

# Klasifikasi Multilabel pada Teks Effect Kartu Monster Permainan Kartu Yu-Gi-Oh!

1<sup>st</sup> Danit Hafiz Pamungkas  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia  
danithp@students.telkomuniversity.ac.id

2<sup>nd</sup> Kemas Muslim Lhaksana  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia  
kemasmuslim@students.telkomuniversity.ac.id

**Abstrak** —*Yu-Gi-Oh! Trading Card Game* adalah sebuah permainan kartu dimana pemain membangun *deck*, menyusun strategi dan menghubungkan kemampuan atau *effect* suatu kartu dengan kemampuan kartu lainnya. Saat ini terdapat lebih dari 10000 kartu berbeda dengan *effect* berbeda sehingga dapat menyulitkan untuk mencari kartu dengan *effect* tertentu yang cocok dengan strategi yang ingin dilakukan. Terdapat aplikasi resmi yang dapat mencari kartu, termasuk dengan cara mencari kemampuan dari kartu tersebut. Namun, aplikasi tersebut memiliki kekurangan pada mesin pencariannya yang sangat sederhana dan dapat menghasilkan *false positive*. Dalam penelitian ini dibangun klasifier multilabel yang dapat mengklasifikasikan *effect* kartu untuk membantu pencarian kartu, dan juga menentukan praproses yang tepat untuk klasifikasi ini. Dilakukan pendekatan transformasi problem dimana klasifikasi multilabel dipecah menjadi 6 klasifikasi biner sesuai banyaknya label. Lalu, prediksi klasifikasi biner tersebut digabungkan menjadi prediksi klasifikasi multilabel. Klasifikasi dengan menggunakan praproses penghapusan *stop word* menghasilkan *micro average f1-score* terbaik dengan nilai 0.54. Walaupun begitu, nilai ini kurang baik dan menunjukkan bahwa klasifier belum dapat melabeli data dengan baik, sehingga klasifier yang dibangun belum dapat membantu pemain mencari kartu dengan kelas *effect* yang sesuai harapan.1

**Kata kunci**— klasifikasi, multilabel, stemming, penghapusan stop word, yu-gi-oh

## I. PENDAHULUAN

*Collectible Card Game* (CCG) atau *Trading Card Game* (TCG) adalah permainan kartu yang menggabungkan pembangunan *deck* secara strategis dan cara bermainnya [1]. Salah satu diantaranya yaitu *Yu-Gi-Oh!*, sebuah TCG yang dirilis pada Maret 2002 [2]. Pada Oktober 2019, jumlah kartu berbeda pada *Yu-Gi-Oh!* telah menembus 10000 kartu [3], dan dengan rilisnya produk-produk baru, jumlah ini akan terus bertambah.

Dalam *Yu-Gi-Oh!*, terdapat 3 jenis kartu, yaitu *Monster*, *Spell*, dan *Trap*. Sebuah *Monster* dapat memiliki informasi berupa nama, *Attribute*, *Level*, *Type* atau ras, dan *ATK* dan *DEF Point*, sementara kartu *Spell* dan *Trap* hanya memiliki nama dan sub-kategori khusus masing-masing kartu tersebut. Lalu, selain *Normal Monster*, setiap kartu dapat memiliki 1

atau lebih *Effect*, yaitu kemampuan yang dapat mempengaruhi keadaan permainan.

Berbeda dengan *Trading Card Game* lain, *Yu-Gi-Oh!* tidak memiliki sistem *resource* untuk membatasi gerakan yang dapat dilakukan seorang pemain pada giliran pemain tersebut pada umumnya. Sehingga *deck* pada *Yu-Gi-Oh!* biasanya dibuat *combo-oriented*, dimana strategi *deck* tersebut dibuat dengan menghubungkan sebuah *effect* kartu dengan *effect* kartu lainnya untuk dengan cepat mengeluarkan kartu utama dari *deck* tersebut.

