ISSN [e]: XXXX-XXX
DOI: xxx

Perolehan Flesch Reading Ease dari Cerpen Bahasa Inggris Menggunakan N-Gram

Anak Agung Advaita Paramtapa[1], Milda Gustiana Husada[2], Jasman Pardede[3]
[1,2,3] Program Studi Informatika, Institut Teknologi Nasional Bandung
Email: agung.diva@mhs.itenas.ac.id
Received DD MM YYYY **|** *Revised* DD MM YYYY **|** *Accepted* DD MM YYYY

ABSTRAK

Penelitian ini menjelaskan bagaimana menerapkan metode N-Gram dalam meninjau perolehan nilai Flesch Reading Ease pada cerpen berbahasa Inggris. Nilai FRE diperoleh berdasarkan data uji berupa cerpen berbahasa Inggris serta dukungan berupa kamus kata-kata bersuku yang digunakan dalam N-Gram. Formula Flesch Reading Ease digunakan untuk memperoleh nilai keterbacaan. Hasil penelitian menunjukkan bahwa, dari 40 cerpen yang diuji, 39 cerpen memiliki nilai Flesch Reading Ease di atas 100. Selain itu, waktu pemrosesan dipengaruhi oleh banyaknya kata dalam satu cerpen.

***Kata kunci**: Readability, Cerpen, bahasa Inggris, Flesch Reading Ease*

ABSTRACT

This study explains how to apply the N-Gram method in reviewing the acquisition of Flesch Reading Ease scores in English short stories. This study utilizes test data in the form of short stories in English and supporting data in the form of a dictionary of syllable words to cross-examine the results of cutting data using N- Gram. In reviewing readability, one formula is used in the Flesch-Kincaid Readability Level method which is still used in research involving documents. This formula, known as the Flesch Reading Ease, is used to find the readability value of a short story by utilizing a computer system using the N-Gram method. The results showed that, of the 40 short stories tested, 39 out of 40 short stories had a Flesch Reading Ease score above 100, while the recount results showed that there was one short story with a value above 100. In addition, processing time was influenced by the number of words in one short story.

***Keywords**: Readability, Short story, English, Flesch Reading Ease*

1. PENDAHULUAN

Cerpen menjadi media pembelajaran yang lebih efektif dibanding novel atau jenis karya tulis lainnya (Rohman, 2020) dan dapat disajikan dalam beberapa bahasa, termasuk bahasa Inggris. Namun demikian, adanya keterbatasan penyampaian cerita seperti rendahnya pemahaman membaca (Crossley et al., 2017; Yulisna, 2016) menyebabkan mereka yang membuat karya tersebut mengalami kesulitan. N-Gram sebagai metode pemrograman bersifat sederhana namun kompleks (van Gompel & van den Bosch, 2016) dan digunakan untuk meninjau *readability* menggunakan metode Flesch-Kincaid *Readability Level*. Flesch *Reading Ease Formula* merupakan rumus pertama dalam meninjau *readability* suatu teks yang didasari oleh panjang teks dan banyaknya suku kata per kata (Crossley et al., 2017). Namun, meninjau apa yang disebut *readability* terdapat kesulitan dalam menghitung jumlah kata, kalimat, dan suku kata yang terdapat di dalam cerpen mengingat cerpen di masa sekarang dapat mengandung antara 3000 hingga 10000 kata (Rohman, 2020).

Setiap cerpen menghasilkan keluaran keterbacaan yang berbeda, sehingga dirumuskan bagaimana cara sistem/program komputer memperoleh tingkat keterbacaan cerpen tersebut dan bagaimana cara mengukur tingkat keterbacaan cerpen tersebut. N-Gram tergolong metode sederhana, tetapi dapat digunakan di segala bidang, sehingga dirumuskan bagaimana jabaran proses N-Gram dalam memperoleh data, baik pengambilan data maupun pemrosesan untuk mencari hasil akhir data.

Berdasarkan pembahasan ini dirumuskan tujuan yaitu untuk mengukur *readability* suatu cerpen menggunakan formulasi yang terdapat pada metode Flesch-Kincaid *Readability Level* yang dibantu oleh metode N-Gram sebagai metode untuk pemecahan suatu kata. Metode penelitian yang digunakan menggunakan metode air terjun dengan parameter yang dikaji berupa nilai penghitungan untuk jumlah kata, kalimat, suku kata, dan Flesch *Reading Ease*.

