang digunakan adalah *term frequency-inverse document frequency* untuk mengumpulkan informasi penting dari teks [5]. *Support vector machine* digunakan untuk mengklasifkasikan citra dokumen ke dalam kategori yang sesuai sesuai dengan informasi yang dimuat.

2. Metode

2.1. Blok Diagram Sistem

Sistem yang dirancang berupa aplikasi pengklasifikasi citra dokumen berdasarkan konten yang dimuat. Blok diagram sistem pengklasifikasi citra dokumen dapat dilihat pada Gambar 1. Masukan sistem berupa citra dokumen yang berisi berita atau sastra. Konten citra dokumen diambil menggunakan blok pengolah citra dengan hasil berupa teks. Hasil dari pengolah citra diproses pada bagian pengolah dokumen dengan keluaran berupa runtun kata.

Ekstraksi ciri dan klasifikasi akan mengolah runtun kata sebagai ciri dari citra dokumen menjadi bentuk vektor menggunakan *term frequency-inverse document frequency*, kemudian mengklasifikasikan citra dokumen berdasarkan vektor ciri tersebut menggunakan *support vector machine*. Hasil akhir dari bagian ini berupa kategori yang sesuai untuk citra dokumen yang diuji. Pengguna dapat mengetahui kategori dari berbagai citra dokumen yang berbeda tanpa membaca isi dari citra dokumen satu demi satu.



Gambar 1. Diagram blok sistem pengklasifikasi citra dokumen

2.2. Sistem Pengklasifikasi Citra Dokumen

Sistem dirancang menggunakan Bahasa pemrograman Python 2.7 dengan tambahan aplikasi luar berupa Tesseract sebagai mesin OCR. Masukan sistem berupa citra dokumen. Konten citra dokumen berupa teks didapatkan menggunakan mesin OCR Tesseract. Ekstraksi ciri dan klasifikasi menggunakan *library* Sklearn pada Python. Hasil akhir klasifikasi ditampilkan pada *graphical user interface* (GUI) pada Python [6].

2.2.1. Pengolah Citra

Pengolah citra merupakan bagian yang berfungsi untuk mengambil konten dari citra dokumen berupa teks. Bagian ini tersusun atas penajaman citra dan OCR. Penajaman citra bertujuan untuk memperjelas teks pada citra dokumen agar kesalahan karakter pada OCR minimum. Penajaman citra dirancang menggunakan Python *image library* (PIL). OCR bertujuan untuk mengambil konten berupa teks dari citra dokumen yang telah ditajamkan [7]. OCR dirancang menggunakan Tesseract sebagai mesin inti dari proses OCR dengan antarmuka menggunakan *library* PIL pada Python. Hasil akhir dari bagian pengolah citra berupa teks yang akan diolah pada bagian selanjutnya.

2.2.2. Pengolah Dokumen

Pengolah dokumen merupakan bagian yang berfungsi untuk mengolah teks hasil dari pengolah citra maupun teks dari data pelatihan agar dapat digunakan oleh bagian ekstraksi ciri dan klasifikasi. Pengolah dokumen tersusun atas *parser* dan pengolah teks. *Parser* merupakan bagian yang berfungsi untuk mengolah data pelatihan dengan

bentuk XML. *Parser* dirancang agar dapat mengolah berkas XML dengan berbagai jenis *tag*.
Pengolah teks bertujuan agar konten berupa teks dari citra dokumen dapat langsung diolah pada bagian ekstraksi ciri dan klasifikasi. Pengolah teks tersusun atas tokenisasi, penghilangan *stop word*, dan *stemming*. Tokenisasi merupakan proses pemecahan teks ke dalam kata. Penghilangan *stop word* adalah proses penyaringan kata yang tidak penting menggunakan suatu daftar kata, sehingga teks hanya berisi kata penting yang mengandung informasi [8]. *Stemming* merupakan proses pengubahan suata kata menjadi bentuk dasar. Hasil akhir dari bagian pengolah dokumen berupa runtun kata.