Pada Mei 2011, diumumkan bahwa teks-teks pada kartu *Yu-Gi-Oh!* akan dirombak menggunakan metode *Problem Solving Card Text* (PSCT) untuk memperjelas teks tersebut, namun masih dapat dibaca oleh pemain [4], selain itu, kartu dengan *effect* serupa akan diseragamkan cara penulisannya. Perubahan ini mulai diimplementasikan dengan rilisnya *booster set Generation Force* pada 16 Agustus 2011[5].

Pada Pembangunan *deck* secara strategis sangatlah penting untuk memainkan sebuah *Trading Card Game*, termasuk *Yu-Gi-Oh!*, dan strategi pada *deck Yu-Gi-Oh!* umumnya dibuat dengan cara menghubungkan *effect* satu kartu dengan *effect* kartu lainnya (*combo*). Oleh karena itu, penting bagi seorang pemain untuk mengetahui *effect* yang ada pada kartu-kartu tersebut.

Banyaknya jumlah kartu dan beragamnya detail teks *effect* kartu-kartu tersebut dapat menyulitkan pemain untuk mencari kartu dengan *effect* yang sesuai dan dibutuhkan untuk strategi atau *combo* yang ingin dibangun. Untuk memudahkan pencarian kartu, terdapat perangkat lunak resmi yang bernama *Yu-Gi-Oh! Card Database* [6] yang juga diakses oleh aplikasi mobile resmi bernama *Yu-Gi-Oh! Neuron* [7].

*Yu-Gi-Oh! Card Database* dan *Yu-Gi-Oh! Neuron* memiliki sistem pencarian dan filter yang detail, dimana pengguna dapat mencari kartu berdasarkan Nama, Level, ras, dan lainnya, termasuk kata dalam teks *effect*. Namun, pencarian teks ini sangat sederhana dan dapat memunculkan kartu-kartu dengan konteks teks *effect* yang sangat berbeda.

Untuk contoh, seorang pemain melakukan pencarian kartu *monster* dengan teks *effect* menggunakan kalimat "destroy 1 card" pada aplikasi *Yu-Gi-Oh! Neuron* dengan harapan untuk mencari kartu yang dapat menghancurkan

(destroy) kartu lawan. Pencarian ini menghasilkan sebanyak 2226 kartu dengan 2 diantaranya yaitu monster bernama "Adreus, Keeper of Armageddon" dan "Alien Ammonite". Monster "Adreus, Keeper of Armageddon" memiliki effect untuk "target 1 face-up card your opponent controls; destroy it." [8] yang berarti monster ini dapat menghancurkan 1 kartu milik lawan. Sementara "Alien Ammonite" memiliki effect untuk "You can target 1 Level 4 or lower "Alien" monster in your graveyard; Special Summon it, but destroy it during the End Phase." [9] yang berarti monster ini dapat memanggil (summon) monster lain dari kuburan (dimana kartu monster yang sudah mati atau hancur disimpan), namun monster itu akan hancur lagi di akhir giliran.

Dari contoh hasil pencarian di atas, dapat dilihat bahwa konteks kedua effect tersebut sangatlah berbeda, namun dihasilkan dari satu pencarian yang sama. Untuk memperbaiki hasil pencarian tersebut, dapat dilakukan klasifikasi terhadap konteks dan tujuan dari teks-teks effect pada kartu.

Penelitian ini bertujuan untuk melakukan klasifikasi terhadap effect yang tertera pada kartu Yu-Gi-Oh!. Klasifikasi ini diharapkan dapat membantu pemain untuk mencari kartu yang memiliki teks effect, dimana konteks dan tujuan dari effect tersebut sesuai yang diharapkan oleh pemain yang melakukan pencarian kartu.

Karena satu kartu dapat memiliki lebih dari 1 label yang diakibatkan karena satu kartu dapat memiliki lebih dari 1 effect, klasifikasi dilakukan secara multilabel.

Jumlah kartu Yu-Gi-Oh! juga masih akan bertambah, sehingga diperlukan juga klasifikasi effect kartu baru yang berada diluar dataset untuk menilai ketepatan klasifier yang dibangun.