2. TINJAUAN PUSTAKA

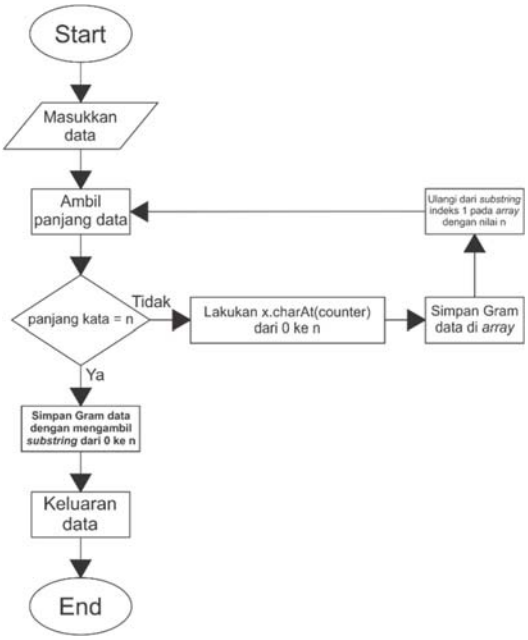
2.1.N-Gram

N-Gram merupakan teknik yang memisahkan kumpulan blok data menjadi blok-blok kecil dengan variabel n sebagai panjang dari posisi awal untuk mengambil data. N-Gram terbagi berdasarkan nilai n yang digunakan. Jika 1 disebut unigram, 2 disebut bigram, 3 disebut trigram, dan seterusnya (Lisangan, 2013). Dalam metode ini terdapat *pre-processing* yang "membersihkan" masukan blok teks (*string*) menjadi pecahan kata-kata yang disimpan di array. *Pre-processing* ini menghilangkan karakter-karakter khusus yang ada pada teks tersebut dan menampilkan keluaran dalam bentuk pecahan kata yang disimpan di dalam *array*. Proses ini dikenal sebagai *tokenization* (Kumar & Bhatia, 2013).

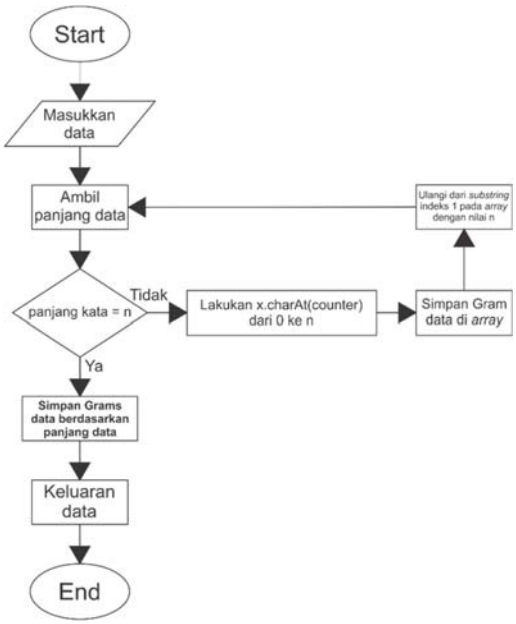
Pada N-Gram model pembagian data berdasarkan huruf demi huruf, pencacahan dilakukan dengan mencacah sebuah kata menjadi kumpulan huruf-huruf di dalam *array*. Cara mencacah kata menjadi kumpulan huruf adalah dengan sistem mengambil posisi atau indeks huruf (dalam pemrograman Java terdapat fungsi charAt(n) pada setiap variabel *string* yang mengambil satu huruf pada indeks bernilai n) dan kemudian menyatukannya hingga posisi penghitung dari nol mencapai nilai n dalam N-Gram.

Pada N-Gram model pembagian data berdasarkan kata demi kata, tekniknya sama dengan sebelumnya, tetapi perbedaannya terletak pada banyaknya kata di dalam *array*. Satu indeks array dapat menyimpan satu kata atau lebih, tergantung nilai n yang digunakan. Ketika sistem atau aplikasi meminta nilai 1 (unigram) pada N-Gram, keluaran sistem yang dihasilkan sama persis dengan keluaran *pre-processing* yang "membersihkan" masukan blok teks (paragraf) menjadi pecahan kata-kata. Jika tidak (misalnya nilai 2, 3, atau lebih), sistem N-Gram kemudian bekerja dengan menghitung ukuran *array* hasil *pre-processing* dan mengukur titik

henti (*stop point*) untuk mengambil data dari titik awal hingga titik henti.



Gambar 1. Diagram alir algoritma kerja N-Gram yang memecah huruf-huruf dengan *trailing spaces*



Gambar 2. Diagram alir algoritma kerja N-Gram yang memecah huruf-huruf tanpa *trailing spaces*

Pemakaian N-Gram juga ditujukan untuk mengganti atau menghilangkan bagian dari *string* (blok-blok huruf) yang terdapat di dalam *full string* (kata utuh). Penggantian atau penghilangan ini memanfaatkan fungsi *replace* yang mengganti *string* spesifik yang ditemukan berdasarkan panjang kata yang digunakan saat pencarian dengan *string* yang baru atau kosong. Contoh: kata antitesis dipecah menjadi 3-gram berdasarkan kata sis (3 huruf). Dalam pecahan yang menghasilkan 7 data (dengan indeks *array* dimulai dari angka 0), indeks terakhir (indeks ke-6) mengandung kata sis, sehingga sistem akan melakukan penggantian pada posisi kata tersebut. Dalam kasus ini, kata yang ditemukan dihilangkan sehingga menghasilkan keluaran antite yang digunakan untuk pencarian selanjutnya.