2.2.3. Ekstraksi Ciri dan Klasifikasi

Bagian ini terdiri atas ekstraksi ciri menggunakan *term frequency-inverse document frequency* (TF-IDF) dan klasifikasi menggunakan *support vector machine* (SVM). TF-IDF mengubah runtun kata hasil dari pengolah dokumen menjadi vektor ciri sesuai dengan frekuensi kemuncul kata t pada suatu dokumen (d) maupun pada keseluruhan dokumen (D). *Term frequency* (TF) merupakan frekuensi kemunculan suatu kata t pada dokumen d yang dapat dituliskan sebagai berikut [5]:

$$tf(t,d) = \sum_{x \in d} fr(x,t) \quad (1)$$

dengan $fr(x,t)$ merupakan fungsi yang dinyatakan sebagai berikut:

$$fr(x,t) \begin{cases} 1, & jika\ x = t \\ 0, & yang\ lain \end{cases} \quad (2)$$

Inverse document frequency (IDF) merupakan bobot kata berdasarkan frekuensi kemunculan kata t pada keseluruhan dokumen pada *corpus* (D). Persamaan IDF dapat dituliskan sebagai berikut [5]:

$$idf(t,D) = log \frac{N}{1+|\{d \in D : t \in d\}|} \quad (3)$$

dengan N merupakan jumlah dokumen pada *corpus*, dan$|\{d \in D : t \in d\}|$ adalah banyaknya dokumen d pada *corpus* yang mengandung kata t.
TF-IDF didapatkan dengan mengalikan hasil TF dengan hasil IDF. Persamaan tersebut dapat dituliskan sebagai berikut [5]:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \quad (4)$$

Bobot TF-IDF suatu kata akan tinggi saat kata tersebut sering muncul pada suatu dokumen dan kata tersebut jarang ditemui pada dokumen lain. Hasil TF-IDF berupa vektor yang digunakan untuk memetakan data di ruang ciri. SVM merupakan pengklasifikasi yang bekerja berdasarkan posisi suatu data di ruang ciri. Setiap kelas data menempati bagian tersendiri yang berbeda dengan kelas lain pada ruang ciri. *Hyperplane* sebagai pemisah antar kelas dibuat berdasarkan posisi kelas tersebut.
Klasifikasi data untuk menentukan kelas bergantung terhadap posisi data tersebut dari *hyperplane*. Proses pelatihan dan klasifikasi menggunakan model SVM dilakukan dengan *inner product* antara dua vektor data ($\boldsymbol{x}$) menggunakan sutu fungsi *kernel* K. *Linear* dan *radial basis function* (RBF) merupakan fungsi *kernel* yang digunakan pada penelitian ini. Kedua fungsi *kernel* tersebut dapat dituliskan sebagai berikut [9]:

1. *Kernel Linear*

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i^T} \cdot \boldsymbol{x_j} \quad (5)$$

2. *Kernel* RBF

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp(-\gamma \left|\left| \boldsymbol{x_i} - \boldsymbol{x_j} \right|\right|^2) \quad (6)$$

Klasifikasi dilakukan menggunakan salah satu fungsi *kernel* pada fungsi pengambilan keputusan. Proses klasifikasi membutuhkan data berupa vektor penyangga ($\boldsymbol{x_{i)}}$, kelas vektor penyangga (y_i), dan pengali lagrangian (a_i) hasil dari proses pelatihan. Fungsi pengambilan keputusan untuk mengetahui kategori suatu data adalah sebagai berikut [9]:

$$f(\boldsymbol{x_d}) = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{K}(\boldsymbol{x_i} \cdot \boldsymbol{x_d}) + b \quad (7)$$
$$0 \le a_i \le C$$

