

Perancangan Sistem Deteksi Plagiarisme Skripsi (Judul Dan Abstrak) Berbasis Matlab Menggunakan Algoritma Winnowing

Windy Pramudita¹, Berny Pebo Tomasouw^{2*}, Zeth Athur Leleury³, Monalisa E. Rijoly⁴

¹Jurusan Matematika FMIPA Universitas Pattimura, Jl. Ir. M. Putuhena, Ambon. Indonesia.

*Email: bptomasouw@gmail.com

Manuscript submitted : October 2021.

Accepted for publication : November 2021.

Doi : <https://doi.org/10.30598/tensorvol2iss2pp67-76>

Abstract: Plagiarism is an act of plagiarizing the work of others who will then acknowledge the work as one's own work without mentioning the source of the work. This research aims to create a plagiarism detection system using the winnowing algorithm in MATLAB to prevent plagiarism in the final project of the Mathematics Department students. In order to get the best k-gram value and window size that will be used in the system, a testing process is carried out between document I (100% data) and document II (80% data) by using variations in k-gram values and window sizes. The test results show that the best k-gram and window size are 12 and 4.

2010 Mathematical Subject Classification : 68W05.

Keywords: Algorithm, Matlab, Plagiarisme, Winnowing, Thesis.

1. Pendahuluan

Dalam dunia perkuliahan skripsi atau tugas akhir merupakan salah satu mata kuliah wajib yang harus diselesaikan oleh setiap mahasiswa. Perguruan tinggi (mahasiswa dan dosen) diharapkan mampu menghasilkan suatu karya ilmiah yang benar-benar hasil karyanya yang tidak memiliki unsur plagiarisme [9]. Skripsi merupakan hasil penelitian yang dibuat sendiri oleh mahasiswa. Untuk menentukan fokus penelitian maka setiap mahasiswa terlebih dahulu harus menentukan topik penelitiannya. Untuk penulisan penelitian, maka mahasiswa akan melakukan pencarian topik dengan membaca jurnal penelitian baik lokal, nasional maupun internasional, mengikuti penelitian yang dilakukan dosen, membaca tugas akhir yang telah dibuat mahasiswa atau dosen sebelumnya, serta melakukan observasi masalah yang terjadi baik di lingkup perguruan tinggi maupun diluar perguruan tinggi [3]. Untuk mendapatkan topik tugas akhir dapat dilakukan dengan beberapa cara, akan tetapi untuk mencari topik tugas akhir tidaklah mudah. Sehingga banyak mahasiswa yang kesulitan dalam mengerjakan tugas akhir tersebut karena belum mendapatkan topik penelitian yang sesuai.

Di zaman yang sangat canggih ini, teknologi sangat membantu dalam penulisan skripsi. Tidak jarang mahasiswa banyak yang langsung menyalin skripsi orang lain dengan alasan kesulitan dalam mengerjakan

tugas akhir tersebut. Tindakan tersebut sering kali disebut dengan plagiarisme. Karena dengan adanya tindakan tersebut maka dilakukan pendeteksian praktik plagiat. Pendeteksian praktik plagiat merupakan solusi yang sebaiknya dilakukan sehingga tindakan curang tersebut dapat diminimalkan untuk menghindari plagiarisme yang terus terjadi sampai saat ini.

Ada banyak algoritma yang telah digunakan pada penelitian-penelitian sebelumnya diantaranya algoritma Booyer Moore, Smith Waterman, Rabin Karp, dan *Winnowing*. Banyak algoritma yang tidak menggunakan tahap processing, yang dimana tahap tersebut berpengaruh pada akurasi similarity. *Winnowing* merupakan algoritma yang menggunakan tahap processing yang mana merupakan implementasi dari algoritma Rabin Karp. Algoritma *winnowing* dapat dihindarkan untuk mendeteksi plagiarisme karena telah memenuhi salah satu prasyarat dari algoritma deteksi plagiat yaitu *whitespace insensitivity*, yaitu dengan cara membuang karakter-karakter yang tidak relevan pada teks. Misalnya spasi, tanda baca, dan karakter lainnya, sehingga nantinya akan tersisa karakter yang hanya berupa huruf atau angka lainnya yang akan diproses lebih lanjut.