2.2.Readability

Readability berasal dari dua gabungan kata bahasa Inggris *read* dan *ability*. *Read* berarti baca dan *ability* berarti kemampuan. Gabungan kedua kata ini menciptakan apa yang disebut "kemampuan membaca" atau istilah saat ini yang disebut "keterbacaan". *Readability* merupakan suatu instrumen untuk menentukan seberapa terbacanya teks, artikel, atau tulisan bagi orang yang membacanya (Zamanian & Heydari, 2012). Dalam kategorisasi metode maupun rumus dalam teori *readability* ada rumus *readability* klasik seperti Flesch *Reading Ease*, *New* Dale-Chall *Readability Formula* (Crossley et al., 2019), dan *Gunning Fog* (Zamanian & Heydari, 2012), dan ada rumus *readability* modern seperti *Automated Readability Index* (ARI), Coleman-Liau, SMOG, LIX (Jensen, 2009), Fry *Readability Graph*, dan Flesch-Kincaid *Grade Level* (Zamanian & Heydari, 2012). Pada penelitian ini digunakan Flesch *Reading Ease*.

2.3.Flesch Reading Ease

Flesch *Reading Ease* adalah komponen dari Flesch-Kincaid *Readability Level* yang merupakan rumus keterbacaan paling awal untuk mendapatkan pengaruh luas terhadap pengembangan dan seleksi teks. Rumus ini didasari oleh panjang kalimat dan banyaknya suku kata per kata (Crossley et al., 2017). Metode ini dijabarkan dalam persamaan berikut:

$$FRE = 206{,}835 - 1{,}015\left(\frac{W}{S}\right) - 84{,}6\left(\frac{Syl}{W}\right) \quad (1)$$

Keterangan :

FRE : nilai Flesch *Reading Ease*
Syl : jumlah suku kata yang ditemukan
W : jumlah semua kata
S : jumlah semua kalimat

2.4.Cerpen

Cerpen, atau kependekan dari cerita pendek, adalah suatu bentuk prosa naratif fiktif atau cerita rekaan yang pendek yang cenderung padat dan langsung pada tujuannya (Yulisna, 2016). Cerpen tergolong dalam sebuah karya sastra yang bersifat rekaan dan singkat. Ia dapat dijadikan suatu media pembelajaran yang memiliki peranan dan unsurnya masing-masing, memiliki jumlah kata yang dapat diselesaikan tergolong singkat, membuat respons pembaca intensif dan membuat pembaca menjadi "merasakan apa yang dirasakan si tokoh", dapat memuat nilai-nilai kehidupan yang disampaikan dalam pesan, baik itu secara implisit atau eksplisit, dan lebih mudah dipelajari daripada novel atau genre lainnya untuk tujuan mengidentifikasi unsur-unsur fiksi (Rohman, 2020). Menurut Rohman (Rohman, 2020), ada 4 jenis cerpen berdasarkan panjang kata, yaitu cerpen yang pendek (*short short story*), cerpen

yang sangat pendek (*flash fiction*), cerpen cukupan (*middle short story*), dan cerpen yang panjang (*long short story*).

Fan fiction tergolong suatu karya seni, salah satunya karya tulis seperti cerpen. *Fan Fiction* berasal dari dua kata bahasa Inggris, *fan* yang berarti penggemar, dan *fiction* yang berarti fiksi. Jika diterjemahkan, *fan fiction* dapat diartikan sebagai fiksi penggemar. *Fan fiction* atau dapat disingkat sebagai *fanfic* merupakan sebuah karya tulis (puisi, cerpen, novel, dan karya tulis lainnya) yang dipublikasikan di Internet, yang mana dalam konteks ini, penulis cerita memiliki wawasan maupun keterkaitan dengan kesukaan akan suatu media, baik lisan maupun tertulis (Parrish, 2007). Salah satu penelitian yang menjelaskan bidang ini adalah penelitian yang menjelaskan Wattpad sebagai platform untuk mempublikasikan karya tulis seperti ini (Syaharani & Mahadian, 2017). Karya *fan fiction* tampil dalam bab-bab, yang menjadikan karya ini dapat disamakan dengan cerpen, novelet, atau novel, tergantung banyaknya kata dan isi yang disampaikan pada karya tersebut. *One Shot* dapat digolongkan juga sebagai *fan fiction*, hanya perbedaannya terdapat pada banyaknya kata di dalam isi ceritanya dan hanya terdiri atas satu bab yang merupakan badan cerita. Tergantung banyaknya kata pada cerita berbasis *Oneshot* ini, ini dapat digolongkan antara cerpen yang sangat pendek, cerpen yang pendek, dan cerpen cukupan. Cerita berbasis *One Shot* tidak pernah panjang dan dapat disebut sebagai cerpen jika ditinjau dari berbagai aspek.

2.5.Deteksi Kalimat

Komputer membaca simbol spesifik dengan mengandalkan kode ASCII pada sistemnya, serta perlu ada pemrograman untuk membuat komputer menerjemahkan simbol tersebut untuk menyatakan apakah simbol tersebut disebut penunjuk untuk menyatakan sebuah kalimat. Sama dengan model *character swapping*, untuk mendeteksi kalimat digunakan *array scanning* secara keseluruhan. *Array scanning* secara keseluruhan berarti *scanning* dilakukan paragraf demi paragraf (baris per baris) dalam dokumen. Dengan *scanning* semua data yang ada pada paragraf, baik huruf, angka, spasi, dan tanda baca, sistem akan menentukan *substring* spesifik yang menentukan apakah tanda tersebut dapat disebut sebagai tanda baca.

