

## **MULTI-ASPECT SENTIMENT ANALYSIS TERHADAP NVIDIA RTX DENGAN SUPPORT VECTOR MACHINE DAN WORD EMBEDDINGS**

IGD Raditya Wibhawa MN<sup>1</sup>, Aniq Atiqi Rohmawati<sup>2</sup>, Yuliant Sibaroni<sup>3</sup>

Universitas Telkom, Bandung

radityawibhawa@student.telkomuniversity.ac.id<sup>1</sup>, aniqatiqi@telkomuniversity.ac.id<sup>2</sup>, yuliant@telkomuniversity.ac.id<sup>3</sup>

### Abstrak

Tren pembelian Graphic Processing Unit(GPU) terutama seri RTX semakin meningkat dalam kurun waktu beberapa bulan ini, banyak orang berlomba-lomba membuat PC untuk meningkatkan kinerja dalam kondisi WFH (Work From Home) selama terjadi pandemi. Peningkatan penggunaan GPU seri RTX mengundang pendapat dari berbagai kalangan yang menciptakan pro dan kontra kalangan PC enthusiast, terutama pada beberapa platform media sosial. Kemunculan pro dan kontra merupakan suatu fenomena yang dapat diteliti untuk melakukan ekstraksi emosi terhadap komentar dan opini yang mengandung pro dan kontra tersebut dengan menggunakan metode Analisis Sentimen sehingga diketahui opini positif, negatif dan netral yang dapat dijadikan sebagai rujukan untuk pengembangan GPU dan referensi bagi pengguna GPU. Metode Analisis Sentimen terdiri dari tahapan proses yaitu tahap pertama adalah preprocessing yang terdiri dari Noise Removal, Tokenizing, Stopword Removal selanjutnya proses kedua yaitu Ekstraksi Fitur dengan Word Embeddings, dan tahap ketiga Support Vector Machine. Penelitian ini bertujuan untuk menganalisa performansi model dalam mengklasifikasikan sentimen sebuah teks dengan model SVM. Berdasarkan Analisis Performansi pada model yang dibuat, SVM kernel Linear menunjukkan hasil akurasi sebesar 64% pada dataset Performance, 63% pada dataset Ray Tracing, dan 65% pada dataset DLSS. Sedangkan kernel rbf pada penelitian ini menunjukkan hasil yang kurang memuaskan meskipun telah dilakukan perubahan parameter pada Gamma dan C.

**Kata kunci:** *support vector machine, preprocessing, word embeddings*

### I. PENDAHULUAN

Tren pembelian *graphics processing unit* (GPU) terutama seri *Ray Tracing Texel eXtreme* (RTX) kian meningkat. Dengan kondisi *Work From Home* (WFH) banyak orang berlomba-lomba membeli *Personal Computer* (PC) untuk meningkatkan kinerja selama terjadi pandemi. Peningkatan terhadap penggunaan GPU seri RTX ini tentunya mengundang pendapat dari berbagai macam kalangan. Hal ini menciptakan pro dan kontra di kalangan PC *enthusiast*. Beberapa komentar muncul, terutama pada *platform* media sosial. Dari komentar tersebut kita dapat melihat tingkat emosional orang-orang yang berpendapat terhadap Nvidia RTX yang sangat laku di pasaran. Hal ini dapat diteliti lebih lanjut dengan *sentiment analysis*. Metode *sentiment analysis* ini dapat lebih mudah membedakan mana opini yang negatif, mana yang positif, dan mana yang netral.

Tahapan yang perlu dilakukan untuk melakukan *sentiment analysis* ini yang pertama adalah *preprocessing* yang terdiri dari *noise removal*, *tokenizing*, *stopword removal*. Kemudian dilanjutkan dengan proses yang kedua, yaitu ekstraksi fitur dengan *word embeddings*. Terakhir menggunakan model *support vector machine* (SVM) terhadap hasil dari ekstraksi fitur *word embeddings* yang telah dihasilkan sebelumnya.

Ada beberapa riset yang terkait dengan *multi-aspect sentiment*, antara lain adalah *Machine Learning-Based Sentiment Analysis for Twitter Accounts*[1], *Sentiment analysis on twitter data using support vector machine*[2], *Sentiment Analysis of Twitter Data Using Naive Bayes Algorithm*[3], *Sentiment analysis on Twitter about the use of city public transportation using support vector machine method* [4]. Pada riset tersebut, model *machine learning* yang sering dipakai

untuk *sentiment analysis* adalah *naïve Bayes* dan *support vector machine*. Beberapa *feature extraction* yang digunakan adalah *bag-of-words*, *tf-idf*, *word embeddings*, *lexicon based approach*. Kelemahan dari riset ini adalah, NLP biasanya hanya terbatas menggunakan *naïve Bayes* dan *support vector machine classifier* karena rata-rata *accuracy* yang dihasilkan tinggi bila dibandingkan *classifier* yang lain, sehingga penggunaan model *machine learning* yang lain jarang ditemukan. Potensi riset ini dapat dikembangkan lagi analisis sentimennya ke beberapa tipe GPU tipe RTX tertentu karena penelitian ini merujuk pada RTX secara umum.

Analisis performansi model dalam mengklasifikasikan sentimen sebuah teks dengan model SVM dibuat berdasarkan riset terhadap beberapa *paper*, serta banyaknya pendapat di beberapa *platform* sosial media mengenai faktor kenaikan penggunaan dan pembelian *Nvidia RTX Series*. Metode *word embeddings* digunakan untuk merepresentasikan sejumlah kata dengan baik dalam bentuk vektor dan berujung pada performansi yang lebih baik, serta penggunaan model SVM dapat menghasilkan hasil yang baik dengan menerima *input*/fitur berupa *vector* bila dibandingkan dengan model klasifikasi lainnya untuk *multi-aspect sentiment*[5].