Oleh karena itu, pada penelitian ini dibuat perancangan sistem menggunakan algoritma *winnowing* yang memiliki tingkat kemiripan yang efisien yang diharapkan dapat menghasilkan persentase kemiripan teks secara cepat dan tepat serta mampu menyediakan informasi tersebut kepada seluruh mahasiswa untuk melihat tingkat kesamaan dalam penulisan. Hasil tingkat kesamaan tersebut dapat dijadikan sebagai pertimbangan dalam menentukan apakah penulisan karya tersebut dapat diterima atau ditolak sebagai suatu karya ilmiah yang baru.

1.1. Plagiarisme

Plagiarisme merupakan suatu tindakan praktik penyalahgunaan hak kekayaan intelektual milik orang lain orang dan pekerjaan itu diakui tidak sah sebagai akibat dari pekerjaan pribadi. Faktor yang dapat menyebabkan terjadinya plagiarisme pada suatu dokumen adalah kemalasan mereka sendiri, karena mereka merasa stres, memiliki keyakinan bahwa perilakunya tidak akan diketahui, dan perilakunya bukanlah hal yang salah untuk dilakukan atau berbahaya.

Adapun jenis-jenis plagiarisme [5], yaitu:

- a. Plagiarisme kata per kata, yaitu dengan menyalin setiap kata secara langsung tanpa melakukan perubahan sama sekali.
- b. Plagiarisme pengarang, yaitu mengakui suatu karya orang lain sebagai karya sendiri.
- c. Plagiarisme ide, yaitu menggunakan kembali pemikiran asli dari sebuah sumber teks tanpa mencantumkan teks sumber yang diambil.

Plagiarisme berdasarkan persentase dibagi menjadi 3, yaitu:

- i. Plagiarisme ringan, untuk tingkat kesamaannya dibawah 30%.
- ii. Plagiarisme sedang, untuk tingkat kesamaannya antara 30% hingga 70%.
- iii. Plagiarisme berat, untuk tingkat kesamaannya diatas 70%.

1.2. Text Processing

Text Processing adalah proses yang sering digunakan untuk melakukan *text mining*. Tujuan dari *text processing* yaitu untuk mengembalikan teks menjadi bahasa yang alami. Secara umum dalam tahap – tahap *text processing* [5], yaitu:

- a. *Case Folding* adalah suatu proses untuk memanipulasi teks, yang semua masukan teks tersebut akan diubah menjadi huruf kecil.
- b. *Tokenizing* adalah proses untuk memisahkan antar kata berdasarkan susunan kata. Hasil dari pemisahan kata disebut token.
- c. *Punctuation Removal* adalah penghapusan semua karakter yang unik berupa karakter tanda seru, tanda tanya, tanda koma dan sebagainya.

- d. *Stopword Removal* adalah kata-kata yang tidak deskriptif yang dapat dibuang. Contoh stopword adalah “dalam”, “yang”, “di”, “dan” dan sebagainya.
- e. *Stemming* adalah proses untuk merubah kata menjadi kata dasar.

1.3. ASCII

Nilai *hash* yang akan dicari dengan fungsi *hash* dalam algoritma *winnowing* merupakan representasi dari nilai ASCII (*American Standar Code for Information Interchange*) yang menempatkan angka numerik pada karakter, angka, tanda baca dan karakter-karakter lainnya [10].

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Gambar 1. ASCII

1.4. Metode k-gram

Metode *k*-gram adalah suatu metode yang terdapat dalam proses tokenisasi. Metode *k*-grams dapat berguna membentuk *substring* sepanjang *k* karakter dari sebuah *string*. Yang menjadi *substring* adalah kata. Semakin kecil nilai karakter *k*, maka pencarian nilai *similarity* akan semakin efektif dalam menghitung kemiripan dokumen [8].

1.5. Pembentukan Window

Nilai-nilai *hash* yang telah terbentuk pada *rolling hash*, akan dilanjutkan dennga membentuk nilai hash tersebut menjadi beberapa *window* dengan ukuran *w*. *Window* adalah pembagian atau pengelompokan beberapa nilai *hash* dengan ukuran yang akan ditentukan. Setelah *window* terbentuk, maka dilanjutkan dengan pemilihan nilai *hash* terkecil pada tiap *window* untuk dijadikan *fingerprnt* [8].

1.6. Algoritma Winnowing

Langkah-langkah dalam penerapan Algoritma *Winnowing* adalah sebagai berikut [2].