Tabel 1. Tanda baca yang menyatakan suatu kalimat

No.	Tanda Baca	Simbol	Kode ASCII
1	Seru	!	33
2	Titik	.	46
3	Titik dua	:	58
4	Titik koma	;	59
5	Tanya	?	63
6	Strip (*dash*)	–	150
7	Strip (*dash*) panjang	—	151

Terdapat contoh kalimat "Until Boruto's back in one piece, I appreciate your help." Sistem kemudian memeriksa kumpulan kata tersebut pada sebuah kalimat atau paragraf untuk menemukan tanda baca spesifik yang sudah ditentukan di dalam sistem yang dimulai dari posisi/indeks *array* 0 (huruf U). Saat sistem menemukan tanda baca titik (yang ketika diterjemahkan ke kode ASCII menghasilkan keluaran angka 46), sistem akan menganggapnya sebagai *identifier* untuk menyatakan sebuah kalimat.

2.6.Kamus Suku Kata

Menghitung *readability*, jika ditinjau dari rumus yang digunakan, tidak memandang bentuk kata, tetapi banyaknya suara (pecahan kata di dalam kata) yang terdapat di dalam kata tersebut. Hal inilah yang disebut sebagai suku kata. Suku kata merupakan pecahan dari suatu kata (representasi pengucapan) yang membentuk sebuah kata. Dalam bahasa Indonesia, contoh sederhana ini Budi menghasilkan 4 suku kata, yaitu i, ni, Bu, dan di. Dalam bahasa Inggris, pola pengambilannya berbeda, dan hal ini bergantung pada banyaknya suku kata dalam satu kata.

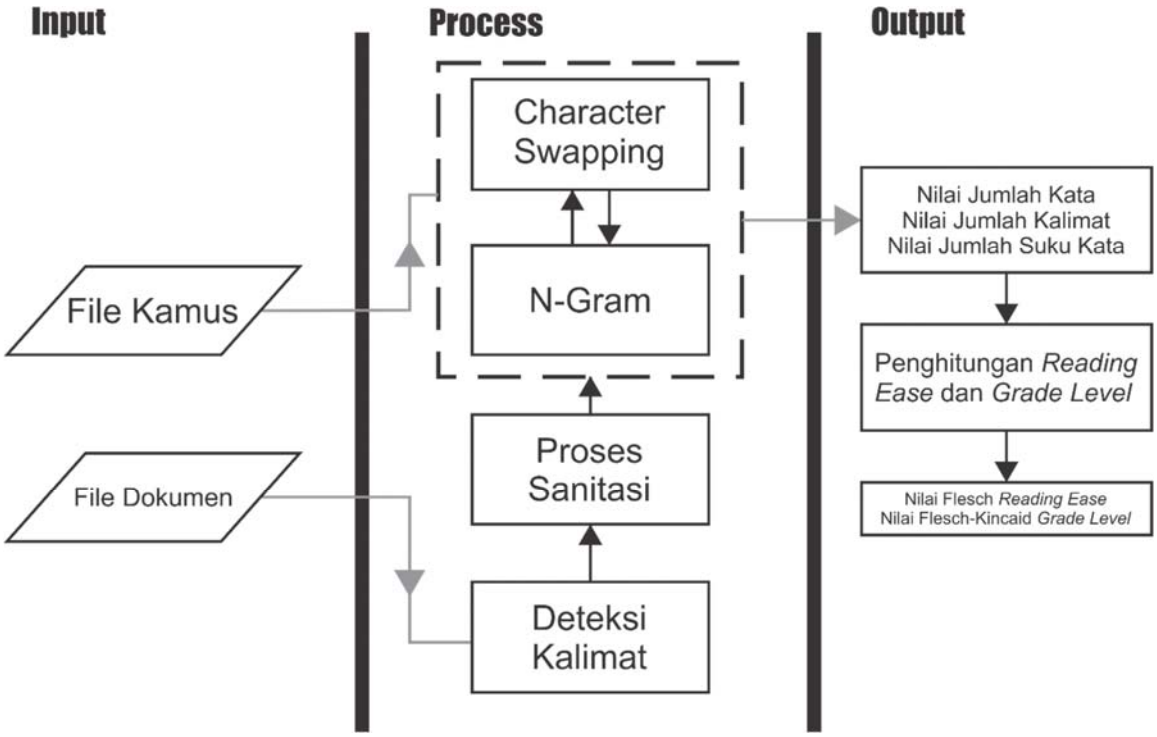
Kamus yang dibuat sendiri oleh penulis (manual) didasari dari situs web Stack Exchange (https://english.stackexchange.com/questions/64506/is-there-a-list-of-syllables-contained-in-us-english) yang menautkan sumber daya ke index.txt yang mengandung 15.832 kata berikut padanan suku kata pada setiap katanya dalam bahasa Inggris. Kata-kata tersebut belum dipecah mengikuti aturan pembagian kata menjadi suku kata dan hanya menampilkan bentuk suara dari kata yang dimaksud. Dari bentuk inilah tercipta kamus suku kata yang dibuat sendiri mengikuti aturan pembagian suku kata yang sudah disediakan melalui Internet (salah satunya dokumen dalam bentuk PDF). Kamus suku kata yang dibuat sendiri merupakan kamus yang dibuat tanpa bantuan komputer seperti perangkat lunak untuk menghasilkan *input-output* data.

