Pembuat Sampiran Pantun Otomatis berbasis *Pattern-matching*

Helena Nurramdhani Irmanda[1]*, Ria Astriratma[2], Nurul Chamidah[3], Mayanda Mega Santoni[4]
Fakultas Ilmu Komputer, Program Studi Sistem Informasi [1], [2]
Fakultas Ilmu Komputer, Program Studi Informatika [3], [4]
Universitas Pembangunan Nasional Veteran Jakarta
Jakarta Selatan, Indonesia
helenairmanda@upnvj.ac.id[1], astriratma@upnvj.ac.id[2], nurul.chamidah@upnvj.ac.id[3], megasantoni@upnvj.ac.id [4]

***Abstract*— This research was conducted to make an automatic sampiran pantun text based on pattern-matching and to analyze the level of naturalness of the resulting pantun. In the early stages, a database will be built consist of templates and a dictionary of terms. This system requires input from the user in the form of the contents of the pantun as a keyword. The next stage is determination of the template. From the keywords entered by the user, rhymes will be obtained. These rhymes will be matched into the dictionary of terms. Then, the system will retrieve the terms with the corresponding rhymes. The last step is to combine the variables in the template with the selected terms so as to create a complete sampiran. For the naturalness evaluation stage, it is carried out by giving questionnaire to respondents. The evaluation assesses the results of the sampiran text from the aspects of readability, clarity, and general appropriateness. The results of this study indicate that the pattern-matching method can be used to create sampiran pantun texts automatically according to the rules, both in terms of the number of lines and corresponding rhymes. This is in line with the results of the good naturalness evaluation from users in terms of readability, clarity, and general approriateness which are quite high at 95%, 93% and 97.5%, respectively.**

Keywords*— *pantun, pattern-matching, nlp, text processing

***Abstrak*— Penelitian ini dilakukan untuk membuat sampiran pantun otomatis berbasis pencocokan pola dan menganalisis tingkat *naturalness* dari pantun yang dihasilkan. Pada tahapan awal akan dibangun database berisi template dan kamus istilah. Sistem ini memerlukan *input* dari pengguna berupa isi dari pantun sebagai kata kunci. Kemudian dilakukan penentuan template. Dari kata kunci yang dimasukan oleh pengguna akan diperoleh rima. Rima ini akan dicocokan ke dalam database kamus istilah dan mengambil istilah dengan rima yang bersesuaian. Langkah terakhir yaitu melakukan penggabungan antara *variable* pada template dengan istilah yang terpilih sehingga membentuk teks sampiran yang utuh. Untuk tahap evaluasi naturalness dilakukukan dengan memberikan survey kepada responden untuk menilai hasil teks sampiran dari aspek keterbacaan, kejelasan, dan ketepatannya. Hasil dari penelitian ini menunjukan bahwa metode *pattern-matching* dapat digunakan untuk membuat teks sampiran pantun secara otomatis sesuai dengan kaidah, baik secara jumlah larik dan rimanya. Hal ini sejalan dengan hasil evaluasi *naturalness* yang baik dari pengguna dalam aspek *readability, clarity, dan general approriateness* yang cukup tinggi masing-masing sebesar 95%, 93% dan 97,5%.**

Kata Kunci*— *pantun, pattern-matching, nlp, text processing artikel

I. PENDAHULUAN

Pantun merupakan sebuah tradisi lisan dalam kebudayaan melayu yang sering digunakan sebagai sarana komunikasi masyarakat dalam menyampaikan pesan tertentu [1]. Di beberapa daerah di Indonesia, pantun sering digunakan dalam berbagai kegiatan misalnya tradisi berbalas pantun dalam acara pesta perkawinan di Tanjung Pura [2] dan di Sambas [3], dalam acara radio daerah [4] serta dalam berbagai acara televisi nasional [5]. Pantun dianggap mampu menjadi jembatan komunikasi, penghias, berdakwah, pendidikan, hiburan, serta sebagai simbol kebudayaan. Eksistensi pantun di tengah masyarakat modern ini menjadi pertanyaan dalam menghadapi aneka karya sastra populis lainnya. Perkembangan pantun menunjukan kecenderungan menyimpang dari kaidah pantun asli seperti jumlah baris dan kata, tidak adanya sampiran sebagai pengantar pantun, serta kesalahan pada perumusan rima [1].