1. Membuang karakter yang tidak relevan.Menghapus semua spasi, tanda baca bahkan simbol seperti =, #, %, &, (,), -, _ \$, @, !, /,," pada kalimat.
2. Membentuk rangkaian *k*-gram. Dengan membentuk rangkaian karakter sepanjang *k* dari hasil pembuangan karakter yang tidak relevan.
3. Menghitung nilai *hash* untuk tiap *k*-gram yang telah ditentukan. Dengan melakukan perhitungan nilai *hash* dari setiap *k*-gram, maka fungsi yang digunakan untuk menghasilkan nilai hash dari rangkaian *k*-gram dalam algoritma *winnowing* adalah *rolling hash*. Setelah melakukan *roling hash* maka fungsi yang menghasilkan nilai ini disebut fungsi *hash*, sedangkan nilai yang dihasilkan disebut nilai *hash*.

Fungsi *hash* $H(c_1..c_k)$ didefinisikan sebagai berikut:

$$H(c_k) = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_k * b^{(k-k)} \tag{1}$$

Keterangan :
H = Nilai Hash

c = Nilai ASCII karakter

b = Basis (Bilangan Prima)

k = Banyak karakter

4. Membentuk *window* yang berasal dari nilai *hash*. Pembentukan *window* berasal dari nilai *hash*.
5. Langkah kelima dilakukan pemilihan *fingerprint* dari setiap *window*. Memilih nilai terkecil dari setiap *window* untuk dijadikan *fingerprint*.
6. Persamaan *Jaccard Coefficient*. Nilai *fingerprint* berasal dari algoritma *winning* yang digunakan untuk mengukur persentase kemiripan teks pada persamaan *Jaccard Coefficient*. Persamaan *Jaccard Coefficient* digunakan untuk menghitung kemiripan (*similarity*) dari kumpulan kata-kata yang telah dihitung nilai *hash*-nya. Berikut ini rumus persamaan *Jaccard Coefficient*.

$$Similarity = \frac{(\text{Jumlah fingerprint sama})}{(\text{Total seluruh fingerprint})} \times 100\% \quad (2)$$

2. Hasil dan Pembahasan

2.1. Pemilihan Nilai k -gram dan Ukuran *Window*

Tahap ini bertujuan untuk mendapatkan nilai k -gram dan ukuran *window* yang terbaik sehingga dapat digunakan dalam aplikasi sistem plagiarisme yang akan dibuat. Skema yang digunakan adalah dengan cara membentuk dokumen I yang berisikan 100% data (400 judul dan abstrak) dan dokumen II yang hanya berisikan 80% data (320 judul dan abstrak). Kedua dokumen ini akan dibandingkan, sehingga bisa diasumsikan bahwa persentase kemiripan antara kedua dokumen tersebut adalah sebesar 80%. Dalam proses perbandingan ini akan digunakan variasi nilai k -gram dan juga variasi ukuran *window*. Nilai k -gram dan ukuran *window* yang dipilih adalah nilai k -gram dan ukuran *window* yang memberikan hasil yang sama atau mendekati 80%. Selanjutnya akan dilanjutkan dengan perhitungan menggunakan k -gram 5 sampai 12 dan langsung mengambil k -gram 15 dan 20 untuk melihat nilai persentase apakah semakin naik atau bahkan menurun jika menggunakan k -gram tersebut. Dilanjutkan dengan ukuran *window* yang di ambil yaitu 4, 5 dan 6.

Tabel 1. Hasil Perbandingan dengan ukuran *window* sama dengan 4

k -gram	5	6	7	8	9	10	11	12	15	20
Persentase (%)	86.42	83.92	82.61	81.70	81.01	80.58	80.23	79.89	79.26	79.19

Tabel 2. Hasil Perbandingan dengan ukuran *window* sama dengan 5

k -gram	6	7	8	9	10	11	12	15	20
Persentase (%)	82.61	82.61	81.70	81.01	80.58	80.23	79.89	79.26	79.19

Tabel 3. Hasil Perbandingan dengan ukuran *window* sama dengan 6

k -gram	7	8	9	10	11	12	15	20
Persentase (%)	79.29	81.70	81.01	80.58	80.23	79.89	79.26	79.19