### II. KAJIAN TEORI

Beberapa penelitian dan kajian teori sebagai perbandingan dan penunjang penelitian ini, berhubungan dengan Analisis Sentimen, namun menggunakan beberapa *Text Preprocessing*, *Feature Extraction*, dan pemodelan *Machine Learning* yang berbeda.

### A. Document Level Sentiment Classification

Pada penelitian [6] menggunakan *dataset* yang telah ditentukan sentimen positif dan negatif. Pemodelan *machine learning* yang digunakan adalah *support vector machine* dan *naïve Bayes*. *Text preprocessing* yang digunakan adalah *n-grams* yaitu *bigrams*. Hasil yang didapatkan menunjukkan *accuracy* dari *naïve Bayes* memiliki skor yang lebih tinggi dibandingkan dengan *support vector machine*.

Pada penelitian [4] menggunakan *dataset* yang di *crawl* dari *twitter* berhubungan dengan angkutan publik dan telah ditentukan sentimen positif dan negatifnya. Pemodelan *machine learning* yang digunakan adalah *support vector machine*. *Text preprocessing* yang digunakan adalah POS (*Part of Speech*) *tagging* dengan HMM dan *rule based POS tagging*, kemudian di *weighting* dengan *tf-idf*.

Pada penelitian [7], [8], [9] menggunakan *dataset* yang diambil berdasarkan masing masing topik dimana telah ditentukan sentimen positif dan negatifnya. Penelitian [7] menggunakan Bengali *dataset* ABSA, Penelitian [8] menggunakan *dataset* berhubungan dengan *smartphone*, penelitian [9] menggunakan *dataset* yang berhubungan dengan *self-driving cars*. Pemodelan *machine learning* yang digunakan oleh ketiganya adalah *support vector machine*. *Text preprocessing* pada [7] menggunakan *n-grams*, sedangkan pada [8] menggunakan *Part-Of-Speech* (POS) *tagging*. *Feature extraction* yang digunakan ketiganya adalah *tf-idf*.

Pada penelitian [10], [11], [12] menggunakan *dataset* masing masing topik, sentimen positif dan negatif telah ditentukan. Penelitian [10] menggunakan *dataset* yang di *crawl* dari *Twitter* berkaitan dengan COVID-19, penelitian [11] menggunakan *Cornell Movie Review Dataset*, penelitian [12] menggunakan *dataset review google play store* pada aplikasi by.U. Pemodelan *machine learning* yang digunakan [10] dan [12] adalah *support vector machine*, sedangkan penelitian [11] menggunakan *k-nearest neighbor*. *Feature extraction* yang digunakan oleh ketiganya adalah *tf-idf*.

Pada penelitian [13] menggunakan *dataset* yang di *crawl* dari *Twitter* yang berhubungan dengan *brimob* dan *mako brimob*, sentimen positif dan negatif ditentukan dengan cara manual. Pemodelan *machine learning* yang digunakan adalah *support vector machine* dan *naïve Bayes*. *Text preprocessing* yang digunakan adalah *tokenization*, *text normalization*, dan *stopword removal*, tanpa menggunakan *feature extraction* manapun. Hasil akurasi dan *classification report support vector machine* sedikit lebih tinggi dibandingkan dengan *naïve Bayes*.

Pada penelitian [14] menggunakan *dataset* dari *Kaggle* yang berisi *review mobile phone* sebanyak 400.000 data, sentimen positif dan negatif di *labelling* secara manual. Pemodelan *machine learning* yang digunakan adalah *support vector machine*, *naïve Bayes*, *decision tree*, *random forest*. Menggunakan *tf-idf* setelah melakukan *text preprocessing*. Hasil yang paling tinggi didapatkan oleh *support vector machine* dibandingkan dengan *classifier* lainnya.

### B. Analisis Sentimen

Analisis sentimen merupakan bidang studi yang menganalisa tentang opini masyarakat, sentimen dan emosi terhadap suatu produk yang berkaitan dengan isu sosial, acara,

dll. Hal ini sangat penting karena dapat membantu untuk mengetahui apakah seseorang menyukai atau tidak menyukai suatu produk [7]. Tiap opini yang dikeluarkan oleh masyarakat melalui sosial media ini memiliki polaritas. Polaritas merujuk pada bentuk dasar, yang menunjukkan apakah teks tersebut positif atau negative [6]. Penelitian terhadap penggunaan *Twitter* menunjukkan bahwa 19% dari *tweets* merujuk pada sebuah *brand* atau produk, 20% *tweet* juga menunjukkan sebuah ekspresi *sentiment* dari suatu *brand* [15]. Namun teks dari sosial media ini tidak terstruktur dengan baik sehingga menjadi tantangan untuk analisis sentimen. Teks tersebut biasanya bermasalah pada: panjang teks, *misspelling*, penggunaan slang, dan kata yang diperpendek [16].

### C. Aspect Based Sentiment Analysis

*Aspect based sentiment analysis* merupakan salah satu analisis sentimen yang mendeteksi sebuah sentimen terhadap aspek tertentu dalam sebuah kata. Kalimat yang memiliki beberapa aspek ini dapat dikategorikan sebagai *overlapping* atau *non-overlapping*. Sebuah kalimat dapat dikatakan sebagai *non-overlapping* apabila dua dari aspek kalimat tersebut tidak tumpang tindih [17]. *Aspect based sentiment analysis* bertujuan untuk mengenali polaritas dari sebuah *sentiment* terhadap aspek spesifik dari sebuah kalimat. Target dari aspek merujuk pada sebuah kata atau frasa yang mendeskripsikan aspek dari sebuah entitas [18].