Pembuatan pantun harus sesuai dengan syarat yakni pantun terdiri atas 4 larik dengan rima akhir a/b/a/b. Setiap larik *biasanya* terdiri atas 4 kata, larik 1-2 merupakan sampiran, larik 3-4 merupakan isi [6]. Permasalahan yang terjadi yaitu pada saat akan membuat pantun, seringkali sulit menemukan sampiran yang cocok dan sesuai dengan rima pada isi yang dimaksud. Berdasarkan latar belakang ini, dibutuhkan suatu Pembuat Teks Sampiran Pantun Otomatis sebagai inspirasi dalam pembuatan pantun sesuai dengan kaidah pantun itu sendiri.

Salah satu bidang dalam kecedasan buatan yang dimana komputer didesain untuk dapat berkomunikasi dengan manusia menggunakan bahasa alami, misalnya Bahasa Indonesia yaitu *Natural Language Processing* (NLP) [7]. Tujuan dari NLP yaitu memahami arti dari kalimat yang diberikan dalam bahasa manusia dan memberikan respon yang sesuai, seperti melakukan suatu aksi maupun menampilkan data [8]. Salah satu penerapan NLP dalam penelitian ini yaitu pembangkit sampiran pantun otomatis berbasis pattern-*matching*.

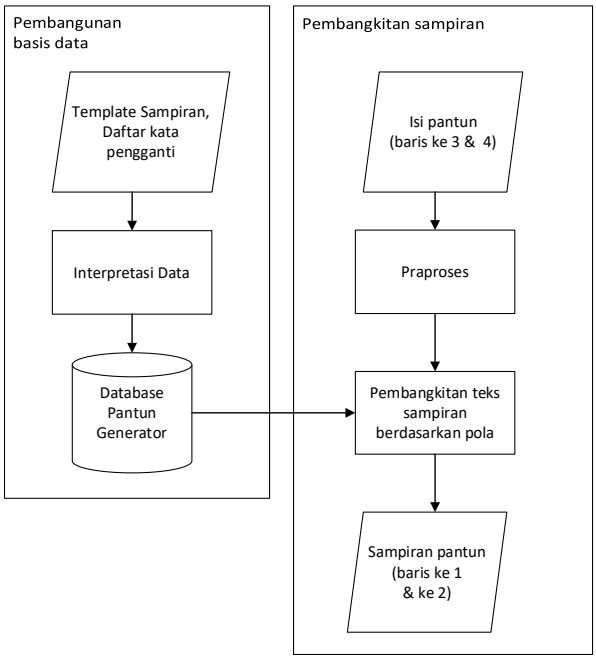
p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021

Pattern-matching atau pencocokan pola adalah salah satu metode dalam kecerdasan buatan yang bertujuan untuk melakukan pemeriksaan urutan token atau kata yang ada di dalam suatu kalimat [9]. Sebagai contoh adalah fasilitas search pada suatu halaman *web*. Misalnya akan dicari pola "lah" maka akan ditemukan pada kata – kata "salah", 'kalah" dan lain – lain. Sistem akan mencari pola dalam pengetahuan yang cocok dengan *input* dari user yaitu isi pantun kemudian mengambil respon/sampiran yang bersesuaian. Respon itu yang nantinya dikirimkan kembali kepada user. Semakin banyak *pattern* yang tersimpan maka semakin baik, karena akan membuat hasil pembangkitan kalimat sampiran pantun lebih alami [10].

Beberapa penelitian terkait telah dilakukan sebelumnya seperti pada penelitian *Web* Pembuat Puisi Otomatis Menggunakan Metode *Monte Carlo* [11]. Sistem menerima *input* tema puisi dari user, kemudian sistem akan menampilkan hasil puisi baru. Cara kerjanya yaitu sistem akan mengacak kata awal 90% dari tabel kata awal dan 10% dari tabel kata menggunakan metode Monte Carlo. Dari hasil penelitian dapat diambil kesimpulan bahwa aplikasi ini sudah mencukupi dalam membantu orang membuat puisi. Penelitian berikutnya yang terkait dengan pembuatan teks otomatis yaitu Pemanfaatan *Sentence-Similarity Measurement* untuk Proses Pencarian Pola pada *Chatbot* berbasis *Pattern-matching*. *Chatbot* yang dibangun menonjolkan interaksi antara manusia dan komputer dapat dilakukan dengan bahasa alami melalui media tulisan dengan pencocokan pola (*pattern-matching*) [10]. Pola – pola yang mungkin ditemukan selama proses percakapan tersimpan sebagai pengetahuan dalam bentuk *plain text* atau basis data. Berdasarkan latar belakang tersebut, maka dalam penelitian ini akan diimplementasikan metode *pattern-matching* untuk membuat sampiran pantun otomatis berdasarkan isi yang dimasukan oleh pengguna.