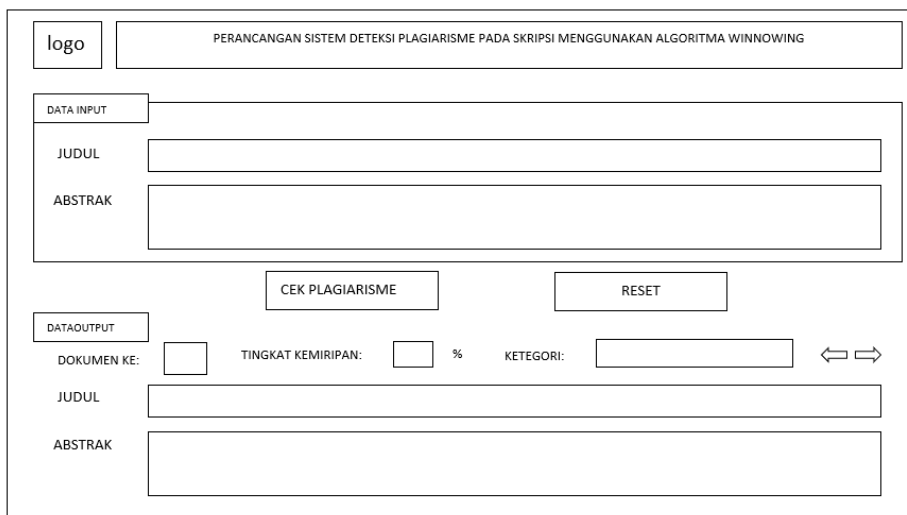
Berdasarkan uji coba di atas, diperoleh nilai persentase yang mendekati 80% terdapat pada k -gram = 11 dan k -gram = 12 dan ukuran *window* = 4, 5, dan 6 dengan nilai presentase sebesar 80.23 dan 79.89%. Dapat dilihat dari hasil tingkat plagiarisme yang lebih mendekati 80% yaitu k -gram = 12. Untuk pengujian *window* maka diperoleh semakin banyaknya ukuran *window* dapat menyebabkan sistem berjalan lama maka ukuran *window* yang paling efisien untuk menghitung tingkat kemiripan maka yang akan digunakan yaitu ukuran *window* = 4.

2.2. Implementasi

Implementasi merupakan tahap pembuatan sistem untuk mengetahui apakah sistem yang dirancang akan menghasilkan sistem yang diinginkan. Pembahasan mengenai implementasi sistem adalah sebagai berikut.

2.2.1. Design Sistem

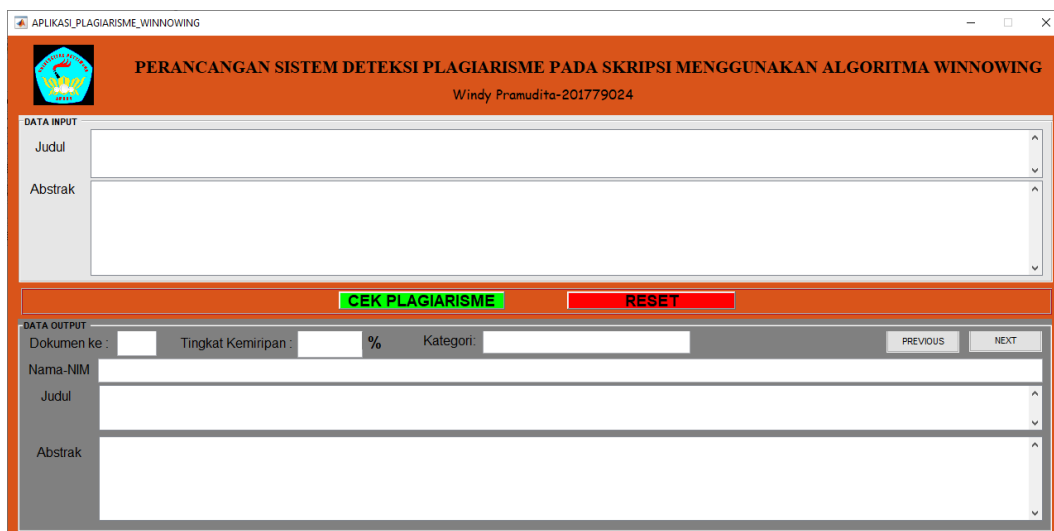
Design Sistem Perancangan sistem untuk mendeteksi suatu plagiarisme dalam skripsi sangat diperlukan karena dengan adanya perancangan tersebut maka dapat mempermudah dalam membentuk alur dari sistem yang akan dibangun. Sistem yang akan dibangun merupakan sistem yang berbasis aplikasi yang dimana akan dijalankan di aplikasi GUI MATLAB.