### D. Support Vector Machine

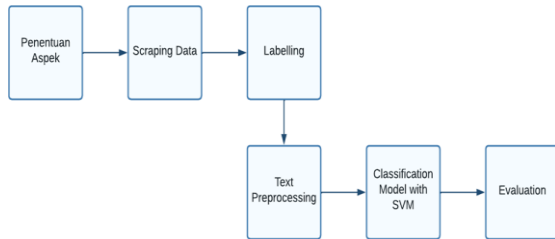
*Support Vector Machine* merupakan salah satu model *machine learning* untuk klasifikasi yang biasanya sering digunakan pada Analisis Sentimen. Kemampuan dari SVM itu sendiri adalah kemampuan untuk membaca pola data klasifikasi dengan akurasi dan reproduktifitas yang tinggi. *Decision function* dari SVM adalah memilih “*hyperplane*” yang paling optimal yang berfungsi untuk memisahkan observasi yang dimiliki oleh satu kelas dengan kelas lainnya berdasarkan pola informasi dari observasi tersebut yang disebut *feature* [19]. SVM telah digunakan untuk masalah *two-class classification* dan dapat diaplikasikan pada *kernel linear* dan *non-linear*. *Non-linear* (*polynomial*, *radial basis*, *sigmoid*) *classifier* menggunakan berbagai macam *kernel* untuk memaksimalkan margin antara *hyperplane* [20]. Beberapa kekurangan dari *support vector machine* antara lain adalah biaya komputasi pada *dataset* yang besar karena *training kernel matrix* tumbuh dengan bentuk *quadratic* dari ukuran *dataset*, dan tidak mudah untuk menyelesaikan masalah dengan *multi-classification problem* [21].

### E. Word Embeddings

*Word embeddings* merupakan salah satu metode *Natural Language Processing* (NLP) yang sering digunakan untuk mengolah teks, biasanya digunakan pada *Sentiment Analysis*, *Recommender System*, dll. *Word Embeddings* adalah sebuah teknik untuk merepresentasikan sebuah teks menjadi sebuah vektor numerik yang bertindak sebagai vektor dari sebuah kata [22]. *Word Embeddings* memiliki 2 kategori untuk metode evaluasi yaitu adalah *Extrinsic Evaluator* yang menggunakan *Word Embeddings* sebagai *Input*, Analisis Sentimen adalah salah satu contohnya, yang satunya adalah *Intrinsic Evaluator*

yang mengukur hubungan semantik antara sebuah kata secara langsung [23]. Tidak hanya digunakan untuk *Classification*, namun *Word Embeddings* juga dapat digunakan sebagai *Clustering* dan *Regression* [24].

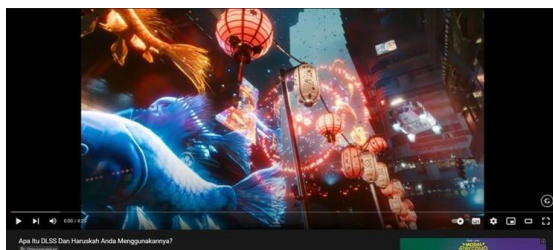
### III. METODE



GAMBAR 1  
FLOWCHART SISTEM

#### A. Penentuan Aspek

Sebelum melakukan *scraping data*, perlu melakukan analisa terlebih dahulu apa saja yang menjadi faktor/aspek penting ketika seorang *customer* memutuskan untuk membeli GPU Nvidia seri RTX. RTX sendiri adalah salah satu generasi baru GPU keluaran Nvidia pada tahun 2020. RTX sendiri baik dari segi fitur jauh lebih baik dibanding pendahulunya, dimana GPU ini memiliki performa yang sangat kuat, memiliki fitur terbaru yang disebut dengan *Deep Learning Super Sampling (DLSS)* untuk meningkatkan performa ketika *gaming*, dan *Ray Tracing* yang mampu meningkatkan kualitas visual sehingga terlihat nyata. Berdasarkan hal ini, akan *discraping* sentimen yang berhubungan dengan aspek fitur *ray tracing*, aspek fitur DLSS, dan aspek fitur performa. Penentuan teks itu sendiri apakah termasuk ke dalam aspek *ray tracing*/DLSS/performa, tergantung dari data yang *discrap*, apabila saat melakukan *scraping* yang berkaitan dengan harga kemudian teks/*sentiment* tersebut muncul tanpa harus menyinggung terkait harga secara langsung, maka akan termasuk ke dalam *dataset* harga, begitu pula dengan aspek yang lainnya.

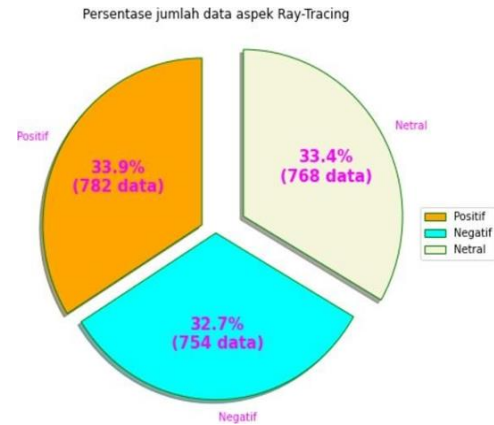


GAMBAR 2  
CONTOH VIDEO

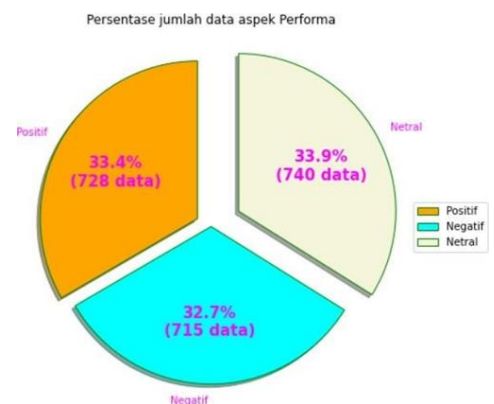


GAMBAR 3  
CONTOH COMMENT

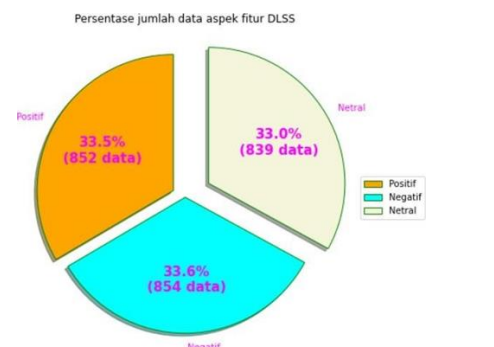
Gambar 2 menunjukkan sebuah video yang memperkenalkan fitur DLSS. Gambar 3 menunjukkan salah satu komentar yang ada pada video tersebut. Komentar tersebut akan masuk ke dalam *dataset* fitur DLSS karena saat *scraping* komen tersebut akan muncul, dan secara tidak langsung, komentar tersebut menyinggung tentang fitur DLSS.