II. METODE PENELITIAN

Metode penelitian yang digunakan untuk membuat sampiran teks pantun otomatis ini yaitu dengan metode *pattern-matching.* Rancangan sistem yang akan dibangun ditunjukan pada Gambar 1.



Gambar 1. Arsitektur Pembuat Sampiran Pantun Otomatis

Berdasarkan Gambar 1, terdapat dua proses besar dalam penelitian ini, yaitu pembangunan *database* berdasarkan *input*an data pantun sebelumnya dan pembangkitan teks sampiran.

A. Pembangunan Basis Data

Masukan dari pembangunan basis data yaitu *template* pantun dan daftar kata pengganti (kamus). *Template* yang dibentuk dalam bentuk struktur linguistik berisi slot informasi yang relevan yang perlu diisi untuk mendapatkan output yang terbentuk dengan baik. Informasi yang relevan untuk dimasukkan ke dalam slot disimpan dalam tabel informasi yang relevan [12].

Template sampiran yang dikumpulkan sebanyak 5 jenis seperti pada Tabel 1.

Tabel 1.Daftar *template* sampiran

No	*Template* sampiran
1	Jalan-jalan ke kota [nama_kota] Jangan lupa membeli [benda]
2	Buah [nama_buah], Buah [nama_buah] Buah [nama_buah], Buah [nama_buah]
3	Ikan [nama_ikan], ikan [nama_ikan] Ikan [nama_ikan], ikan [nama_ikan]
4	Burung [nama_burung], burung [nama_burung] Burung [nama_burung], burung [nama_burung]
5	Hari Minggu pergi ke [nama_tempat] Di tengah jalan bertemu [nama_orang]

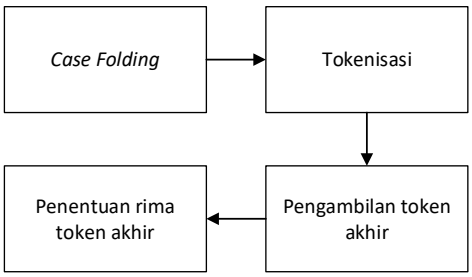
Daftar kata pengganti merupakan daftar kata-kata/istilah yang bisa menggantikan *variable* (yang ada di antara tanda "[]") pada *template* sampiran. Daftar kata pengganti ini disesuaikan dengan konteks pada *template* sampiran sehingga terbentuk beberapa konteks kata antara lain: Nama kota, benda, buah, ikan, burung, tempat, dan orang. Pada tahapan berikutnya yaitu interpretasi data, *template* sampiran dan kamus akan diinterpretasikan menjadi tabel dalam *database*.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021

B. Pembangkitan Teks Sampiran Pantun

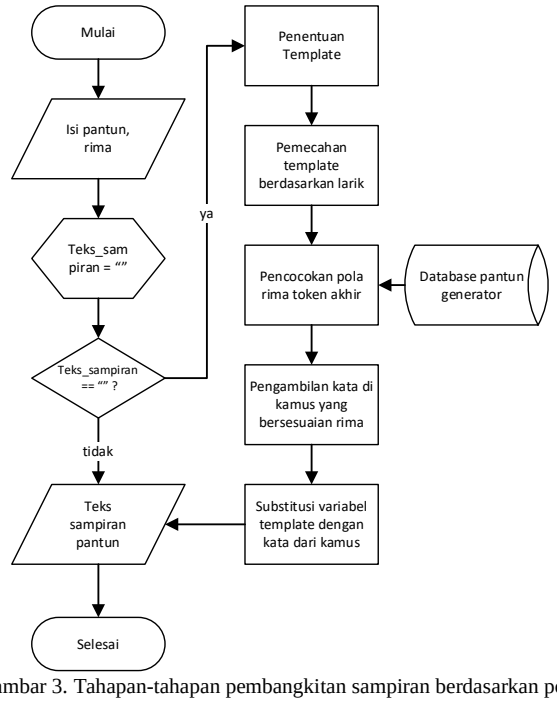
Pada tahapan pembangkitan teks sampiran pantun, pengguna diminta untuk memasukan isi pantun (baris ketiga dan keempat). Selanjutnya sistem akan melakukan praproses dan pencocokan pola, sehingga didapatkan teks sampiran pantun yang sesuai dengan isi pantun yang sudah di*input*kan oleh pengguna.

Praproses didilakukan pada teks isi pantun masukan dari pengguna dan dibagi ke dalam beberapa tahapan diantaranya: *case folding*, tokenisasi, pengambilan token akhir dari setiap baris isi pantun, dan penentuan rima dari token akhir.