Dengan a_i merupakan pengali lagrangian dengan nilai yang dibatasi oleh parameter C, yaitu *trade-off* antara margin pada *hyperplane* dengan kesalahan klasifikasi. Jika fungsi pengambilan keputusan menghasilkan nilai ≥ 1, maka data yang diujikan berada pada kelas 1. Jika fungsi pengambilan keputusan menghasilkan nilai ≤ 1, maka data yang diujikan berada pada kelas 2. Kedua kelas tersebut digunakan untuk memberikan label terhadap data dengan kategori berita atau sastra. Nilai parameter C atau γ pada fungsi *kernel* dapat dicari menggunakan metode validasi silang *grid search*.

2.3. Validasi Silang *Grid Search*

Fungsi *kernel* pada pengklasifikasi SVM mempunyai parameter berupa C untuk *kernel linear* dan RBF serta parameter tambahan berupa γ untuk *kernel* RBF. Nilai pada parameter tersebut ditentukan terlebih dahulu agar proses klasifikasi menggunakan SVM *kernel linear* atau RBF dapat menghasilkan akurasi yang tinggi. Penentuan nilai parameter pada *kernel* SVM menggunakan metode validasi silang *grid search*. Proses validasi silang *grid search* dilakukan dengan cara melatih dan menguji setiap model yang dibangun dengan suatu nilai parameter pada *kernel* tertentu menggunakan dataset yang berisi dokumen dengan berbagai kategori. Dataset akan dibagi ke dalam K bagian secara acak. $K-1$ dari seluruh dataset akan

digunakan untuk melatih model pengklasifikasi. Sisa data akan digunakan sebagai data validasi untuk menguji model yang telah dilatih tersebut. Pengujian dilakukan sebanyak *K* pengujian untuk setiap model. Data akan dirotasi untuk setiap pengujian. Hasil pengujian tersebut berupa rata-rata akurasi dari *K* pengujian. Proses dilanjutkan dengan cara yang sama menggunakan model lain dengan nilai parameter yang berbeda. Hasil akhir validasi silang *grid search* berupa rata-rata akurasi untuk setiap model. Nilai parameter *kernel* pada model yang menghasilkan akurasi tertinggi akan dipilih sebagai model pengklasifikasi dalam pengujian citra dokumen. Hal ini menandakan bahwa nilai parameter *kernel* tersebut merupakan nilai terbaik untuk model pengklasifikasi [10].

Model akhir dibangun menggunakan parameter terbaik hasil validasi silang *grid search*. Model dibangun dengan cara dilatih menggunakan seluruh dataset pelatihan dan disimpan dalam suatu berkas. Model yang telah dibangun dapat digunakan untuk menguji citra dokumen. Penggunaan model tersebut pada pengujian citra akan menghasilkan akurasi yang tinggi pada klasifikasi.

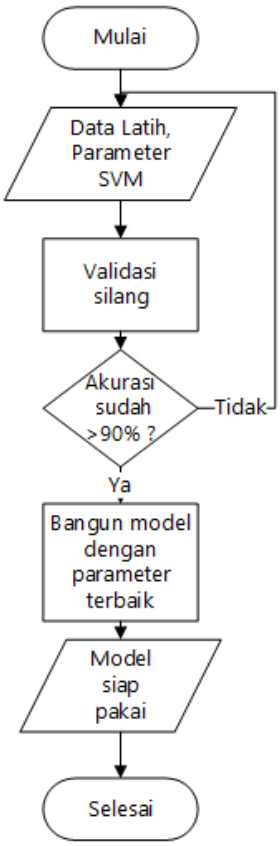
2.4. Perancangan Aplikasi Pengklasifikasi Citra Dokumen

Aplikasi pengklasifikasi citra dokumen berdasarkan konten dirancang menggunakan bahasa pemrograman Python dengan Tesseract sebagai mesin OCR. Aplikasi terdiri atas dua bagian utama, yaitu pengujian parameter dan pengujian citra. Masukan dan keluaran kedua bagian tersebut dirancang menggunakan *graphical user interface* (GUI) dengan berbagai fungsi dari modul yang terpisah.