Untuk mengukur ketepatan dari klasifikasi akan dilakukan dengan melihat f1-score dari klasifikasi tersebut. Klasifikasi dapat dinyatakan baik dan berhasil jika dapat menghasilkan f1-score lebih dari 0.6. Selain itu, perlu diperhatikan pula nilai precision dan recall untuk menentukan apakah nilai f1-score cukup baik atau ada kekurangan.

## II. KAJIAN TEORI

Terdapat dua kategori untuk metode-metode klasifikasi multilabel [10], yaitu Problem Transformation dan Algorithm Adaptation. Problem Transformation adalah kategori metode yang mengubah klasifikasi menjadi biner atau multiclass, sementara Algorithm Adaptation merupakan kategori metode yang mengadaptasikan sebuah algoritma klasifikasi untuk klasifikasi multilabel.

Pada penelitian yang dilakukan oleh Tao Li dan Mitsunori Ogihara [11], metode yang digunakan termasuk dalam metode Problem Transformation, dimana klasifikasi multilabel diubah menjadi klasifikasi biner. Klasifikasi biner ini menentukan tiap data berupa musik dapat memiliki sebuah genre atau tidak, lalu klasifikasi ini dilakukan untuk setiap kelas atau label dengan Support Vector Machine. Klasifikasi ini menghasilkan nilai F-score yang rendah. Namun nilai F-score rendah ini diduga karena ketimpangan frekuensi label atau kelas pada dataset tersebut.

Dalam penelitian lain oleh Boutell dkk. [12], dilakukan perbandingan klasifikasi dengan berbagai macam metode yang termasuk dalam kategori Problem Transformation, diantaranya yaitu mengubah dataset multilabel menjadi

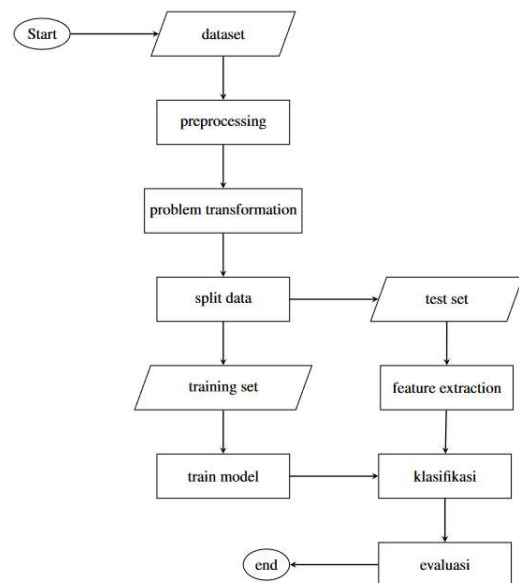
single label (disebut model-s), menghapus data-data yang multilabel (disebut model-i), dan memecah klasifikasi menjadi beberapa klasifikasi biner (disebut model-x). Klasifikasi yang menggunakan model-x menghasilkan rata-rata nilai precision dan recall tertinggi.

Pada penelitian yang dilakukan pada 2019 [13], analisis sentimen pada review hotel dilakukan menggunakan Multinomial Naive Bayes. Penelitian ini menerapkan feature selection, dimana kata atau term dengan frekuensi rendah tidak digunakan sebagai feature saat klasifikasi. Dengan menerapkan feature selection ini, nilai rata-rata f1-score nya naik sekitar 0.48% dari 90.93% menjadi 91.41%.

Selain itu, Eftekhar Hossain dkk. [14] melakukan klasifikasi kelas biner terhadap review buku Bengali menggunakan Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, KNN, SVM, dan Stochastic Gradient Descent. Dengan klasifikasi Multinomial Naive Bayes menghasilkan akurasi tertinggi sebesar 87%. Selain itu, pada penelitian ini juga dibandingkan performa klasifikasi n-gram, dengan uni-gram atau 1-gram menghasilkan akurasi tertinggi dibanding bi-gram dan tri-gram.

## III. METODE

Untuk penelitian ini, sistem dirancang untuk memenuhi task berupa klasifikasi effect ke kelas-kelas yang sesuai atau secara multilabel. Gambar rancangan sistem dapat dilihat pada Gambar 1.