Gambar 2. Tahapan-tahapan praproses

Setelah dilakukan praproses pada Gambar 2. Proses berikutnya yaitu pembangkitan sampiran berdasarkan pola. Tahapan-tahapan pada pembangkitan sampiran pantun berdasarkan rima diilustrasikan pada Gambar 3.



Gambar 3. Tahapan-tahapan pembangkitan sampiran berdasarkan pola

Berdasarkan Gambar 3, *input* pada pembangkitan sampiran berdasarkan pola adalah isi pantun dan rima dari setiap larik. Teks sampiran pada awalnya diinisialisasi dengan *string* kosong. Selama teks sampirannya masih kosong, akan dilakukan penentuan *template* yang diperoleh secara acak dengan membangkitkan bilangan acak 1 sampai dengan banyaknya *template*. Tahapan berikutnya yaitu pemecahan *template* berdasarkan larik, sehingga akan di dapatkan *template* sampiran larik ke 1 dan *template* sampiran larik ke 2. Setelah didapatkan *template* setiap larik, dilakukan pencocokan pola dari rima token akhir setiap larik dengan kamus yang ada pada database. Pencocokan pola juga dilakukan sesuai dengan konteks *template* yang diambil pada Gambar 1. Jika ada kata di kamus yang bersesuaian dengan rima token akhir, maka dilakukan subsitusi *variable* pada *template* dengan kata yang ada di kamus. Akhir dari proses ini yaitu didapatkannya teks sampiran pantun yang siap digabungkan dengan isi pantun dari *input* pengguna.

C. Evaluasi

Untuk mengevaluasi keberhasilan dari system ini digunakan penilaian naturalness. *Naturalness* mengukur kepuasan pengguna dalam memahami suatu teks yang dibangkitkan secara keseluruhan. Penilaian *naturalness* menggunakan kuesioner yang berisi tiga aspek evaluasi yaitu aspek *readability*, *clarity*, dan *general appropriateness* [13]. *Readability* mengukur seberapa mudah dibaca dan dipahami pengguna teks sampiran pantun yang dihasilkan. *Clarity* mengukur seberapa jelas penyampaian informasi dari teks sampiran pantun yang dihasilkan. Sedangkan *general appropriateness* mengukur seberapa sesuai teks sampiran pantun yang dihasilkan sehingga dapat menambah pengetahuan pengguna. Setiap aspek diukur dengan skala 1-5. semakin tinggi skalanya maka semakin baik hasilnya.

$$\%\ skor\ aspek = \frac{\sum total\ skor}{jumlah\ responden}\ x\ 100\% \qquad (1)$$

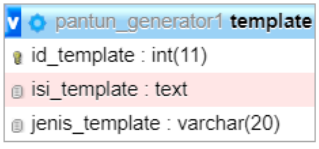
Penilaian akan dilakukan oleh 30 responden dari berbagai latar belakang pendidikan. Pengambilan responden ini berdasarkan pertimbangan bahwa sampiran pantun yang dihasilkan harus mampu dipahami oleh semua kalangan.

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari pembuat sampiran pantun otomatis dengan metode *pattern-matching* dijelaskan berdasarkan diagram yang telah direncanakan sebelumnya di bab 2 bagian metode penelitian.

A. Hasil pembangunan basis data

Pembangunan basis data diawali dari implementasi ke dalam *Database Managemen System*. DBMS yang digunakan pada penelitian ini yaitu MySQL. Dibentuk suatu database pantun *generator* untuk menyimpan data-data pantun yang dibutuhkan. *Template* yang telah didapatkan pada Gambar 2, disimpan dan diimplementasinya dalam bentuk tabel dengan struktur tabel pada Gambar 4.



Gambar 4. Tabel *Template*

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021

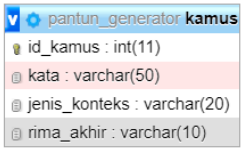
Berdasarkan Gambar 4, tabel *template* terdiri atas 3 field yaitu id_*template* dengan tipe data integer auto increment, isi_*template* bertipe data text, dan jenis_*template* bertipe data varchar(20). Pembuatan *template* dilakukan secara manual dan di*input* ke dalam tabel *template*.

Selain *template*, *input* yang berikutnya yaitu daftar kata pengganti (kamus) atau istilah yang bisa menggantikan *variable* pada area ("[]") di *template* sesuai dengan konteksnya. Contoh beberapa daftar kata pengganti dengan konteks Kota dideskripsikan pada Tabel 2.