Gambar 2. Perancangan Sistem Awal

2.2.2. Perancangan Antar Muka

Untuk dapat melihat tingkat kemiripan judul dan abstrak skripsi mahasiswa jurusan matematika FMIPA UNPATTI maka pada penelitian ini telah dirancang suatu program yang dibuat dengan menggunakan *toolbox* yang telah disediakan MATLAB, sehingga prosesnya menjadi lebih mudah dengan menggunakan *Graphical User Interface Design (GUIDE)*. Dibawah ini adalah tampilan antar muka dari rancangan program tersebut.



Gambar 3. Tampilan Program

Pada Gambar 3. digunakan untuk menginput data untuk mendeteksi plagiarisme pada skripsi (judul dan abstrak). Yang mana proses awal akan dilakukan dengan cara menginput data yang akan diuji. Setelah penginputan data tersebut klik tombol "CEK PLAGIARISME" untuk memulai proses hitung kemiripan pada database. Untuk lebih jelasnya dapat dilihat pada tabel berikut yang merupakan keterangan dari fungsi-fungsi tombol yang terdapat pada tampilan utama program.

Tabel 4. Fungsi Tombol pada Program

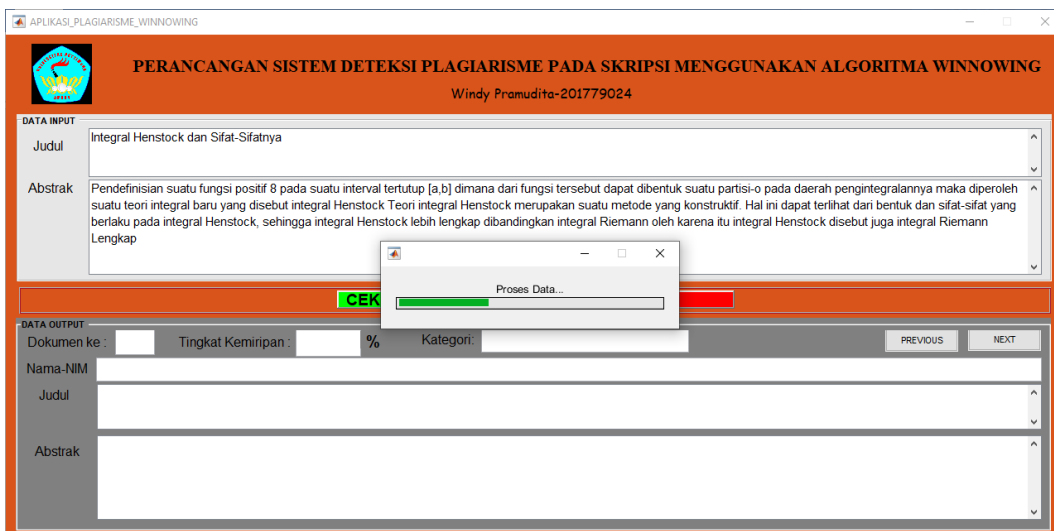
Tombol	Fungsi
Data Input	Ada 2 kotak kosong pada bagian ini yang dimana isinya berupa judul dan abstrak yang akan hitung kemiripannya
Cek Plagiarisme	Untuk menghitung dan menampilkan tingkat kemiripan dari Data Input (3 data dengan tingkat kemiripan yang paling tinggi)
Reset	Untuk mengatur ulang form program, sehingga form tersebut dapat diisi dengan data baru
Data Output	Ada 4 tampilan yang akan muncul yaitu dokumen keberapa, tingkat kemiripan, nama-NIM, judul, dan abstrak yang memiliki kemiripan yang paling tinggi dengan membaca hasil input dari judul dan abstrak. Pola input ini diperoleh dari pengisian data <i>input</i>
Previous	Untuk melihat data <i>output</i> sebelumnya
Next	Untuk melihat data <i>output</i> setelahnya

2.3. Pengujian Sistem

Pengujian sistem dilakukan guna untuk melihat apakah sistem yang telah dibangun sesuai dengan perancangan sistem yang dilakukan sehingga sistem tersebut mencapai tujuan yang diinginkan. Berikut ini dapat dilihat contoh penggunaan program dengan menggunakan data judul dan abstrak dari salah satu skripsi mahasiswa atas nama Lexy Jan Zen Sinay-1368100001.