GAMBAR 1.  
Gambaran Sistem

Secara umum, satu effect kartu dapat tergolong dalam kelas pada Tabel 1. Ada pula effect lainnya yang tidak atau dapat tergolong dalam kelas-kelas tersebut, namun untuk penelitian ini difokuskan pada 6 kelas tersebut.

TABEL 1.  
TABEL KELAS EFFECT

No	Kelas	Kriteria	Kata Kunci
1	<i>Advantage</i>	Memindahkan kartu dari deck dan/atau graveyard ke tangan	<i>Add, search, draw</i>
2	<i>Removal</i>	Memindahkan kartu lawan dari area permainan ke graveyard, banish, tangan atau deck baik dengan cara destroy atau lainnya	<i>Destroy, banish, return</i>
3	<i>Modify</i>	Mengubah, meningkatkan dan/atau mengurangi ATK, DEF, atau Level sebuah kartu, dapat juga berupa penggantian nama, attribute, type.	<i>Gains, becomes</i>
4	<i>Negation</i>	Menggagalkan dan/atau menghilangkan effect dari kartu lain.	<i>Negate</i>
5	<i>Protection</i>	Melindungi kartu dari effect kartu dan/atau pertarungan	<i>Unaffected, targeted</i>
6	<i>Summon</i>	Memanggil (summon) Monster dari tangan, Graveyard, Deck, atau Banish ke area permainan	<i>Summon</i>

Pada dataset yang digunakan, teks effect didapat dari query yang dilakukan pada API yang dibuat oleh situs YGOPRODeck [15] dengan batasan yang telah disebutkan sebelumnya, yaitu kartu berupa Monster yang memiliki effect dan telah rilis secara global dari Agustus 2011 sampai Februari 2023.

Data yang digunakan berupa kartu-kartu Monster yang memiliki effect dan rilis pada rentang waktu Agustus 2011 sampai Februari 2023 secara global, yaitu dari pack Generation Force sampai Photon Hypernova. Lalu penentuan kelas terhadap effect kartu dilakukan secara manual. Jumlah kartu yang digunakan sebanyak 1000 kartu dengan sebaran kelas pada Tabel 2.

Untuk klasifikasi menggunakan data diluar dataset, digunakan kartu terbaru yang rilis secara global pada 5 Mei 2023 sebanyak 5 kartu Monster dan 1 kartu Spell.

TABEL 2. TABEL KELAS EFFECT

No	Kelas	Jumlah Data
1	<i>Advantage</i>	313
2	<i>Removal</i>	239
3	<i>Modify</i>	299
4	<i>Negation</i>	194
5	<i>Protection</i>	215
6	<i>Summon</i>	476

Untuk mengurangi dimensi dari dataset, ada 6 praproses yang dilakukan terhadap dataset, yaitu penghapusan nama, penghapusan karakter non-alfabet, mengubah teks "GY" menjadi "graveyard", case folding penghapusan stop word, dan stemming.

Penghapusan nama pada teks effect dilakukan karena seringkali teks effect menyebutkan nama kartu, baik kartu itu sendiri maupun kartu lain. Selain itu, nama yang disebutkan dapat pula berupa nama keluarga atau archetype dari suatu kumpulan kartu. Untuk contoh, monster "Altergeist Multifaker" termasuk dalam keluarga monster "Altergeist".

Baik nama kartu maupun nama keluarga tersebut dianggap tidak penting untuk klasifikasi ini karena yang diklasifikasikan merupakan konteks dan tujuan dari teks effect itu, bukan ke-spesifik-an dari effect.

Selain itu, pada teks effect seringkali muncul karakter non-alfabet, baik itu tanda baca maupun angka. Angka yang muncul ini dapat berupa besaran nilai ATK, DEF, ataupun Level dari suatu monster. Besaran nilai ini dianggap tidak penting untuk klasifikasi ini, karena yang dianggap penting adalah kemampuan effect itu untuk mengubah nilai-nilai tersebut, bukan besaran nilainya. Angka juga digunakan untuk misalnya menentukan banyak kartu yang akan ditunjuk oleh effect, seperti "destroy 1 card your opponent controls" menunjukkan bahwa effect ini menghancurkan 1 kartu lawan. Seperti sebelumnya, banyaknya kartu yang dihancurkan dianggap tidak penting, yang dianggap penting hanyalah kemampuan effect untuk menghancurkan kartu lawan.