Tabel 2.Daftar kata-kata pengganti

No	Kata	Jenis Konteks	Rima Akhir
1	Tangerang	Kota	ang
2	Langsa	Kota	sa
3	Lhokseumawe	Kota	se
4	Bandung	Kota	ung
5	Jakarta	Kota	ta

Kata-kata dalam kamus membutuhkan pengolahan data sehingga didapatkan rima akhir dari setiap kata. Struktur tabel Kamus ditunjukan pada Gambar 5.



Gambar 5. Tabel Kamus

Berdasarkan Gambar 5, tabel kamus terdiri atas 4 field yaitu id_kamus dengan tipe data integer auto increment, ikata bertipe data varchar(50), dan rima akhir varchar(10).

B. Hasil pembangkitan teks sampiran pantun

Proses ini diawali dengan *input* pengguna berupa isi pantun larik ke 3 dan larik ke 4, berikut contoh *input* dari pengguna:

ayo belajar hari ini
untuk masa depan yang lebih gemilang

Gambar 6. Contoh isi pantun masukan pengguna

Berdasarkan input dari pengguna pada Gambar 6, dilakukan praproses. Praproses dilakukan tahapan case folding, tokenisasi, pengambilan token akhir, dan penentuan rima.

Tahap case folding atau pengubahan teks isi pantun menjadi huruf kecil serta penghapusan karakter selain huruf [14].

Teks input = "Ayo belajar hari ini"

Teks output = "ayo belajar hari ini"

Tahap tokenisasi atau pemecahan teks isi pantun berdasarkan spasi dan tanda baca tertentu.

Teks input = "ayo belajar hari ini"

Teks output = [ayo, belajar, hari, ini]

Hasil dari tokenisasi akan digunakan dalam tahap pengambilan token akhir, dimana token yang diambil adalah token[n-1].

Teks input = [ayo, belajar, hari, ini]

Teks output = "ini"

Tahap terakhir pada praproses yaitu pengambilan rima dari token akhir, rima didapatkan dengan mengambil 2 karakter terakhir dari token. Khusus untuk karakter "ng" dan "ny", maka rima yang didapatkan dengan mengambil 3 karakter terakhir dari token.

Teks input = "ini"

Teks output = "ni"

Proses selanjutnya yaitu pembangkitan sampiran berdasarkan pola. Proses ini terus dilakukan selama teks sampiran yang dibangkitkan masih string kosong. Tahap pertama ialah penentuan template yang diambil secara acak. Contoh output dari penentuan diilustrasikan pada Gambar 7.

Hari Minggu pergi ke [nama_tempat]
Di tengah jalan bertemu [nama_orang]

Gambar 7. Contoh template yang terpilih

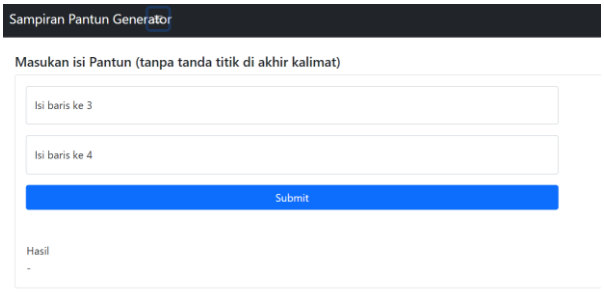
Template tersebut dipecah berdasarkan baris, sehingga akan didapatkan dua buah larik. Larik pertama yaitu "Hari Minggu pergi ke [nama_tempat]" dan larik kedua yaitu "Di tengah jalan bertemu [nama_orang]". Setelah didapatkan template, variable seperti [nama_tempat] dan [nama_orang] akan diisi dengan kata di tabel kamus sesuai dengan rima dari larik isi pantun dan konteks *template*nya. Jika kata yang bersesuaian dengan rima lebih dari satu kata, maka pengambilan kata/istilah dilakukan secara acak, sehingga sampiran yang dihasilkan akan lebih bervariasi. Sebagai contoh rimanya adalah "ni" dan konteksnya adalah nama tempat, maka akan dihasilkan istilah yang bersesuaiannya yaitu "pasar kerajinan seni".

Langkah berikutnya yaitu melakukan substitusi varibel dari *template* dengan kata yang terpilih. Dengan *input* seperti pada Gambar 6, hasil sampiran diilustrasikan pada Gambar 8.