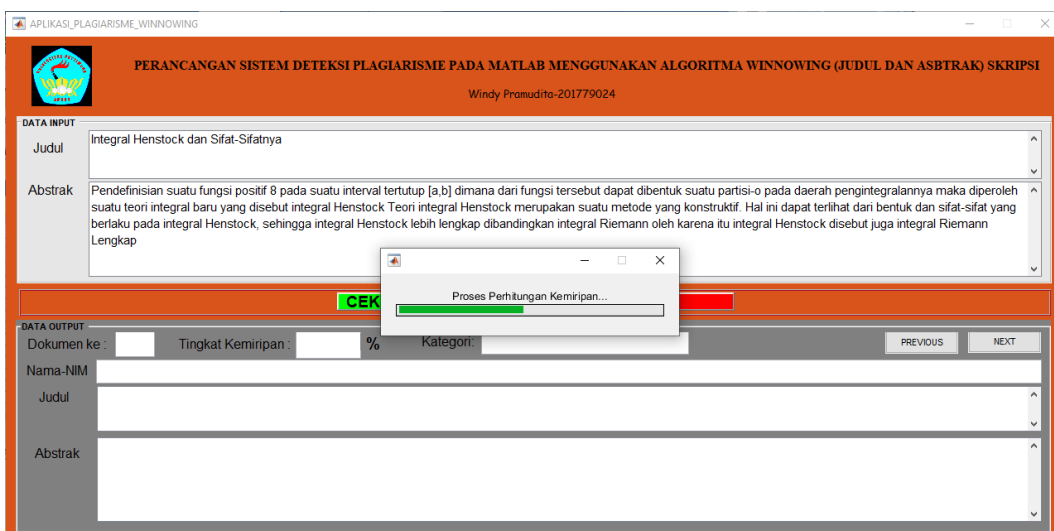
Gambar 4. Memasukkan Judul dan Abstrak Mahasiswa

Pada Gambar 4. data mahasiswa diinput guna untuk mengecek tingkat kemiripan dari data tersebut sehingga dapat diketahui data tersebut masuk di kategori plagiarisme rendah, sedang atau tinggi. Setelah memasukkan judul dan abstrak mahasiswa selanjutnya klik tombol CEK PLAGIARISME pada program.



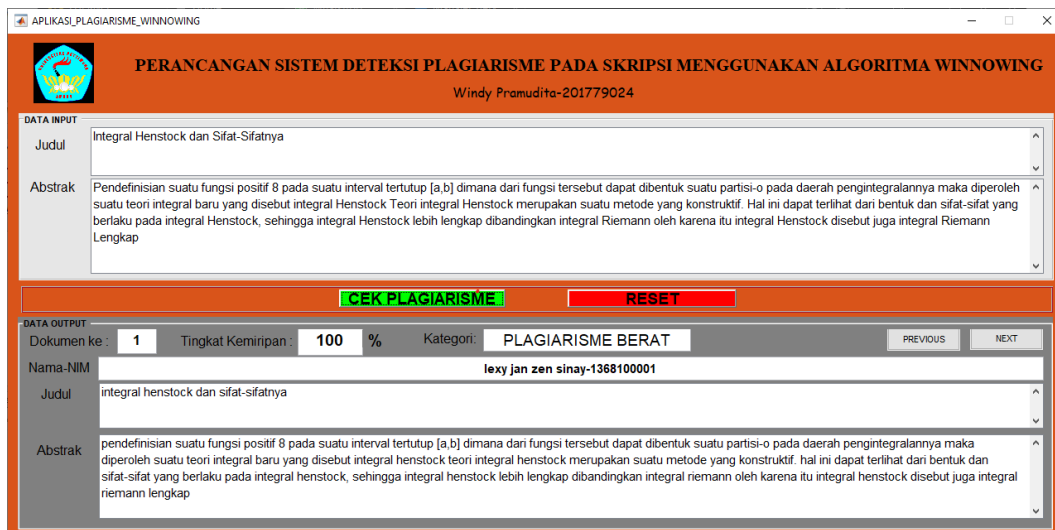
Gambar 5. Text Processing

Pada Gambar 5. setelah mengklik tombol CEK PLAGIARSME, sistem akan menjalankan *Text Processing* dari data yang di uji untuk dilakukan pembandingan dengan database pada sistem.