Pengubahan teks "GY" menjadi "graveyard" dilakukan karena penyingkatan ini dimulai pada tahun 2017 untuk mempersingkat teks effect. Untuk menyeragamkan kartu-kartu dari tahun 2017 ke atas dengan kartu-kartu dari tahun 2011 sampai 2016, teks "GY" dikembalikan menjadi "graveyard".

Pada case folding, huruf besar atau huruf kapital diubah menjadi huruf kecil.

Penghapusan stop word merupakan praproses dimana kata-kata umum yang sering muncul (disebut stop word) dihapus dari teks. Namun, karena penghapusan kata-kata tersebut dapat mengubah konteks dari teks effect, perlu dilakukan perbandingan performa antara klasifikasi yang menggunakan praproses ini dengan yang tidak.

Stemming adalah praproses untuk menghapus imbuhan dari kata-kata pada dokumen teks. Karena stemming dapat mengubah konteks dari teks effect, perlu dilakukan perbandingan untuk mengukur dampak dari praproses ini. Untuk contoh, kata destroyed akan diubah menjadi destroy. Kedua kata tersebut tentu memiliki konteks yang berbeda.

Salah satu pendekatan untuk klasifikasi multilabel adalah untuk mengubah klasifikasi menjadi klasifikasi biner terlebih dahulu. Pendekatan ini disebut juga dengan Problem Transformation.

Untuk melakukan klasifikasi biner, dataset multilabel harus diubah dulu menjadi dataset biner. Karena pada dataset terdapat 6 label, maka akan dilakukan klasifikasi biner sebanyak 6 kali sesuai label-labelnya.

Klasifikasi biner dilakukan terhadap dataset biner berdasarkan kepemilikan suatu label pada dataset binernya. Misalkan, monster "Alien Ammonite" memiliki label Summon, namun tidak memiliki label Removal, sehingga pada klasifikasi biner Summon, kartu tersebut memiliki kelas 1 atau positif, namun pada klasifikasi biner Removal, kelasnya adalah 0 atau negatif.

Setelah didapatkan dataset biner dari proses Problem Transformation, dilakukan feature extraction berupa perhitungan nilai TF-IDF dari setiap kata atau term pada dataset. Dengan perhitungan TF sebagaimana dituliskan pada (1). [16][17]

$$TF_{t,d} = \frac{f(t)}{T} \quad (1)$$

Dimana  $f(t)$  adalah jumlah kemunculan kata  $t$  pada sebuah dokumen  $d$ , dan  $T$  adalah jumlah keseluruhan kata pada dokumen  $d$ .

$$IDF_t = \log \frac{N}{DF_t} \quad (2)$$

Persamaan (2) adalah cara untuk menghitung IDF dari kata  $t$ , Dengan  $N$  merupakan banyaknya dokumen, dan  $DF_t$  adalah jumlah berapa kali kata  $t$  muncul pada kumpulan dokumen.

Setelah didapatkan nilai  $TF$  dan  $IDF$ , nilai  $TF-IDF$  dari suatu kata  $t$  pada dokumen  $d$  dihitung sebagai berikut.

$$TFIDF_{t,d} = TF_{t,d} \cdot IDF_t \quad (3)$$

Selain itu, dilakukan juga feature selection untuk mengurangi noise. Kata-kata atau term yang memiliki  $(df\_t)$  kurang dari  $(0.1)$  tidak akan digunakan sebagai fitur pada saat klasifikasi.