2.4.1. Pengujian Parameter

Pengujian parameter tersusun atas validasi silang *grid search* dan pembangunan model pengklasifikasi. Proses diawali dengan memilih *kernel* yang akan diuji, yaitu *linear* atau RBF, kemudian menentukan nilai parameter *C* atau γ pada *kernel*. Dataset pelatihan yang terdiri atas kategori berita dan sastra digunakan dalam validasi silang *grid search*. Hasil dari proses ini berupa akurasi untuk setiap parameter *kernel* yang diujikan. Parameter yang menghasilkan akurasi terbaik akan dipilih untuk pembangunan model SVM.

Pembangunan model pengklasifikasi SVM terdiri atas pelatihan menggunakan dataset pelatihan, pengujian model, dan penyimpanan model. Pelatihan model pengklasifikasi SVM menggunakan parameter terbaik hasil dari validasi silang *grid search*. Proses pelatihan menggunakan keseluruhan dataset pelatihan. Model yang telah dilatih akan diujikan menggunakan data yang dipakai dalam pelatihan. Proses tersebut dilakukan hingga akurasi 100% yang menandakan bahwa proses pelatihan telah dilakukan dengan sempurna. Hasil pelatihan akan disimpan dalam bentuk model SVM dengan *kernel* tertentu. Model tersebut dapat langsung digunakan untuk mengklasifikasikan citra dokumen yang bekum dikenal.

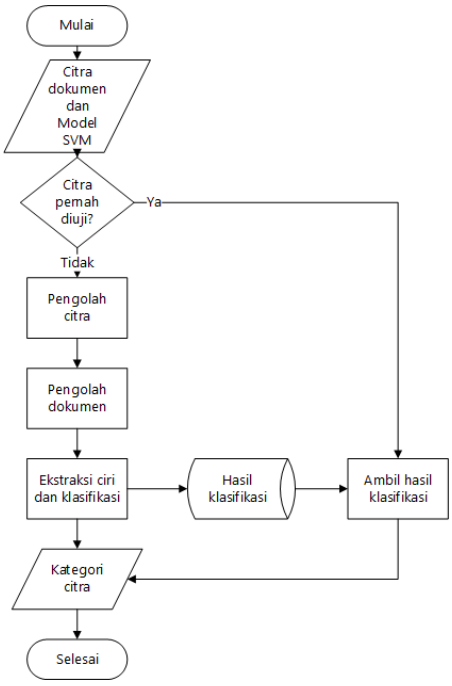


Gambar 2. Diagram alir GUI pengujian parameter

2.4.2. Pengujian Citra

Pengujian citra merupakan bagian yang digunakan untuk mencari kategori dari citra dokumen yang diuji. Bagian ini membutuhkan citra dokumen dan model SVM sebagai masukan. Model SVM berasal dari bagian pengujian parameter dengan satu nilai untuk setiap parameter pada *kernel*.

Proses pengujian citra dokumen diawali dengan memeriksa *database* hasil pengujian. Jika citra dokumen yang akan diuji terdaftar pada *database*, maka sistem akan mengambil informasi dari *database* berupa nama berkas dan kategori berkas kemudian menampilkannya pada GUI. Jika citra dokumen yang akan diuji tidak terdaftar di *database*, maka proses pengambilan konten menggunakan OCR pada pengolah citra akan dimulai. Teks hasil dari proses tersebut kemudian diolah menggunakan pengolah dokumen dengan hasil berupa runtun kata yang digunakan sebagai ciri dari citra dokumen. Runtun kata akan diolah pada ekstraksi ciri TF-IDF dengan hasil berupa vektor ciri. Vektor ciri tersebut kemudian digunakan oleh pengklasifikasi SVM untuk menentukan kategori dari citra dokumen yang diuji. Kategori citra dokumen hasil klasifikasi akan disimpan pada *database* hasil klasifikasi dan akan ditampilkan pada GUI. Proses pengujian citra dokumen dapat ditunjukkan pada Gambar 3.