GAMBAR 4  
PIE CHART ASPEK RAY-TRACING



GAMBAR 5  
PIE CHART ASPEK PERFORMA



GAMBAR 6  
PIE CHART ASPEK DLSS

Gambar 4, Gambar 5, dan Gambar 6, menunjukkan komposisi persentase jumlah data *positif*, *negative*, dan *netral* di masing masing aspek.

B. Scraping Data

Pada bagian ini akan melakukan *scraping* data yang diperlukan dari berbagai sumber. Pada *step* ini, sumber yang akan diambil datanya berasal dari Twitter dan Youtube, dengan menggunakan API yang disediakan oleh website netlytic.org untuk mengumpulkan *sentiment* yang diperlukan.

C. Labelling

Pada tahap ini, *labelling* dilakukan secara manual dengan menentukan apakah teks tersebut masuk kedalam sentimen positif, negatif, atau netral. Contohnya bisa dilihat seperti tabel dibawah :

TABEL 1  
TEXT SENTIMENT

Teks	Sentimen
<i>nobody is going to talk about the thumbnail</i>	Netral(3)
<i>sadly games are more beautiful than ever but theyre getting shittier as well</i>	Negatif(2)
<i>for people like me who are used to play since kid in low resolutions and low quality dlss is like a wonder</i>	Positif(1)

Jumlah *sentiment* pada masing masing aspek sebagai berikut :

TABEL 2  
JUMLAH DATA PER ASPEK

Aspek	Jumlah Data	Jumlah Positif	Jumlah Negatif	Jumlah Netral
Fitur Ray Tracing	2305 data	782 data	754 data	769 data
Fitur Performa	2184 data	729 data	715 data	740 data
Fitur DLSS	2545 data	852 data	854 data	839 data

E. Text Preprocessing

Pada bagian ini akan menghilangkan simbol-simbol, serta mengubah semua huruf menjadi kecil.

TABEL 3  
DAFTAR SIMBOL YANG DIHILANGKAN

!	(	)	-	=	?	{	\
#	'	*	:	>	@	}	^
“	&	+	/	<	[		_
\$	%	,	:	;	]	~	`

Kemudian melakukan *tokenizing*, pada bagian ini memerlukan akses tiap huruf dalam *string*. Maka kita memerlukan *tokenization* untuk memecah teks didalamnya menjadi komponen yang lebih kecil, diperlukan untuk mencari jumlah kata dan kalimat didalam text, menentukan berapa kali kata yang spesifik atau frasa tertentu muncul.

TABEL 4  
CONTOH TOKENIZATION

Text	Tokenized Text
<i>if you want to get a good graphics card with good price just buy it from vietnam you can get a radeon gb for just bucks</i>	<i>[im,glad,im,not,looking,for,a,gpu,right,now,mincraft,runs,at,about,smoothly,with,high,settings,so,im,good,rn]</i>
<i>im glad im not looking for a gpu right now minecraft runs at about smoothly with high settings so im good m</i>	<i>[im,glad,im,not,looking,for,a,gpu,right,now,mincraft,runs,at,about,smoothly,with,high,settings,so,im,good,rn]</i>
<i>oh my god it is so good</i>	<i>[oh,my,god,it,is,so,good]</i>

Kemudian menggunakan *stopword removal*, bertujuan untuk menghilangkan sebuah kata yang tidak memiliki arti atau informasi penting didalamnya, kamus *stopword* yang digunakan adalah kamus *English* tanpa memodifikasi/menambah kata yang perlu dihilangkan.

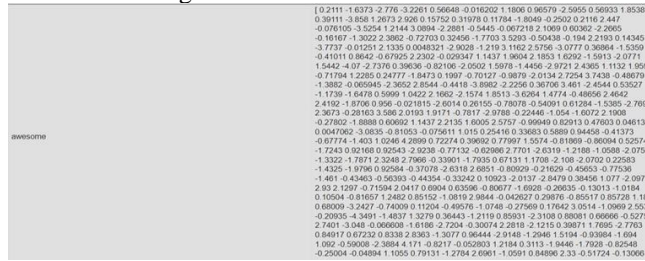
TABEL 5  
CONTOH STOPWORDS

Tokenized	Stopword Removal
<i>[yeah,ruming,ti,to,and,its,supe r,good,iddk,why,people,just,dont,by,used,ti]</i>	<i>[yeah,running,ti,super,good,iddk,people,dont,used,ti]</i>
<i>[im,glad,im,not,looking,for,a,gpu,right,now,mincraft,runs,at,about,smoothly,with,high,settings,so,im,good,rn]</i>	<i>[im,glad,im,looking,gpu,right,mincraft,runs,smoothly,high,settings,im,good,rn]</i>
<i>[oh,my,god,it,is,so,good]</i>	<i>[oh,god,good]</i>

Yang terakhir adalah pengaplikasian *word embeddings*. *Word embeddings* sendiri adalah teknik yang mengubah suatu teks menjadi sebuah vektor. Vektor tersebut merepresentasikan sebuah kalimat yang telah diubah sebelumnya. *Word embeddings* yang digunakan adalah spaCy.vector yang menggunakan algoritma dari GloVe untuk mengubah suatu kalimat menjadi sebuah vector.