Klasifier yang digunakan adalah klasifier naive Bayes, yaitu klasifier probabilistik yang berasumsi kuat bahwa setiap variabel berdiri sendiri tanpa terpengaruh satu sama lain. Pada klasifier ini akan dihitung probabilitas dari masing-masing kelas untuk suatu instansi data, lalu diambil kelas yang probabilitasnya paling tinggi sebagai kelas dari instansi data tersebut. Menurut Haiyi Zhang dkk [18], klasifier naive Bayes dapat ditulis secara matematis sebagai berikut.

$$v_{NB} = \max P(C_j) \prod P(W_i|C_j), C_j \in C \quad (4)$$

Dimana  $P(C_j)$  adalah peluang kelas  $C_j$  yang adalah anggota  $C$ .  $P(W_i|C_j)$  adalah peluang kemunculan kata  $W_i$  pada dokumen berkelas  $C_j$ . Namun, karena klasifikasi ini menggunakan TF-IDF untuk feature extraction,  $W_i$  disubstitusikan dengan  $TFIDF_{(W_i|C_j)}$ .

Ketepatan klasifikasi diukur dengan menghitung nilai precision, recall dan F1-Score [19]. Precision adalah banyaknya tebakan atau prediksi positif yang benar (true positive) dibandingkan dengan banyaknya tebakan positif yang dilakukan. Nilai prediksi dapat dihitung sebagai berikut.

$$precision = \frac{TP}{FP} \quad (5)$$

Dimana  $TP$  adalah banyaknya *true positive* yang merupakan prediksi yang benar untuk kelas positif dari klasifikasi. Sedangkan  $FP$  yaitu *false positive* adalah prediksi yang salah untuk kelas positif tersebut.

Nilai Recall mengukur banyaknya data berkelas positif yang ditebak atau diprediksi dengan benar dari keseluruhan data berkelas positif. Recall dapat dihitung sebagai berikut.

$$recall = \frac{TP}{TP + FN} \quad (6)$$

Dengan  $FN$  atau *false negative* adalah data berkelas positif yang diprediksi negatif oleh klasifier.

Sementara, f1-score adalah rata-rata harmonis dari precision dan recall dan dapat dituliskan sebagai berikut.

$$f1 - score = \frac{precision * recall}{precision + recall} \quad (6)$$

#### IV. HASIL DAN PEMBAHASAN

Untuk pengujian, dataset sebanyak 1000 kartu dibagi secara acak menjadi train set dan test set menggunakan library sklearn dengan variabel `randomstate = 11`, dan `testsize = 0.4` sehingga dataset dipisah dengan rasio 60:40 yang berarti 40% dari dataset menjadi test set.

Dilakukan 4 skenario klasifikasi berdasarkan praproses-praproses yang dilakukan pada klasifikasi-klasifikasi tersebut dengan spesifikasi sebagai berikut.

1. Skenario 1 : Praproses yang digunakan adalah Penghapusan nama, penghapusan karakter non-alfabet, pengubahan kata GY menjadi graveyard, dan case folding.
2. Skenario 2 : Praproses yang digunakan pada Skenario 1 ditambah dengan praproses Penghapusan Stop Word
3. Skenario 3 : Praproses yang digunakan pada Skenario 1 ditambah dengan praproses Stemming
4. Praproses yang digunakan pada Skenario 1 ditambah dengan praproses Penghapusan Stop Word dan Stemming

Tabel 3 menampilkan nilai F1-Score untuk setiap skenario klasifikasi terhadap test set yang dirata-ratakan dengan `micro averaging`, `macro averaging`, `weighted averaging`, dan `samples averaging`. Sementara Tabel 4 menampilkan nilai F1-Score dari klasifikasi terhadap dataset yang berada diluar training set maupun test set.

TABEL 3.

TABEL HASIL PENGUJIAN 1

Skenario	Micro Avg	Macro Avg	Weighted Avg	Samples Avg
1	0.50	0.40	0.46	0.51
2	0.54	0.44	0.50	0.55
3	0.47	0.35	0.42	0.48
4	0.51	0.38	0.45	0.51

TABEL 4.

TABEL HASIL PENGUJIAN TERHADAP DATASET LUAR

Skenario	Micro Avg	Macro Avg	Weighted Avg	Samples Avg
1	0.21	0.15	0.18	0.17
2	0.38	0.34	0.35	0.30
3	0.30	0.19	0.25	0.25
4	0.38	0.31	0.32	0.30

Untuk Tabel 5 sampai Tabel 8 merupakan nilai-nilai akurasi berupa Precision, Recall, dan F1-Score untuk setiap kelas sesuai dengan kasus penggunaan praproses tambahan yang dituliskan pada Tabel 3.