Gambar 3. Diagram alir GUI pengujian citra

3. Hasil dan Analisa

Bagian ini membahas berbagai pengujian yang dilakukan pada program yang telah dirancang dan analisis hasil pengujiannya. Bagian ini dilakukan dalam tiga tahap, yaitu pengujian parameter, pelatihan dan pengujian citra. Pengujian parameter bertujuan untuk mencari nilai terbaik pada parameter *kernel* yang menghasilkan akurasi terbaik. Pelatihan merupakan tahap pelatihan dan pembangunan model pengklasifikasi dengan keseluruhan dataset pelatihan. Pengujian citra digunakan untuk mencari kategori pada citra dokumen.

3.1. Pencarian Parameter Terbaik

Pengujian parameter bertujuan untuk mencari parameter terbaik yang dapat digunakan pada *kernel* SVM menggunakan metode validasi silang *grid search*. Metode tersebut menggunakan variasi nilai parameter *kernel* untuk membangun model SVM kemudian menguji model tersebut menggunakan data latih. Hasil terbaik dari proses tersebut akan digunakan untk melatih model SVM yang akan digunakan pada pengujian citra dokumen.
Kernel yang akan diuji adalah *linear* dengan variasi parameter C dan RBF dengan variasi parameter C serta γ. Variasi parameter pada *kernel* adalah sebagai berikut:

1. *Kernel linear*
   $C = 2^{-3}, 2^{-1}, 2, 2^{3}$.
2. *Kernel* RBF
   $C = 2, 2^{3}, 2^{5}$.
   $\gamma = 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2$

Variasi parameter γ pada *kernel* RBF diterapkan untuk ketiga nilai C pada *kernel* tersebut. Hasil pengujian parameter dapat ditujukkan pada Tabel 1.

Tabel 1. Hasil pengujian menggunakan *kernel linear* dan RBF dengan variasi nilai pada parameter C dan γ

Kernel	Parameter *Kernel*		Akurasi
	C	γ	
Linear	2^{-3}	-	34,2%
	2^{-1}	-	90,7%
	2	-	87,5%
	2^{3}	-	87,5%
RBF	2	2^{-9}	34,2%
		2^{-7}	34,2%
		2^{-5}	34,2%
		2^{-3}	99,2%
		2^{-1}	98,4%
		2	85,0%
	2^{3}	2^{-9}	34,2%
		2^{-7}	34,2%
		2^{-5}	99,2%
		2^{-3}	99,6%
		2^{-1}	98,4%
		2	85,0%
	2^{5}	2^{-9}	34,2%
		2^{-7}	94,3%
		2^{-5}	96,7%
		2^{-3}	99,6%
		2^{-1}	98,4%
		2	85,0%