Berdasarkan dari [25], *pre-trained word embeddings* memiliki bentuk 300 dimensi, dimana keseluruhan kata

di dalam teks ini merepresentasikan 300 dimensi yang ada, contoh *input* dan *output* teks yang telah mengalami vektorisasi sebagai berikut :



GAMBAR 7  
INPUT DAN OUTPUT WORD EMBEDDINGS

F. Classification Model with SVM

Pada tahap ini akan membuat model klasifikasi menggunakan *Support Vector Machine (SVM)*. SVM sendiri bekerja dengan cara menganalisis data dan menentukan *decision boundaries* dengan memiliki *hyperplane*, yang mana *hyperplane* ini memisahkan vektor dari satu kelas dengan kelas lain [26]. SVM menyediakan hasil akurasi yang lebih baik dalam mengklasifikasikan teks karena SVM sendiri termasuk *binary classifier* yang sering digunakan dalam *text classification* sehingga cocok digunakan untuk mengklasifikasikan polaritas dari sebuah kalimat [7]. Walaupun SVM adalah sebuah *binary classifier*, hasil yang diharapkan akan tinggi sekalipun *classifiernya* lebih dari dua, untuk *kernel* yang digunakan nantinya akan menggunakan *kernel* yang memiliki akurasi yang tinggi seperti yang ditunjukkan pada bagian *evaluation*. Ketika melakukan *fit dataset*, yang akan menjadi X adalah teks yang telah di *cleaning* dan di vektorisasi, Y adalah nilai sentimennya.

TABEL 6  
CONTOH HASIL PREDIKSI DAN AKTUAL

Teks	Hasil Prediksi	Aktual
<i>it probably took a big effort to make this not by you by your pc rip</i> <i>your pc</i>	3(neutral)	3(neutral)
<i>minecraft has its own feeling realistic graphic looks good but it doesnt give us feeling of minecraft that blocky game</i>	1(positif)	2(negatif)
<i>i cant even get fps normally without shaders lol</i>	1(positif)	1(positif)
<i>whats rtx</i>	3(neutral)	3(neutral)

<i>im watching this video in p because my computer cannot stand even</i>	3(neutral)	1(positif)
<i>this video lmao iwtd</i>	3(neutral)	3(neutral)
<i>everyone just had epic pc and im just sitting here with my old laptop</i>	3(neutral)	3(neutral)
<i>and still using core i without graphics card c</i>	3(neutral)	3(neutral)
<i>crysis is so bad it hasnt a walk animation</i>	2(negatif)	2(negatif)
<i>my pc crying after he seen this video</i>	1(positif)	2(negatif)
<i>that was so fuckin good that i just start sleeping so relax</i>	1(positif)	1(positif)
<i>nice but nothing against ray tracing</i>	1(positif)	1(positif)

Tabel 6 menunjukkan beberapa teks yang dibandingkan antara hasil prediksi, dengan nilai yang aktual/sebenarnya

G. Evaluation

Pada bagian ini akan menampilkan *classification report* yang terdiri dari *precision*, *recall*, dan *F1-score*, dimana masing masing metrik tersebut memiliki rumus sebagai berikut:

$$Precision = \frac{TP}{(TP+FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{2}$$

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

Pada bagian *evaluation* ini, akan menggunakan *cross validation* terhadap 4 *kernel* yang akan dipilih sebagai bagian dari SVM, yaitu *kernel RBF*, *sigmoid*, *polynomial*, serta *linear*.

IV. HASIL DAN PEMBAHASAN

Telah dilakukan eksperimen terhadap 3 macam *dataset* dan pemodelan *machine learning* yang telah dipilih, yaitu SVM. 3 *dataset* ini terdiri dari faktor yang berbeda-beda, yang pertama adalah faktor performa, kedua adalah faktor *ray tracing*, dan ketiga adalah faktor DLSS. Berikut akan ditampilkan *precision*, *recall*, dan F1-score dan akurasi dari masing masing *dataset*,

A. Dataset Performance

Pada *dataset scalper* ini terdapat sebanyak 2997 data, *setting* Gamma = 1 dan C = 1(*default*), cv = 10 ,kemudian untuk akurasinya terdapat 4 akurasi, yang pertama menggunakan *kernel* RBF dengan akurasi **41%**, kemudian menggunakan *kernel sigmoid* dengan akurasi **51%**, kemudian dengan *kernel polynomial* dengan akurasi **61%**, dan *kernel linear* dengan akurasi **64%**. Jumlah masing masing sentimen, *precision*, *recall*, dan f1-score dari masing masing *kernel* ditunjukkan pada tabel 8,

TABEL 7  
JUMLAH DATASET PERFORMA

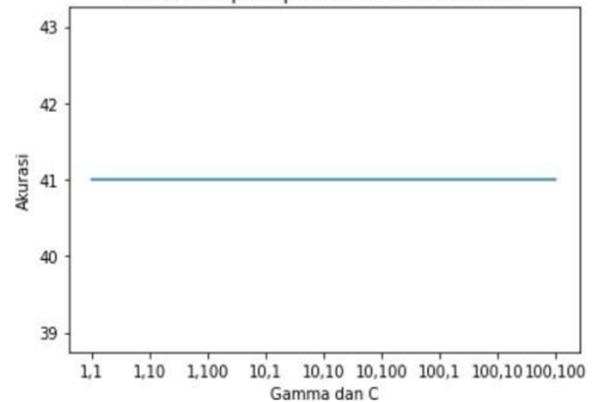
Label	Jumlah
1(Positive)	729
2(Negative)	715
3(Neutral)	740
Total	2184