Kamus suku kata yang sudah tersedia (di Internet) merupakan kamus kata yang sudah terdapat pemisah setiap blok katanya, yang menjadikan sebuah kata sudah diketahui berapa banyak suku kata yang ada pada kata tersebut. Kamus yang diambil di situs web GitHub ini (https://github.com/gautesolheim/25000-syllabified-words-list) didasari oleh dua sumber, yaitu daftar dari 180.000 kata-kata bersuku yang dirilis di domain publik melalui Proyek Gutenberg dan daftar dari kata-kata bahasa Inggris paling umum yang berasal dari *Google Web Trillion Word Corpus* oleh Peter Norvig.

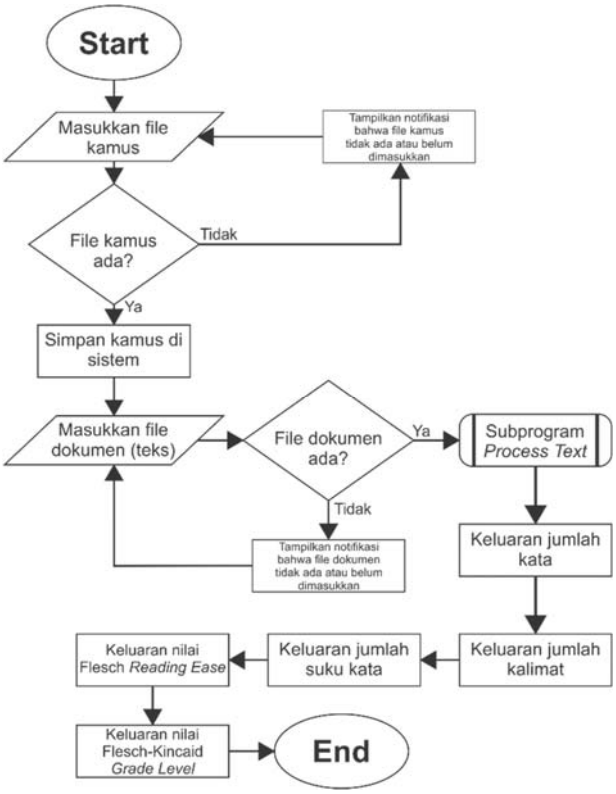
Kamus alfabet dan numerik digunakan apabila sistem memeriksa kata hingga sampai ke titik unigram dalam pemrosesan N-Gram. Kamus ini digunakan untuk membuat sistem membaca *singular letter* (huruf tersendiri atau satu huruf) yang tidak ditemukan dalam kamus suku kata. Kamus ini terdiri atas huruf alfabet dari A sampai Z (huruf besar dan huruf kecil, tetapi sistem akan mengonversikan semua huruf menjadi huruf kecil untuk konsistensi) dan bilangan dari 0 sampai 9. Kamus ini digunakan untuk menemukan suku kata dalam bentuk singkatan, seperti A.C.M.E., CV, APFSDS, dan lain sebagainya.

2.7.Cara Kerja Sistem

Pada Gambar 3 dan 4 dijelaskan blok diagram dan diagram alir kerja sistem yang menjelaskan penerapan dari metode yang digunakan dalam penelitian ini. Pada blok diagram dan diagram alir tersebut dapat dijelaskan bahwa aplikasi dapat berjalan ketika file kamus dan dokumen cerpen sudah dimuat ke dalam sistem, dan setelah proses aplikasi selesai, sistem akan menampilkan hasil akhir berupa nilai perolehan kalimat, kata, suku kata, dan nilai Flesch *Reading Ease*. Sistem tidak akan berjalan jika file kamus (minimal 1), file dokumen cerpen, atau kedua-duanya tidak dimuat.