TABEL 5.

HASIL PENGUJIAN 1 SKENARIO 1 PER KELAS

Kelas	Precision	Recall	F1-Score	Support
<i>Advantage</i>	0.93	0.32	0.48	131
<i>Removal</i>	1.00	0.04	0.08	91

Kelas	Precision	Recall	F1-Score	Support
Modify	0.83	0.23	0.36	124
Negation	1.00	0.22	0.37	76
Protection	0.84	0.20	0.33	79
Summon	0.78	0.74	0.76	195

TABEL 6.  
HASIL PENGUJIAN 1 SKENARIO 2 PER KELAS

Kelas	Precision	Recall	F1-Score	Support
Advantage	0.95	0.44	0.60	131
Removal	1.00	0.07	0.12	91
Modify	0.88	0.30	0.45	124
Negation	1.00	0.30	0.46	76
Protection	1.00	0.15	0.26	79
Summon	0.77	0.74	0.76	195

TABEL 7.  
HASIL PENGUJIAN 1 SKENARIO 3 PER KELAS

Kelas	Precision	Recall	F1-Score	Support
Advantage	0.94	0.34	0.49	131
Removal	1.00	0.00	0.00	91
Modify	0.90	0.23	0.36	124
Negation	1.00	0.16	0.27	76
Protection	1.00	0.13	0.22	79
Summon	0.74	0.71	0.72	195

TABEL 8.  
HASIL PENGUJIAN 1 SKENARIO 4 PER KELAS

Kelas	Precision	Recall	F1-Score	Support
Advantage	0.93	0.43	0.59	131
Removal	1.00	0.00	0.00	91
Modify	0.85	0.27	0.41	124
Negation	1.00	0.26	0.42	76
Protection	1.00	0.06	0.12	79
Summon	0.75	0.72	0.73	195

Berdasarkan pada data yang didapat dari Tabel 3, praproses penghapusan stop word berdampak positif terhadap akurasi untuk klasifikasi teks effect pada kartu. Sementara stemming berdampak negatif, hal ini dapat dilihat dengan perbandingan nilai-nilai f1-Score pada skenario 3 dan 4 terhadap skenario 2, dimana skenario 3 dan 4 menggunakan praproses stemming. Namun, untuk Pengujian 2 dimana data yang diklasifikasikan merupakan data diluar dari dataset training maupun testing, stemming dapat berdampak positif terhadap nilai f1-score dari klasifier, seperti dilihat pada skenario 2 pada Tabel 4. Walaupun begitu, klasifikasi skenario 2 menghasilkan nilai f1-score yang relatif lebih baik.

Jika dilihat pada Tabel 5 sampai Tabel 8, nilai precision untuk seluruh kelas dan skenario sangat baik. Namun nilai recall rendah kecuali untuk kelas Summon. Dilihat dari definisi recall, maka klasifier yang dibangun belum dapat membedakan kelas positif dan negatif dengan baik dan benar.

## V. KESIMPULAN

Klasifier yang dibangun dengan kombinasi metode-metode yang digunakan pada penelitian ini belum dapat menghasilkan klasifikasi yang baik. Klasifier hanya dapat memprediksi effect berkelas Summon dengan baik, sehingga klasifikasi ini belum dapat membantu pemain Yu-Gi-Oh! untuk mencari kartu dengan kelas effect yang diharapkan. Walaupun begitu, dengan tingginya nilai f1-score untuk kelas Summon, klasifier ini dapat membantu pemain Yu-Gi-Oh! untuk mencari kartu berkelas Summon.

Untuk praproses yang digunakan pada klasifikasi teks effect pada kartu Yu-Gi-Oh!, penghapusan stop word dapat meningkatkan kinerja dari klasifier, hal ini dapat diukur dengan meningkatnya nilai f1-score satu kelas paling besar 0.12. Stemming juga dapat meningkatkan f1-score untuk beberapa kelas, namun dampak negatifnya lebih terasa. Diperlukan penelitian lanjutan apakah dampak-dampak ini terjadi pula jika menggunakan metode lainnya untuk kasus yang sama.