TABEL 8  
AKURASI DAN METRIK DATASET PERFORMA

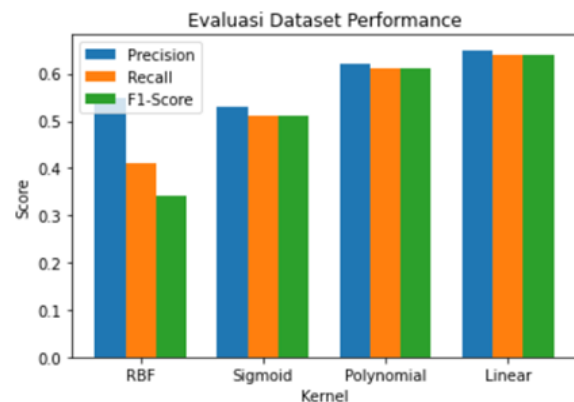
Kernel	Metrik			Akurasi
	Precision	Recall	F1-Score	
RBF(Gamma = 1, C = 1)	0.36	0.90	0.52	0.41 = 41%
	0.38	0.11	0.18	
	0.91	0.20	0.33	
Sigmoid	0.55	0.52	0.53	0.51 = 51%
	0.60	0.40	0.48	
	0.44	0.60	0.51	
Polynomial	0.70	0.54	0.61	0.61 = 61%
	0.60	0.54	0.57	
	0.56	0.74	0.64	
Linear	0.62	0.72	0.67	0.64 = 64%
	0.60	0.56	0.58	
	0.72	0.64	0.68	

Pada parameter Gamma dan C yang digunakan oleh *kernel* RBF pada SVM, menunjukkan bahwa pada perubahan nilai Gamma dan C mulai dari Gamma = 1, C = 1 hingga Gamma = 100, C = 100, tidak terjadi perubahan skor akurasi seperti pada gambar 8:

Akurasi rbf pada perubahan Gamma dan C



GAMBAR 8  
GRAFIK PERBANDINGAN SKOR AKURASI TERHADAP PARAMETER GAMMA DAN C



GAMBAR 9  
VISUALISASI EVALUASE DATASET PERFORMA(AVERAGE)

Di bagian ini *kernel linear* memiliki metrik paling tinggi secara keseluruhan, dari keseluruhan metrik, skor *precision* paling tinggi, persebaran skor pada *sigmoid*, *polynomial*, dan *linear* seimbang. Pada RBF, skor *recall* dan F1-score cukup jauh dibanding *precision*.

B. Dataset Ray Tracing

Pada *dataset harga* terdapat sebanyak 2100 data, *setting* Gamma = 1 dan C = 1(*default*), cv = 10 ,terdapat 4 akurasi, yang pertama adalah *kernel* RBF dengan akurasi **40%**, kemudian *kernel sigmoid* dengan akurasi **51%**, *kernel polynomial* dengan akurasi **62%**, *kernel linear* dengan akurasi **63%**. Berikut adalah jumlah masing masing sentimen, *precision*, *recall*, dan f1 dari masing masing *kernel*.

TABEL 9  
JUMLAH DATASET RAY TRACING

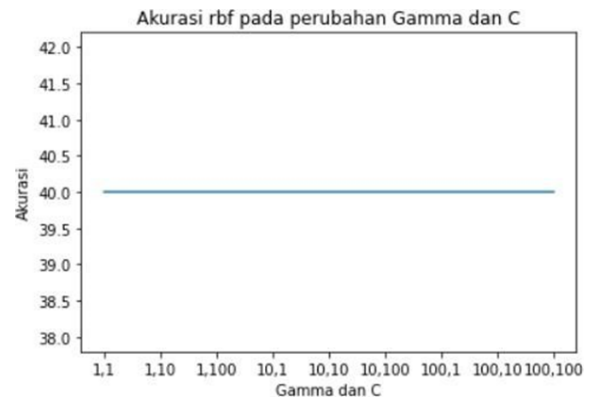
Label	Jumlah
1(Positive)	782
2(Negative)	754
3(Neutral)	769
Total	2305

TABEL 10  
AKURASI DAN METRIK DATASET RAY-TRACING

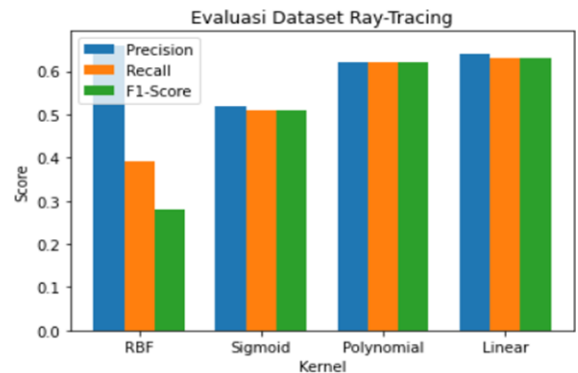
Kernel	Metrik			Akurasi
	Precision	Recall	F1-Score	
RBF(Gamma = 1, C = 1)	0.36	1.00	0.53	0.36 = 36%
	0.67	0.01	0.01	
	0.95	0.17	0.29	
Sigmoid	0.58	0.54	0.56	0.46 = 46%
	0.51	0.44	0.47	
	0.47	0.57	0.51	
Polinomia 1	0.67	0.59	0.63	0.63 = 63%
	0.59	0.53	0.56	
	0.61	0.74	0.67	
Linear	0.61	0.67	0.64	0.65 = 65%
	0.59	0.59	0.59	
	0.71	0.63	0.67	

Pada parameter Gamma dan C, yang hanya digunakan pada *kernel* RBF, menunjukkan bahwa perubahan nilai Gamma dan C, mulai dari Gamma = 1, hingga Gamma

= 100, C = 100, tidak memiliki pengaruh terhadap perubahan skor akurasi,



GAMBAR 10  
HASIL PERUBAHAN SKOR GAMMA DAN C PADA RBF



GAMBAR 11  
VISUALISASI EVALUASI DATASET RAY-TRACING(AVERAGE)

Di bagian ini, *kernel linear* memiliki skor *precision*, *recall*, *F1-score* paling tinggi dibandingkan *kernel* lainnya. Persebaran skor evaluasi dari *kernel sigmoid*, *polynomial*, dan *linear* terlihat seimbang, terkecuali *kernel RBF*, dimana skor *recall* dan *F1-score* berbeda jauh dibandingkan dengan skor *precision*.