Hari minggu pergi ke pasar kerajinan seni
di tengah jalan bertemu dalang

Gambar 8. Contoh hasil sampiran

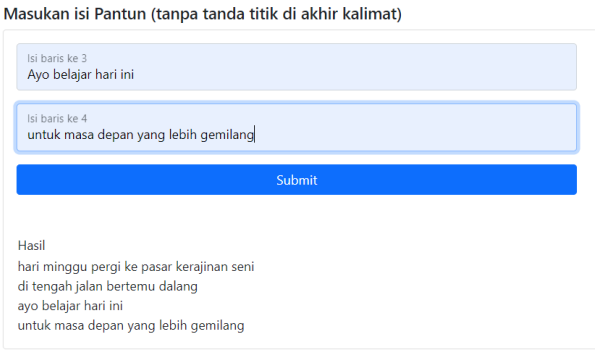
Pembuat teks sampiran pantun otomatis ini diimplementasikan dengan program berbasis *web* menggunakan bahasa pemrograman php. Pada Gambar 9 ditunjukan tampilan antarmuka *web* sampiran pantun generator.



Gambar 9. Tampilan antar muka sampiran pantun generator

Pengguna diminta untuk memasukan isi dari larik/baris ketiga dan keempat, sehingga dihasilkan pantun seperti pada Gambar 10.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021



Gambar 10. Contoh penggunaan sampiran pantun generator

C. Hasil pengujian *naturalness*

Pengujian *naturalness* dilakukan untuk menilai *readability*, *clarity*, dan *general appropriateness*. Terdapat 10 pantun yang sampirannya dibangkitkan oleh sistem seperti dideskripsikan pada Tabel 3.

Tabel 3.Daftar pantun yang dibangkitkan oleh sistem

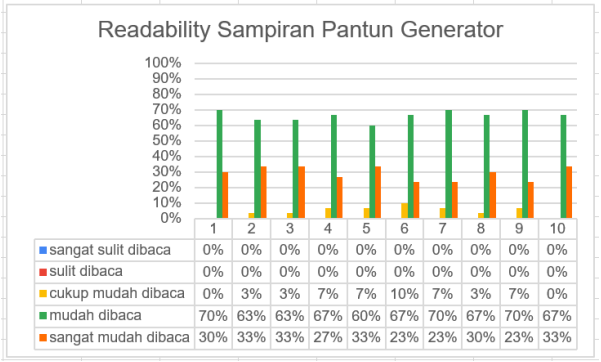
No	Pantun
1	Buah Paprika, Buah Duku Buah Lai, Buah Gandaria Biasakanlah membaca buku Sebab ia jendela dunia
2	Hari minggu pergi ke pasar kerajinan seni di tengah jalan bertemu dalang ayo belajar hari ini untuk masa depan yang lebih gemilang
3	Burung Kacer, Burung Manyar Burung Kenari, Burung Mantenan bagi siapa yang rajin belajar pasti akan sukses kemudian
4	Ikan Kerapu, Ikan Mas Ikan Tongkol, Ikan Tambakan ayo teman kita bersihkan kelas agar belajar makin nyaman
5	Buah Limau, Buah Mahkota Dewa Buah Beri Emu, Buah Lengkeng nenek menangis sambil tertawa melihat kakek bermain kelereng
6	Jalan-jalan ke Kota Batu jangan lupa membeli jambu semarang seseram - seramnya hantu lebih seram kecoa terbang
7	Burung Cililin, Burung Tengkek Udang Burung Ekek, Burung Sikatan bila hati kita senang hidup ini pun terasa ringan
8	Hari minggu pergi ke balai Kota di tengah jalan bertemu Siti walau dirimu jauh di mata namun tetap dekat di hati
9	Jalan-jalan ke kota Singkawang jangan lupa membeli sawo manila bila ada umur panjang boleh kita berjumpa pula
10	Ikan Layang, Ikan Tambakan Ikan Nilem, Ikan Terbang maksud hati ingin jalan namun sayang tidak punya uang

Pantun-pantun yang dihasilkan oleh sistem dinilai oleh 30 responden. Responden diminta memberikan rating pada setiap pantun dengan pertanyan yang mencakup 3 aspek antara lain *readability*, *clarity*, dan *general appropriateness*. Pemetaan aspek dan pertanyaan dideskripsikan pada Tabel 4.