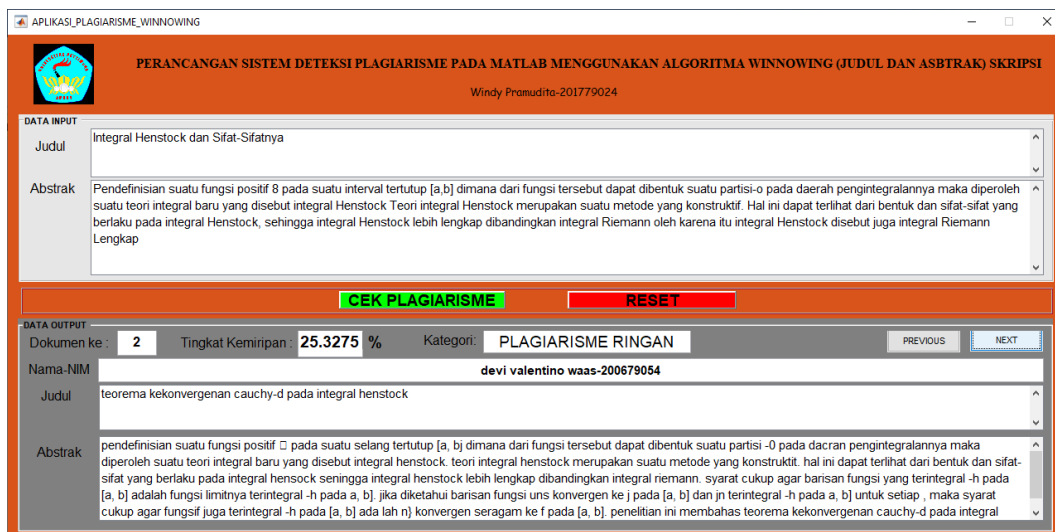
Gambar 6. Proses Perhitungan Kemiripan

Setelah *Text Processing* selesai berjalan, maka sistem melanjutkan perhitungan kemiripan data yang terlihat pada Gambar 6. Sistem akan menampilkan tingkat kemiripan dengan perhitungan dari algoritma winnowing yang akan muncul pada panel data output sebagai berikut.



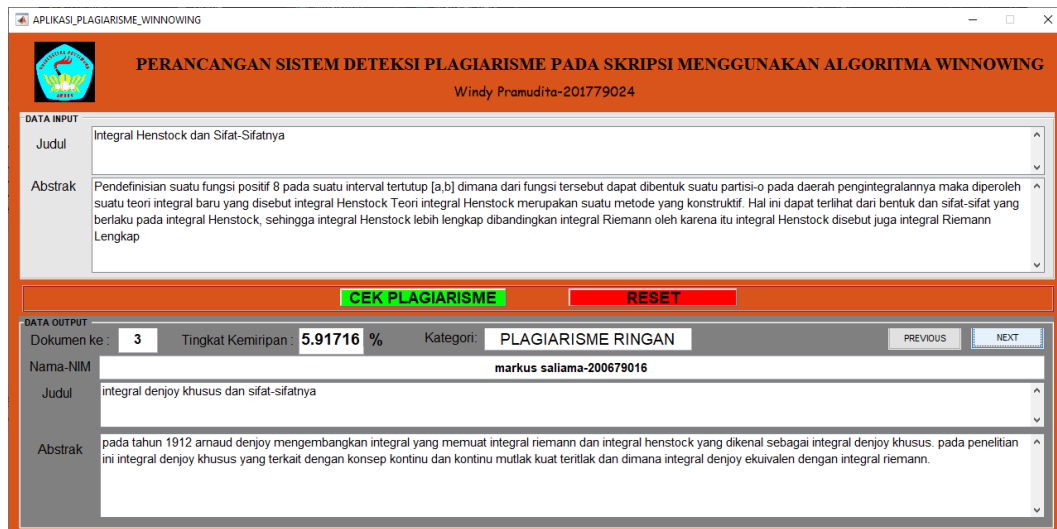
Gambar 7. Hasil Tingkat Kemiripan Data Ke-1

Pada Gambar 7. terlihat bahwa data *input* merupakan data yang sama sehingga untuk dokumen ke-1 memiliki tingkat kemiripan sebesar 100% sehingga dari pendeteksian kemiripan data uji dengan *database* pada sistem maka dinyatakan data tersebut masuk pada kategori Plagiarisme Berat.