## REFERENSI

- [1] Turkey, Selen, Sonam Adinolf, and Devayani Tirthali. "Collectible card games as learning tools." *Procedia-Social and Behavioral Sciences* 46 (2012): 3701-3705.
- [2] J.J. Miller. *Scrye Collectible Card Game Checklist & Price Guide*. Krause Publications, 2003.
- [3] K. Morrisy. "Yu-Gi-Oh! Card Game Now Has Over 10,000 Unique Cards". Internet: <https://www.animenewsnetwork.com/interest/2019-10-18/yu-gi-oh-card-game-now-has-over-10000-unique-cards/>.152351, Oct. 18, 2019 [Mar. 13, 2023].
- [4] K. Tewart. "Problem-Solving Card Text, Part 1: Reading the Cards of Tomorrow". Internet: <https://yugiohblog.konami.com/articles/?p=2906>, [Mar. 13, 2023].
- [5] "Generation Force". Internet: <http://www.yugioh-card.com/en/products/bp-genf.html>, Jun. 3, 2011 [Mar. 13, 2023].
- [6] "Yu-Gi-Oh! Card Database". Internet: [https://www.db.yugioh-card.com/yugiohdb/?request\\_locale=en](https://www.db.yugioh-card.com/yugiohdb/?request_locale=en) [Mar. 13, 2023].
- [7] "Yu-Gi-Oh! Neuron". Internet: [https://www.konami.com/games/eu/en/products/yugioh\\_neuron](https://www.konami.com/games/eu/en/products/yugioh_neuron) [Mar. 13, 2023].
- [8] "Adreus, Keeper of Armageddon | Card Details | Yu-Gi-Oh! TRADING CARD GAME - CARD DATABASE". Internet: [https://www.db.yugioh-card.com/yugiohdb/card\\_search.action?ope=2&cid=9846](https://www.db.yugioh-card.com/yugiohdb/card_search.action?ope=2&cid=9846) [Aug. 31, 2023].
- [9] "Alien Ammonite | Card Details | Yu-Gi-Oh! TRADING CARD GAME - CARD DATABASE". Internet: <https://www.db.yugioh->

card.com/yugiohdb/card\_search.action?ope=2&cid=8034 [Aug. 31, 2023].

[10] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007): 1-13.

[11] Li, Tao, and Mitsunori Ogihara. "Detecting emotion in music." (2003).

[12] Boutell, Matthew R., et al. "Learning multi-label scene classification." *Pattern recognition* 37.9 (2004): 1757-1771.

[13] Farisi, Arif Abdurrahman, Yuliant Sibaroni, and Said Al Faraby. "Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier." *Journal of Physics: Conference Series*. Vol. 1192. No. 1. IOP Publishing, 2019.

[14] Hossain, Eftekhar, Omar Sharif, and Mohammed Moshiul Hoque. "Sentiment polarity detection on bengali book reviews using multinomial naive bayes." *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2020*. Singapore: Springer Singapore, 2021. 281-292.

[15] "Yu-Gi-Oh! API by YGOPRODeck". Internet: <https://ygoprodeck.com/api-guide> [Mar. 13, 2023].

[16] Qaiser, Shahzad, and Ramsha Ali. "Text mining: use of TF-IDF to examine the relevance of words to documents." *International Journal of Computer Applications* 181.1 (2018): 25-29.

[17] Grootendorst, Maarten. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." *arXiv preprint arXiv:2203.05794* (2022).

[18] Zhang, Haiyi, and Di Li. "Naïve Bayes text classifier." *2007 IEEE International Conference on Granular Computing (GRC 2007)*. IEEE, 2007.

[19] Yun-tao, Zhang, Gong Ling, and Wang Yong-cheng. "An improved TF-IDF approach for text classification." *Journal of Zhejiang University-Science A* 6 (2005): 49-55.