B. Dataset DLSS

Pada *dataset* ini terdapat sebanyak 2400 data, *setting* Gamma = 1 dan C = 1 (*default*), cv = 10, terdapat 3 akurasi dari masing masing *kernel*, yang pertama dengan *kernel RBF* dengan akurasi **36%**, kemudian dengan *kernel sigmoid* dengan akurasi **46%**, *kernel polynomial* dengan akurasi **63%**, *kernel linear* dengan akurasi **65%**.

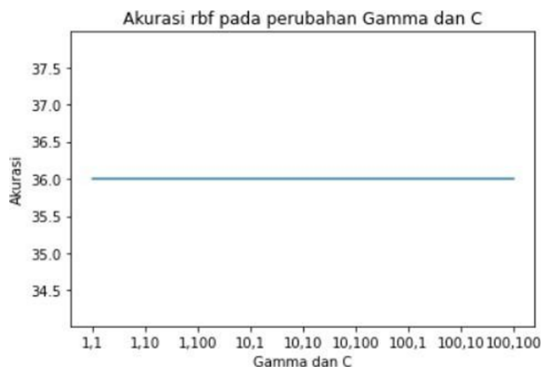
TABEL 11 - JUMLAH DATASET DLSS

Label	Jumlah
1(Positive)	852
2(Negative)	854
3(Neutral)	839
Total	2545

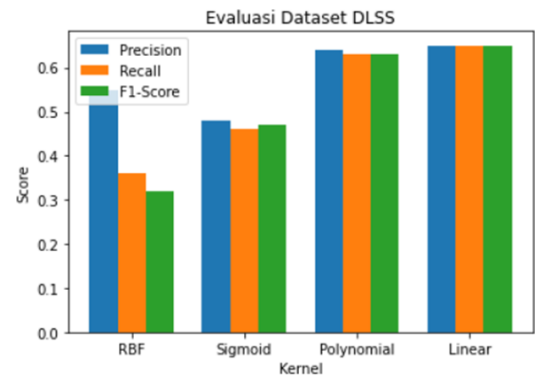
TABEL 12  
AKURASI DAN METRIK DATASET DLSS

Kernel	Metrik			Akurasi
	Precision	Recall	F1-Score	
RBF(Gamma = 1, C = 1)	0.35	0.40	0.37	0.36 = 36%
	0.34	0.60	0.44	
	0.97	0.09	0.16	
Sigmoid	0.55	0.46	0.50	0.46 = 46%
	0.50	0.42	0.46	
	0.39	0.51	0.44	
Polinomia 1	0.72	0.56	0.63	0.63 = 63%
	0.60	0.58	0.59	
	0.60	0.75	0.67	
Linear	0.64	0.72	0.68	0.65 = 65%
	0.62	0.60	0.61	
	0.70	0.63	0.66	

Pada parameter Gamma dan C, yang hanya digunakan pada *kernel* RBF, menunjukkan bahwa perubahan nilai Gamma dan C, mulai dari Gamma = 1, C = 1, hingga Gamma = 100, C = 100, tidak memiliki pengaruh terhadap perubahan skor akurasi seperti pada grafik berikut:

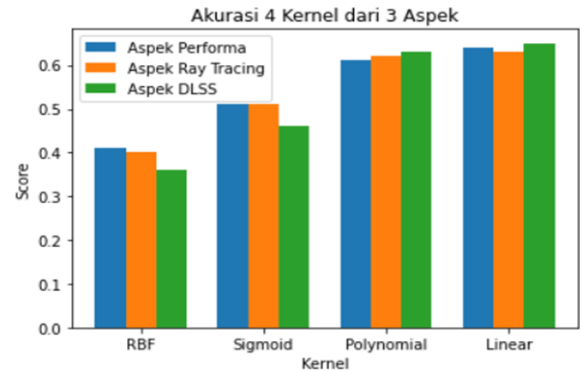


GAMBAR 12  
HASIL PERUBAHAN SKOR GAMMA DAN C PADA RBF



GAMBAR 13  
VISUALISASI EVALUASI DATASET DLSS(AVERAGE)

Di bagian ini, *kernel linear* memiliki skor *precision*, *recall*, *F1-score* paling tinggi dibandingkan *kernel* lainnya. Persebaran skor evaluasi dari *kernel sigmoid*, *polynomial*, dan *linear* terlihat seimbang, terkecuali *kernel RBF*, dimana skor *recall* dan *F1-score* berbeda jauh dibandingkan dengan skor *precision*.



GAMBAR 14  
VISUALISASI AKURASI 3 DATASET

Berdasarkan hasil yang telah ditunjukkan pada masing masing *kernel*, *kernel linear* unggul pada semua *dataset*. Dibawah *kernel linear* ada *kernel polynomial* yang akurasinya tidak berbeda jauh dengan *kernel linear*. Sedangkan *kernel RBF* dan *kernel sigmoid*, akurasi yang dihasilkan kurang memuaskan, parameter yang ditambahkan pada *kernel RBF* dimulai dengan Gamma = 1 C = 1, hingga Gamma = 100 C = 100 tidak memberikan hasil yang signifikan. Diantara 3 aspek, yang memiliki akurasi tertinggi dari keempat *kernel* pada gambar diatas adalah aspek DLSS dengan menggunakan akurasi hingga **65%** dengan *kernel linear*. Dengan membandingkan penelitian lain seperti [27] dan [28] menunjukkan bahwa *kernel linear* unggul pada analisis sentimen dibandingkan dengan *kernel* lainnya. Namun pada penelitian [29] menunjukkan hal berbeda, dimana yang paling unggul adalah *kernel RBF*. Pada grafik yang membandingkan skor dari *kernel* menunjukkan bahwa, *kernel RBF* memiliki skor *recall* dan *F1-score* yang cukup rendah dibandingkan dengan *precision*. Skor *recall* yang rendah menunjukkan bahwa teks dengan sentimen positif yang diprediksi positif tidak banyak. Menunjukkan bahwa *recall* rendah dapat mempengaruhi *F1-Score*.