Gambar 3. Skema desain sistem

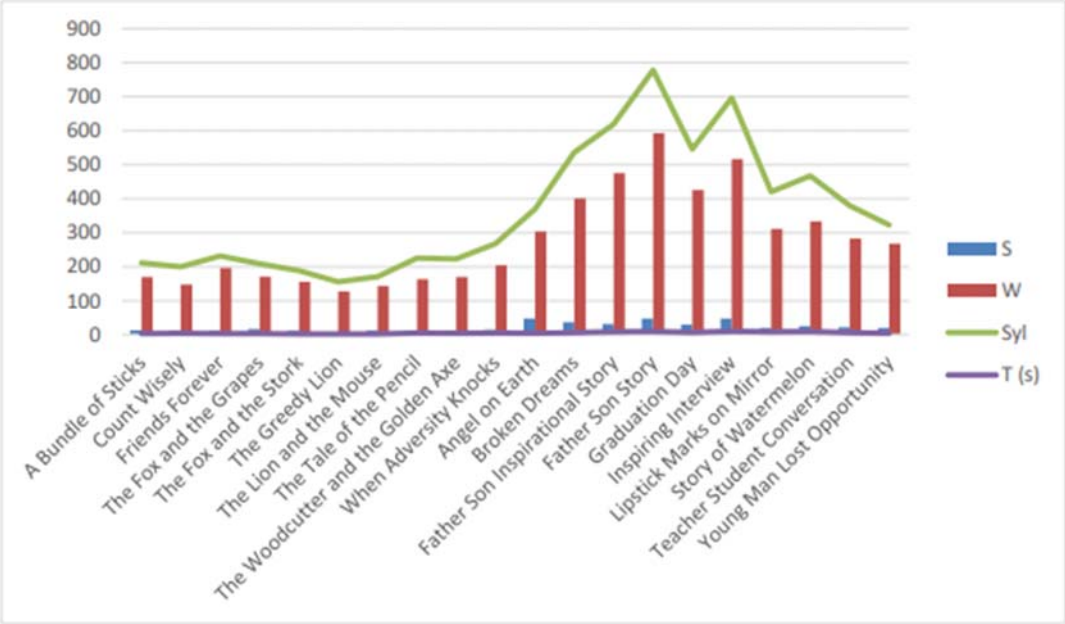


Gambar 4. Diagram alir kerja program

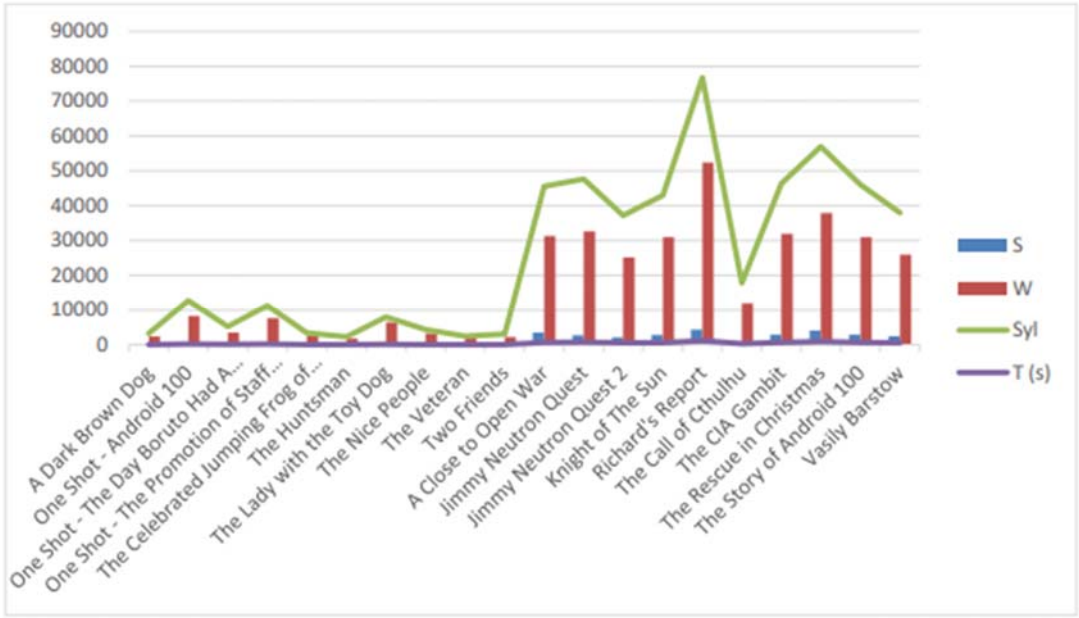
3. HASIL PENELITIAN DAN PEMBAHASAN

3.1.Hasil Penelitian

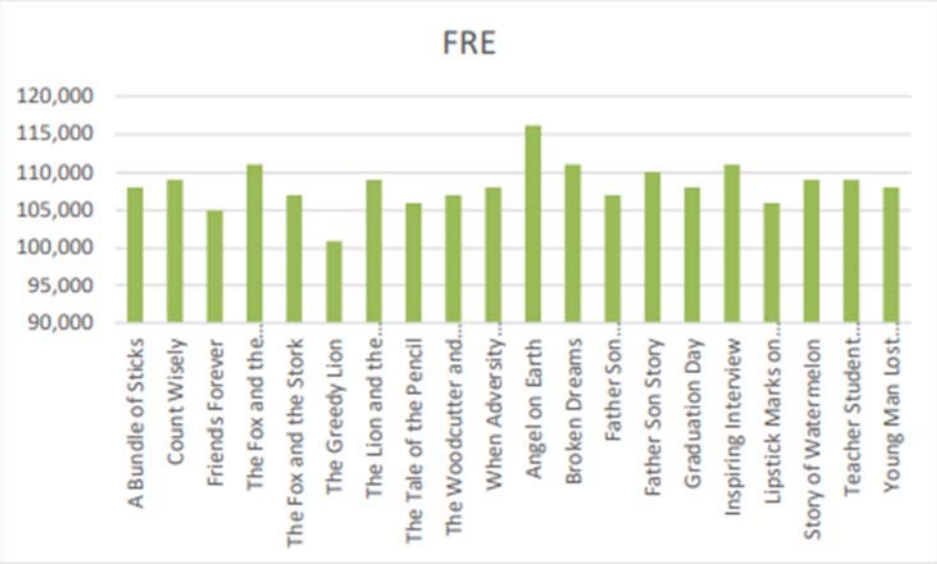
Penelitian ini menguji 40 cerpen berbeda dengan perbedaan panjang/isi cerpen setiap 10 cerpen. Hasil penghitungan dan waktu pemrosesan menampilkan 40 buah cerpen dengan hasil penilaian berupa banyaknya kata, suku kata, kalimat, dan nilai Flesch *Reading Ease*.