Tabel 1 menujukkan hasil pengujian parameter menggunakan validasi silang *grid search* untuk pengklasifikasi SVM dengan *kernel linear* dan RBF serta variasi nilai parameter pada kedua *kernel* tersebut. *Kernel linear* menghasilkan akurasi 34,2% untuk nilai C sebesar 2^{-3}. Akurasi mulai meningkat hingga nilai puncak sebesar 90,7% pada nilai C sebesar 2^{-1}. Akurasi menurun ke nilai 87,5% untuk nilai C yang lebih besar.
Tabel 1 menunjukkkan bahwa akurasi pada *kernel* RBF nilai C sebesar 2 mencapai puncak di nilai 99,2% pada γ sebesar 2^{-3}. Akurasi untuk parameter $C = 2^{3}$ mencapai puncak di nilai 99,6% pada saat γ sebesar 2^{-3}. Akurasi untuk $C = 2^{5}$ mencapai puncak di nilai 99,6% pada saat γ sebesar 2^{-3}. Variasi C dan γ akan menghasilkan akurasi puncak pada nilai γ sebesar 2^{-3} untuk setiap nilai C, sehingga nilai γ tersebut akan digunakan untuk membangun model SVM pada proses pelatihan. Nilai C yang menghasilkan akurasi tertinggi yaitu $C = 2^{2}$ dan $C = 2^{5}$ dengan puncak akurasi yang sama pada 99,6%.
Parameter C merupakan *trade-off* antara lebar margin dengan kesalahan klasifikasi. Parameter yang dicari adalah nilai terkecil pada parameter C yang menghasilkan akurasi tertinggi pada klasifikasi. Parameter C pada *kernel linear* menunjukkan akurasi maksimal sebesar 90,7% pada $C = 2^{-1}$. Parameter $\gamma = 2^{-3}$ pada *kernel* RBF menghasilkan akurasi tertinggi yang sama sebesar 99,6% untuk nilai pada parameter $C = 2^{3}, 2^{5}$. Akurasi untuk *kernel* RBF lebih tinggi dibandingkan akurasi *kernel linear* sehingga RBF

dipilih sebagai *kernel* terbaik untuk pengklasifikasi SVM. Nilai $C = 2^3$ dan $\gamma = 2^{-3}$ dipilih sebagai parameter yang akan digunakan dalam model SVM *kernel* RBF karena parameter tersebut menggunakan nilai C terkecil yang menghasilkan akurasi tertinggi, sehingga *trade-off* antara margin dengan salah klasifikasi tetap optimal. Pemilihan parameter *kernel* dengan nilai tersebut diharapkan akan menghasilkan akurasi yang tinggi pada saat pengujian citra dokumen.

3.2. Pelatihan Pengklasifikasi SVM

Pelatihan dilakukan terhadap pengklasifikasi SVM menggunakan seluruh dataset pelatihan. *Kernel* dan parameter *kernel* pada SVM diambil dari hasil validasi silang *grid search* dengan akurasi tertinggi. Dataset pelatihan yang digunakan terdiri atas 200 dokumen berita dan 200 dokumen sastra. Hasil akhir proses pelatihan adalah model pengklasifikasi yang siap digunakan untuk menguji citra dokumen yang belum dikenal.
Kernel RBF hasil validasi silang *grid search* digunakan dalam proses pelatihan. *Kernel* tersebut akan digunakan untuk melatih model SVM menggunakan keseluruhan dataset pelatihan untuk kategori berita dan sastra. Hasil pelatihan untuk model SVM tersebut dapat ditunjukkan pada Tabel 2.

Tabel 2. Hasil pelatihan untuk model SVM dengan *kernel* RBF

Kernel	Data Latih	Akurasi
RBF	Berita	100%
	Sastra	100%

Tabel 2 menunjukkan akurasi yang digunakan untuk menganalisa kinerja hasil pelatihan model pengklasifikasi SVM untuk *kernel* RBF. Hasil pelatihan pada Tabel 2 menunjukkan akurasi yang maksimal pada 100% untuk model dengan *kernel* RBF. Hal ini menujukkan bahwa pelatihan telah berjalan dengan sempurna karena semua data latih dapat dikenali oleh pengklasifikasi. Model SVM yang telah dilatih secara sempurna dapat langsung digunakan pada bagian pengujian citra untuk mengklasifikasikan citra dokumen.

3.3. Pengujian Citra

Pengujian citra pada bagian ini dilakukan terhadap citra dokumen dengan variasi pada teks. Citra dokumen diuji menggunakan 100%, 75%, 50%, dan 25% data. Pengujian menggunakan variasi tersebut bertujuan untuk menguji kinerja sistem untuk mengklasifikasikan citra dokumen dengan ukuran data yang beragam. Model yang digunakan berupa *kernel* RBF dengan nilai parameter $(C, \gamma) = (2^3, 2^{-3})$. Model SVM tersebut digunakan untuk menguji 30 citra dokumen berita dan 30 citra dokumen sastra untuk setiap variasi ukuran data.