Tabel 4.Pernyataan kuesioner untuk menguji *naturalness*

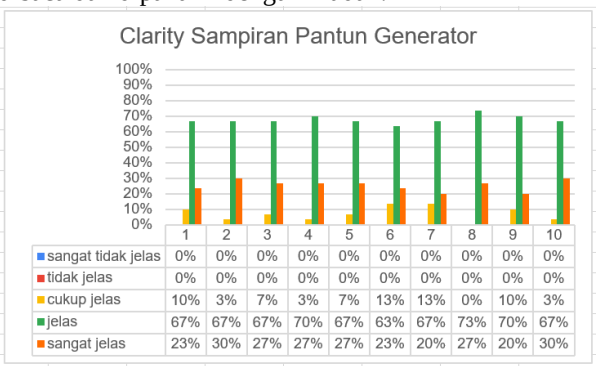
No	Aspek	Pertanyaan
1	*Readability*	Apakah teks pantun berikut ini mudah untuk dibaca dan dipahami oleh anda?
2	*Clarity*	Apakah teks pantun di atas sudah jelas dalam hal hubungan antar kalimat (sampiran dan isi)?
3	*General Appropriateness*	Apakah teks di atas sudah tepat disampaikan sebagai pantun?

Setiap pertanyaan pada Tabel 4, dijawab oleh responden dengan skala 1-5. Hasil pengujian untuk readability disajikan pada Gambar 11.



Gambar 11. Hasil pengujian *readability*

Berdasarkan hasil pengujian *readability* pada Gambar 11, dapat diketahui bahwa nilai rata-rata keseluruhan teks sampiran dari pantun ke 1 sampai ke 10 yaitu 5% responden menyatakan bahwa teks pantun cukup mudah dibaca, 66% responden menyatakan bahwa teks pantun mudah dibaca, dan 29% responden menyatakan bahwa teks pantun sangat mudah dibaca. Sehingga responden yang menilai bahwa teks pantun ini mudah dibaca dan sangat mudah dibaca sebanyak 95%. Nilai rata-rata tersebut membuktikan bahwa secara keseluruhan teks pantun yang dibangkitkan oleh sistem dapat dibaca dan dipahami dengan mudah.

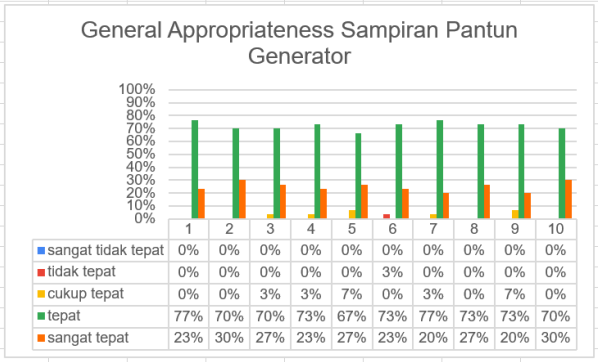


Gambar 12. Hasil pengujian *clarity*

Berdasarkan hasil pengujian *clarity* pada Gambar 12, dapat diketahui bahwa nilai rata-rata keseluruhan teks sampiran dari pantun ke 1 sampai ke 10 yaitu 7% responden menyatakan bahwa teks pantun cukup jelas, 68% responden menyatakan bahwa teks pantun jelas, dan 25% responden menyatakan bahwa teks pantun sangat jelas. Sehingga responden yang menilai bahwa teks pantun ini jelas dan sangat jelas sebanyak 93%. Nilai rata-rata tersebut membuktikan bahwa secara keseluruhan teks pantun yang dibangkitkan oleh

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021

sistem dapat memberikan informasi yang jelas.



	1	2	3	4	5	6	7	8	9	10
sangat tidak tepat	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
tidak tepat	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%
cukup tepat	0%	0%	3%	3%	7%	0%	3%	0%	7%	0%
tepat	77%	70%	70%	73%	67%	73%	77%	73%	73%	70%
sangat tepat	23%	30%	27%	23%	27%	23%	20%	27%	20%	30%

Gambar 13. Hasil pengujian *general appropriateness*

Berdasarkan hasil pengujian *general appropriateness* pada Gambar 12, dapat diketahui bahwa nilai rata-rata keseluruhan teks sampiran dari pantun ke 1 sampai ke 10 yaitu 2,5% responden menyatakan bahwa teks pantun cukup tepat, 72,5% responden menyatakan bahwa teks pantun tepat, dan 25% responden menyatakan bahwa teks pantun sangat tepat. Sehingga responden yang menilai bahwa teks pantun ini tepat dan sangat tepat sebanyak 97,5%. Nilai rata-rata tersebut membuktikan bahwa secara keseluruhan teks pantun yang dibangkitkan oleh sistem sudah tepat.

Dari ketiga aspek dapat disimpulkan bahwa sistem pembuat sampiran pantun otomatis berbasis *pattern-matching* ini sudah baik. Hal ini disebabkan karena sistem yang dibuat sudah diformulasikan mengikuti kaidah pantun secara jumlah larik maupun rimanya.