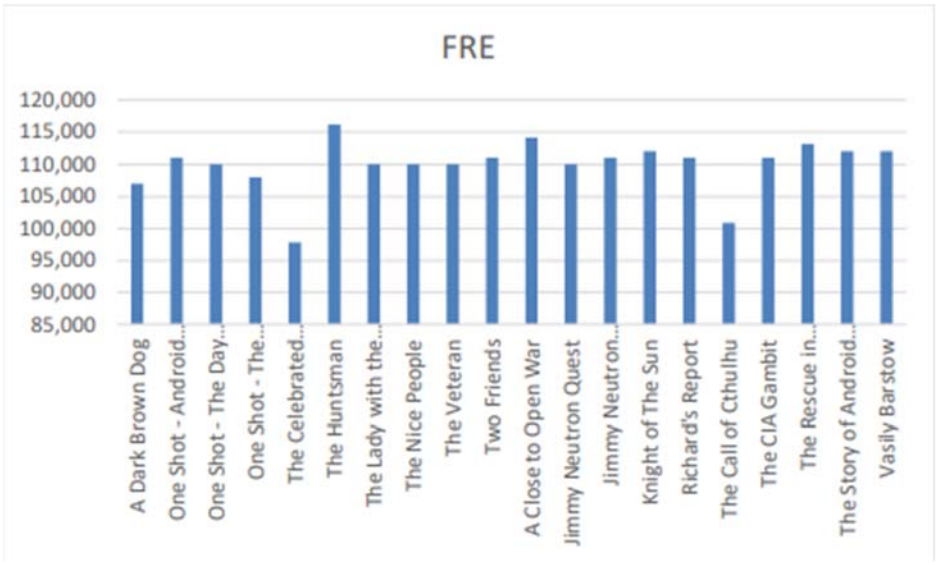
Gambar 5. Grafik pengujian sistem berdasarkan S, W, Syl, dan T (1)



Gambar 6. Grafik pengujian sistem berdasarkan S, W, Syl, dan T (2)



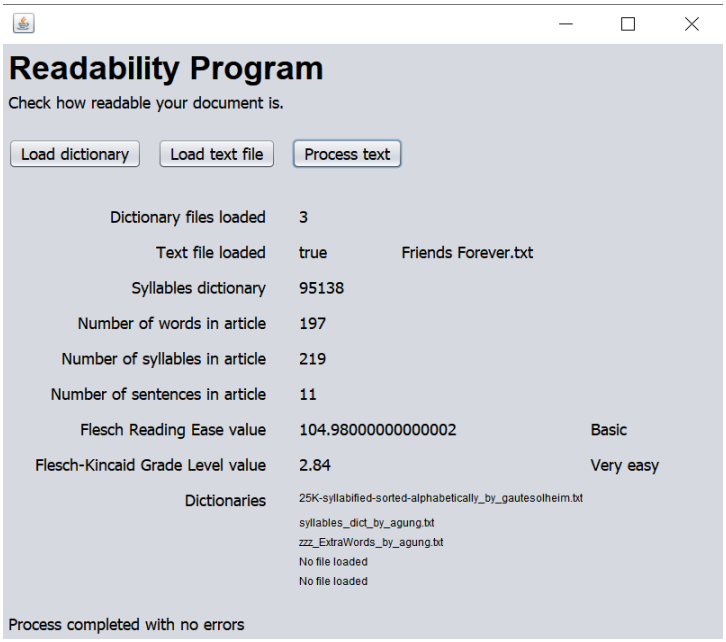
Gambar 7. Grafik FRE (1)



Gambar 8. Grafik FRE (2)

3.2.Pembahasan

Aplikasi ini bekerja dengan memanfaatkan kamus dan sebuah dokumen untuk menghitung keluaran berupa banyaknya kata, kalimat, suku kata, dan nilai Flesch Reading Ease. Penelitian ini menggunakan 3 kamus, yaitu kamus suku kata yang dipecah manual (dibuat sendiri oleh penulis) (15.832 kata), daftar kamus suku kata dengan lebih dari 25.000 kata bahasa Inggris yang paling umum, dan kamus alfabet dan numerik.