## V. KESIMPULAN

Penelitian *multi-aspect sentiment analysis* terhadap Nvidia RTX Series telah menunjukkan beberapa hasil penerapan model *machine learning support vector machine* di beberapa *kernel*. Untuk pembahasan analisis sentimen terhadap produk Nvidia RTX kali ini menunjukkan bahwa SVM dengan *kernel linear* menunjukkan hasil terbaik di keseluruhan *dataset* dengan akurasi sebanyak **64%** pada *dataset performance*, **63%** pada *dataset ray tracing*, dan **65%** pada *dataset DLSS*. Melakukan perbandingan dengan *paper* lain juga menunjukkan *kernel linear* sering menunjukkan hasil terbaik dalam analisis sentimen, namun ada pula selain *kernel linear*, contohnya adalah *kernel RBF* yang menunjukkan hasil yang terbaik pada penelitian lain. Hal ini dipengaruhi beberapa faktor, diantaranya adalah *text preprocessing*, *feature extraction*, *dataset* yang digunakan, dan sebagainya. Penelitian *multi-aspect sentiment analysis* terhadap Nvidia RTX dapat dikembangkan lebih jauh lagi dengan seri GPU yang lain, serta menggunakan model dan metode yang berbeda juga.

## REFERENSI

- [1] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Math. Comput. Appl.*, vol. 23, no. 1, p. 11, 2018.
- [2] S. Rani, Ed., *Sentiment Analysis on Twitter Data using Support Vector Machine*, vol. 7, no. 3. 2016.
- [3] H. Parveen and S. Pandey, "Sentiment analysis on Twitter Data-set using Naive Bayes algorithm," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016.
- [4] V. Effendy, "Sentiment analysis on Twitter about the use of city public transportation using support vector machine method," *Int. J. Inf. Commun. Technol. (IJoICT)*, vol. 2, no. 1, p. 57, 2016.
- [5] H. Li, X. Li, D. Caragea, and C. Caragea, Eds., *Comparison of Word Embeddings and Sentence Encodings as Generalized Representations for Crisis Tweet Classification Tasks*. Proceedings of ISCRAM Asia Pacific, 2018.
- [6] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin, and L. Marlinda, "Comparison of SVM & naive Bayes algorithm for sentiment analysis toward west java governor candidate period 2018-2023 based on public opinion on twitter," in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, 2018.
- [7] S. Arafin Mahtab, N. Islam, and M. Mahfuzur Rahaman, "Sentiment analysis on Bangladesh cricket with support vector machine," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018.
- [8] U. Kumari, A. K. Sharma, and D. Soni, "Sentiment analysis of smart phone product review using SVM classification technique," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017.
- [9] M. Ahmad, S. Aftab, and I. Ali, "Sentiment Analysis of Tweets using SVM," *Int. J. Comput. Appl.*, vol. 177, no. 5, pp. 25–29, 2017.
- [10] P. H. Prastyo, A. S. Sumi, A. W. Dian, and A. E. Permanasari, "Tweets responding to the Indonesian government's handling of COVID-19: Sentiment analysis using SVM with Normalized Poly Kernel," *J. inf. syst. eng. bus. intell.*, vol. 6, no. 2, p. 112, 2020.
- [11] N. O. F. Daeli and A. Adiwijaya, "Sentiment analysis on movie reviews using Information gain and K-nearest neighbor," 2020.
- [12] S. Fransiska, Rianto, and A. I. Gufroni, Eds., *Sentiment Analysis Provider by U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method*, vol. 7, no. 2. Scientific Journal of Informatika, 2020.
- [13] B. Pratama et al., "Sentiment analysis of the Indonesian police mobile brigade corps based on twitter posts using the SVM and NB methods," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, p. 012038, 2019.
- [14] M. Guia, R. Silva, and J. Bernardino, "Comparison of naive Bayes, support vector machine, decision trees and random forest on sentiment analysis," in *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2019.
- [15] J. F. Sánchez-Rada and C. A. Iglesias, "Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison," *Inf. Fusion*, vol. 52, pp. 344–356, 2019.
- [16] N. N. Kham, Ed., *Lexicon Based Emotion Analysis on Twitter Data*. 2019.
- [17] M. Hu et al., "CAN: Constrained attention networks for multi-aspect sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.
- [18] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.
- [19] D. A. Pisner, "Machine Learning // Support vector machine," pp. 101–121, 2020.
- [20] I. Ahmad, M. Basher, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.
- [21] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on

- support vector machine classification: Applications, challenges and trends*,” *Neurocomputing*, vol. 408, pp. 189–215, 2020.
- [22] M. Habib, M. Faris, A. Alomari, and H. Faris, “*AltibbiVec: A word embedding model for medical and health applications in the Arabic language*,” *IEEE Access*, vol. 9, pp. 133875–133888, 2021.
- [23] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo, “*Evaluating word embedding models: Methods and experimental results*,” *arXiv [cs.CL]*, 2019.
- [24] Y. Li and T. Yang, “*Word embedding for understanding natural language: A survey*,” in *Studies in Big Data*, Cham: Springer International Publishing, 2018, pp. 83–104.
- [25] V. Raunak, V. Gupta, and F. Metze, “*Effective Dimensionality Reduction for Word Embeddings*,” in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 2019.
- [26] A. Tripathy, A. Agrawal, and S. K. Rath, “*Classification of sentiment reviews using n-gram machine learning approach*,” *Expert Syst. Appl.*, vol. 57, pp. 117–126, 2016.
- [27] K. D. Yonatha Wijaya and A. A. I. N. E. Karyawati, “*The effects of different kernels in SVM sentiment analysis on mass social distancing*,” *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 9, no. 2, p. 161, 2020.
- [28] R. Mukarramah, D. Atmajaya, and L. B. Ilmawan, “*Performance comparison of support vector machine (SVM) with linear kernel and polynomial kernel for multiclass sentiment analysis on twitter*,” *Ilk. J. Ilm.*, vol. 13, no. 2, pp. 168–174, 2021.
- [29] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, “*Indonesian sentiment analysis: An experimental study of four kernel functions on SVM algorithm with TF-IDF*,” in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 2020.