IV. KESIMPULAN

Metode yang diusulkan yaitu *pattern-matching* dapat digunakan untuk membuat teks sampiran pantun secara otomatis. Hal tersebut didukung dengan hasil evaluasi oleh pengguna yang memiliki nilai *naturalness* yang baik. Hal ini dibuktikan dengan hasil dari ketiga aspek evaluasi yaitu *readability, clarity, dan general approriateness* yang cukup tinggi masing-masing sebesar 95%, 93% dan 97,5%. Meskipun demikian, sistem ini masih memerlukan pemutakhiran seperti menggunakan *template* yang lebih *fleksible* sehingga hasil pembangkitan teks sampiran pantun lebih bervariasi

REFERENCES

[1] R. Setyadiharja, *Khazanah Negeri Pantun*. Deepublish, 2020.

[2] M. I. Rizky and T. Simarmata, "Peran Tradisi Berbalas Pantun dalam Acara Pesta Perkawinan Pada Masyarakat Melayu di Tanjung Pura," *Gondang J. Seni dan Budaya*, vol. 1, no. 2, pp. 91–99, 2017.

[3] A. Aslan and A. Yunaldi, "Budaya Berbalas Pantun Sebagai Media Penyampaian Pesan Perkawinan Dalam Acara Adat Istiadat Perkawinan Melayu Sambas," *J. Transform. (Islamic Stud.*, vol. 2, no. 2, pp. 111–122, 2018.

[4] D. Marfen, Y. Morelent, and R. Isnanda, "KATEGORI DAN FUNGSI PANTUN DALAM ACARA GEMPAR (GEMBIRA JO PARAMEX) DI RADIO ARBES FM PADANG," *J. Fak. Kegur. DAN ILMU Pendidik.*, vol. 4, no. 2, 2015.

[5] K. Hasanah and T. Sya'dian, "STRATEGI KREATIF DALAM PROGRAM ACARA DENDANG PANTUN TVRI SUMATERA UTARA DENGAN TEMA TAKKAN GOYAH DIPECAH BELAH," *J. Mhs. Fak. Seni dan Desain*, vol. 1, no. 1, pp. 280–292, 2020.

[6] D. E. Maulina, "Keanekaragaman Pantun Di Indonesia," *Semantik*, vol. 1, no. 1, pp. 107–121, 2015.

[7] H. Husamuddin, D. B. Prasetyo, and H. C. Rustamadji, "OTOMATISASI LAYANAN FREQUENTLY ASK QUESTIONS BERBASIS NATURAL LANGUGAE PROCESSING PADA TELEGRAM BOT," *Telemat. J. Inform. dan Teknol. Inf.*, vol. 17, no. 2, pp. 145–157, 2020.

[8] D. W. Sari and others, "Implementasi Natural Language Processing pada Chatbot Peribahasa Indonesia," 2018.

[9] M. Jamaluddin, N. Yuniarti, A. Rahmani, and J. Hutahaean, "Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia Menggunakan Algoritma K-Nearest Neighbor (Studi kasus MAN Cimahi)," *Pros. Ind. Res. Work. Natl. Semin.*, vol. 10, no. August 2019, pp. 314–324, Aug. 2020, doi: 10.35313/irwns.v10i1.1404.

[10] A. M. O. Dewi and B. Setiaji, "Pemanfaatan Sentence-similarity Measurement untuk proses pencarian pola pada chatbot berbasis pattern-matching," *SEMNASTEKNOMEDIA ONLINE*, vol. 2, no. 1, pp. 1–12, 2014.

[11] C. Rahmad, D. Puspitasari, and others, "Web Pembuat Puisi Otomatis Menggunakan Metode Monte Carlo," *J. Inform. Polinema*, vol. 1, no. 2, p. 19, 2015.

[12] O. S. Sitompul, E. B. Nababan, D. Arisandi, I. Aulia, and H. Wijaya, "Template-Based Natural Language Generation in Interpreting Laboratory Blood Test," *IAENG Int. J. Comput. Sci.*, vol. 48, no. 1, 2021.

[13] I. Aulia, S. Purnamawati, and J. Junianto, "Pembangkitan Interpretasi Tekstual Berbahasa Indonesia Berdasarkan Data Pemeriksaan Kimia Darah Menggunakan Pendekatan R-Template Based," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 1, 2020.

[14] H. N. Irmanda, R. Astriratma, and others, "Klasifikasi Jenis Pantun Dengan Metode Support Vector Machines (SVM)," *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 4, no. 5, pp. 915–922, 2020.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v10i3.1221, Copyright ©2020
Submitted : July 29, 2021; Revised : October 27, 2021; Accepted : October 27, 2021; Published : November 4, 2021