Citra dokumen berita dianggap sebagai data positif, sedangkan citra dokumen sastra dianggap sebagai data negatif. Parameter pengujian berupa hasil benar (*true*) dan hasil salah (*false*) untuk data positif maupun negatif. Akurasi dapat didefinisikan sebagai perbandingan antara citra yang diklasifikasikan secara benar dengan keseluruhan citra yang diuji.

Tabel 3. Hasil pengujian citra dokumen menggunakan *kernel* RBF

Ukuran Data	Jumlah Citra		TP	FN	FP	TN	Akurasi
	Berita	Sastra					
100%	30	30	30	0	0	30	100%
75%	30	30	30	0	1	29	98,3%
50%	30	30	30	0	4	26	91,7%
25%	30	30	30	0	4	26	91,7%

Tabel 3 Menunjukkan bahwa kinerja program untuk mengklasifikasikan citra dokumen sudah cukup baik karena rata-rata akurasi bernilai 95,4% untuk setiap variasi ukuran data pada citra dokumen yang diuji. Hasil tersebut menunjukkan bahwa bahwa program yang dirancang mampu mengklasifikasikan citra dokumen secara benar sesuai dengan kategorinya. Model SVM dengan *kernel* RBF tersebut dapat digunakan untuk mengklasifikasikan citra dokumen lain yang belum diketahui.

4. Kesimpulan

Kernel yang sesuai untuk keperluan klasifikasi citra dokumen menggunakan *support vector machine* adalah *kernel radial basis function* dengan parameter $C = 2^3$ dan $\gamma = 2^{-3}$. Validasi silang *grid search* menggunakan *kernel* tersebut menghasilkan akurasi sebesar 99,6%. Pelatihan model pengklasifikasi SVM menggunakan *kernel* dengan parameter tersebut telah dilakukan secara sempurna dengan akurasi pelatihan sebesar 100%. Pengujian citra dokumen menunjukkan bahwa sistem dapat mengklasifikasikan citra dokumen dengan beragam ukuran data dengan rata-rata akurasi 95,4%. Hasil pengujian tersebut menunjukkan bahwa siste, pengklasifikasi citra dokumen berdasarkan konten dapat berjalan dengan baik sesuai dengan rancangan.

Referensi

[1]. A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium, and W. Muliady, "Automated Document Classification for News Article in Bahasa Indonesia based on Term Frequency Inverse Document Frequency (TF-IDF) Approach," *Int. Conf. Inf. Technol. Electr. Eng.*, pp. 0–3, 2014.

[2]. B. A. Kuncoro and B. H. Iswanto, "TF-IDF Method in Ranking Keywords of Instagram Users ' Image Captions," *Int. Conf. Inf. Technol. Syst. Innov.*, pp. 1–5, 2015.

[3]. T. Joachims, "T ext Categorization with Support Vector Machines: Learning with Many Relevant Features," *Eur. Conf. Mach. Learn.*, pp. 137–142, 1998.

[4]. R. Smith, "An Overview of the Tesseract OCR Engine," in *Ninth Int. Conference on Document Analysis and Recognition*, 2007, pp. 629–633.

[5]. G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.

[6]. D. Kuhlman, *A Python Book : Beginning Python, Advanced Python, and Python Exercises*. Massachusetts: MIT, 2013.

[7]. R. W. Smith, "The Extraction and Recognition of Text from Multimedia Documen Images," University of Bristol, 1987.

[8]. F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Universiteit van Amsterdam, 2003.

[9]. C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 43, no. 2, pp. 121–167, 1998.

[10]. R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Int. Jt. Conf. Artif. Intell.*, vol. 5, 1995.