Gambar 9. Tampilan akhir aplikasi dengan tampilan hasil keluaran untuk *file* Friends Forever.txt

Cerpen *Richard's Report* memiliki waktu pemrosesan terlama 19 menit 4 detik dengan rerata waktu untuk memproses cerpen panjang 10 menit 51 detik. Cerpen ini merupakan cerpen *fan fiction* karya sendiri dengan jumlah kata terbanyak pada penelitian ini jika dibandingkan dengan cerpen yang lain dan terdiri atas 7 bab. 10 cerpen pertama dengan kategori anak-anak memiliki waktu pemrosesan kurang dari 10 detik. Waktu pemrosesan paling singkat adalah 1 detik (pada cerpen *The Fox and the Stork*, *The Greedy Lion*, dan *The Lion and the Mouse*) dan paling lama adalah 9 detik (pada cerpen *Inspiring Interview*). Dari pengujian sistem dapat dijelaskan bahwa hampir semua cerpen uji menghasilkan nilai *Reading Ease* melebihi batas 100 dan ada satu cerpen yang berada di bawah batas 100, yaitu 97. Meski secara program sudah benar, pengujian repetitif hingga 10 kali atau lebih pada cerpen yang sama (termasuk sebelum dan sesudah perubahan kode untuk memperbaiki *bug* pada penghitungan suku kata) menunjukkan nilai yang sama dan tidak berubah, kecuali ada perubahan atau anomali pada isi cerpen, misalnya adanya pemformatan yang tertinggal. Hasil penelitian ini akan menjadi kajian untuk penelitian selanjutnya.

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa penghitungan untuk mencari *readability* dapat dilakukan dengan menerapkan metode N-Gram dan formulasi/rumus *readability* berdasarkan Flesch *Reading Ease* (FRE), namun hasil keluaran sistem terbaca melebihi batas maksimum FRE yang perlu dikaji ulang untuk penelitian selanjutnya. Dapat dijelaskan bahwa pada penelitian ini semua cerpen tergolong kategori *Very Easy*, dengan parameter FRE menunjukkan 39 dari 40 cerpen uji memiliki nilai melebihi batas maksimum (100). Selain itu, parameter waktu pemrosesan menjelaskan bahwa semakin banyak kata yang terkandung dalam satu dokumen, semakin lama waktu yang digunakan untuk memproses keluaran. Penelitian ini dapat memberikan gambaran mengenai cara kerja *readability* terkomputerisasi meski terdapat *bug* di dalam pemrograman (seperti perancangan aplikasi maupun keluaran sistem yang tidak sesuai). Dalam merancang *flash fiction* atau cerita lainnya seperti dongeng, fabel, dan *fan fiction*, gaya bahasa dan peruntukannya harus diperhatikan.

DAFTAR PUSTAKA

Crossley, S. A., Skalicky, S., & Dascalu, M. (2019). Moving beyond classic readability formulas: new methods and new models. *Journal of Research in Reading*, *42*(3–4), 541–561. https://doi.org/10.1111/1467-9817.12283

Crossley, S. A., Skalicky, S., Dascalu, M., McNamara, D. S., & Kyle, K. (2017). Predicting Text Comprehension, Processing, and Familiarity in Adult Readers: New Approaches to Readability Formulas. *Discourse Processes*, *54*(5–6), 340–359. https://doi.org/10.1080/0163853X.2017.1296264

Jensen, K. T. H. (2009). Indicators of Text Complexity. *Copenhagen Studies in Language*, *37*, 61–80.

Kumar, L., & Bhatia, P. K. (2013). Text Mining : Concepts , Process and Applications. *Journal of Global Research in Computer Science*, *4*(3), 36–39.

Lisangan, E. A. (2013). Implementasi n-Gram Technique Dalam Deteksi Plagiarism Pada Tugas Mahasiswa. *TEMATIKA, Journal of Informatics and Information Systems*, *1*(2), 24–30. https://tematika.uajm.ac.id/index.php/tematika/article/view/10

Parrish, J. J. (2007). Inventing a universe: Reading and writing internet fan fiction. *English*, 196.

Rohman, S. (2020). *Pembelajaran Cerpen* (F. Azzahrah (ed.); 1st ed.). Bumi Aksara. Syaharani, N., & Mahadian, A. B. (2017). Perilaku Menulis Fanfiction Oleh Penggemar Kpop Di Wattpad. *Jurnal Komunikasi Global*, *6*(2), 200–219.

van Gompel, M., & van den Bosch, A. (2016). Efficient n-gram, Skipgram and Flexgram Modelling with Colibri Core. *Journal of Open Research Software*, *4*(August). https://doi.org/10.5334/jors.105

Yulisna, R. (2016). Kontribusi Kemampuan Memahami Cerpen Terhadap Keterampilan Menulis Cerpen Siswa Kelas Xi Sma Negeri 4 Padang. *Gramatika STKIP PGRI Sumatera Barat*, *2*(2), 72–83. https://doi.org/10.22202/jg.2016.v2i2.1101

Zamanian, M., & Heydari, P. (2012). Readability of texts: State of the art. *Theory and Practice in Language Studies*, *2*(1), 43–53. https://doi.org/10.4304/tpls.2.1.43-53