Gambar 8. Hasil Tingkat Kemiripan Data Ke-2

Pada Gambar 8. terlihat bahwa data *input* untuk dokumen ke-2 memiliki tingkat kemiripan sebesar 25.3275% sehingga dari pendeteksian kemiripan data uji dengan *database* pada sistem maka dinyatakan data tersebut masuk pada kategori Plagiarisme Ringan.



Gambar 9. Hasil Tingkat Kemiripan Data Ke-3

Pada Gambar 9. terlihat bahwa data *input* untuk dokumen ke-3 memiliki tingkat kemiripan sebesar 5.91715% sehingga dari pendeteksian kemiripan data uji dengan *database* pada sistem maka dinyatakan data tersebut masuk pada kategori Plagiarisme Ringan.

3. Kesimpulan

Berdasarkan hasil pengujian dari dokumen I yang berisikan 100% data (400 judul dan abstrak) dan dokumen II yang hanya berisikan 80% data (320 judul dan abstrak) dengan menggunakan variasi nilai *k*-gram dan ukuran *window* maka diperoleh bahwa nilai *k*-gram dan ukuran *window* yang terbaik adalah 12 dan 4. Selanjutnya, sistem deteksi plagiarisme yang dibuat menggunakan kedua nilai ini telah diujicobakan dan berjalan dengan baik.

4. Referensi

- [1] Alamsyah, N. (2017). Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi dengan Algoritma Winnowing. *Technologia: Jurnal Ilmiah*, 8(4), 205-213.
- [2] Alamsyah, N. (2017). Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi. *Technologia: Jurnal Ilmiah*. Diakses 5 Mei 2021, <https://Doi.Org/10.31602/Tji.V8i3.1116>
- [3] Astuti, W. (2017). Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (KMP). *ILKOM Jurnal Ilmiah*, 9(2), 167-172.
- [4] Chandra, J. E. H., & Naga, D. S. (2018). Plagiarisme Abstrak Menggunakan Algoritma Winnowing Dan Synsets. *Jurnal Ilmu Komputer dan Sistem Informasi*, 6(2), 121.
- [5] Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp Untuk Pendeteksian Plagiarisme Pada Dokumen Tugas Mahasiswa. *Juita?: Jurnal Informatika*. Diakses 3 Mei 2021, <https://Doi.Org/10.30595/juita.V7i1.4063>
- [6] Istiana, P. (2014). Perpustakaan Dan Plagiarisme. "Workshop Literasi Informasi Bagi Pustakawan", 14 Mei 2013, Universitas Sanata Dharma.
- [7] Pradnya, W. M. (2020). Implementasi Algoritma Winnowing Pada Sistem Penilaian Otomatis Jawaban Esai Pada Ujian Online Berbasis Web. *Jurnal Teknik Komputer*, 6(2), 169-175.
- [8] Ridho, M. (2013). Rancang Bangun Aplikasi Pendeteksian Penjiplakan Dokumen Menggunakan Algoritma Biword Winnowing (Doctoral dissertation, Universitas Islam Negeri Sultan Syarif

Kasim Riau).

- [9] Sunardi, S., Yudhana, A., & Mukaromah, I. A. (2017). Perancangan aplikasi deteksi plagiarisme karya ilmiah menggunakan algoritma winnowing.
- [10] Surahman, A. M. (2015). Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C Dan Pascal Dengan Menggunakan Algoritma Rabin-Karp. *Justin (Jurnal Sistem Dan Teknologi Informasi)*, 1(1), 50-55.
- [11] Tjiawi, A. P., Herwindiati, D. E., & Hiryanto, L. (2017). Perancangan Aplikasi Pendeteksi Tingkat Kesamaan Antar Dokumen Dengan Algoritma Winnowing. *Computatio: Journal of Computer Science and Information Systems*, 2(1), 36-44.
- [12] Wibowo, A. (2012). Mencegah Dan Menanggulangi Plagiarisme Di Dunia Pendidikan. *Kesmas: National Public Health Journal*. Diakses 2 Juni 2021, <https://doi.org/10.21109/Kesmas.V6i5>