

PENERAPAN ALGORITMA TEXTRANK UNTUK AUTOMATIC SUMMARIZATION PADA DOKUMEN BERBAHASA INDONESIA

Eris¹, Viny Christanti M², Jeanny Pragantha³

Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara

Letjend S. Parman no. 1, Jakarta, Indonesia

Email : viny@untar.ac.id², jeanny@fti.untar.ac.id³

ABSTRACT

Automatic Summarization adalah sistem yang digunakan untuk meringkas dokumen secara otomatis. Ada beberapa algoritma untuk membangun sistem tersebut, dalam penelitian ini Automatic Summarization dibangun menggunakan algoritma TextRank. TextRank adalah algoritma peringkat berbasis grafik untuk memproses teks. TextRank menghasilkan ekstraksi kalimat sebagai ringkasan. Salah satu kelebihan dari algoritma ini, tidak diperlukannya pelatihan menggunakan data training pada algoritma yang digunakan. Perumusan dilakukan pada tahap berikut: preprocessing, hitung nilai kesamaan konten yang tumpang tindih, hitung nilai TextRank pada setiap kalimat, dan buat grafik. Hasilnya adalah teks ringkasan yang informatif. Automatic Summarization diuji dengan Q & A Evaluation yang diberikan kepada beberapa responden. Pengujian menunjukkan bahwa, algoritma ini mampu memberikan ringkasan dengan konten informatif hingga 82,48% untuk teks ringkasan 50% dan konten informatif 93,76% untuk teks ringkasan yang dirangkum 75%.

Kata Kunci: Automatic Summarization, Content Overlap Similarity, Graph-Based Ranking Algorithm, Sentence Extraction, TextRank algorithm

PENDAHULUAN

Kebutuhan akan informasi di saat ini merupakan suatu kebutuhan yang sangat penting untuk semua orang. Kebutuhan akan informasi tersebut dapat diperoleh dari berbagai media, seperti halnya media digital. Media digital dapat berupa artikel, *blog*, maupun situs-situs berita yang berisi informasi seperti dokumen (teks). Untuk dapat mengetahui informasi penting dari suatu dokumen, pembaca harus meluangkan banyak waktu. Oleh sebab itu, jika dokumen tersebut dapat diringkas oleh suatu sistem tanpa menghilangkan informasi yang penting maka pembaca dapat menghemat waktu. Karena pembaca dapat memahami dan mengetahui informasi penting dari dokumen tersebut tanpa harus membaca isi dokumen secara keseluruhan (Pratama, 2016).

Meringkas dokumen secara manual oleh manusia, membutuhkan banyak biaya dan waktu apabila dokumen tersebut banyak dan panjang sehingga diperlukan sistem peringkas otomatis (*automatic summarization*) untuk mengatasi banyaknya biaya dan waktu tersebut (Aristoteles, 2013). Sistem

peringkas otomatis yang dirancang harus efisien terhadap waktu dan efektif terhadap ketepatan penyajian informasi. Ada beberapa metoda dan algoritma yang dapat menghasilkan sistem tersebut. Pada penelitian ini akan dibuat sistem peringkas dokumen otomatis yang menggunakan algoritma *TextRank* sebagai metodenya dan juga dibuat berbasis *website* sehingga dapat digunakan oleh siapa saja yang memerlukannya. *TextRank* merupakan *graph-based ranking algorithm* (graf dengan model pemeringkatan) untuk pemrosesan teks (Mihalcea, R., & Tarau, P., 2004).

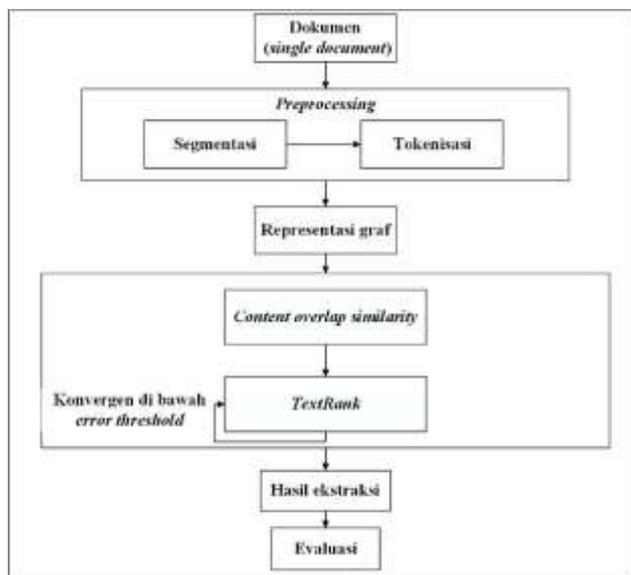
TextRank yang digunakan pada sistem ini adalah metode yang menghasilkan ekstraksi berupa kalimat (*TextRank for Sentence Extraction*). *TextRank* sangatlah fleksibel karena dapat digunakan pada berbagai bahasa tanpa mengubah algoritmanya. Hal ini dikarenakan *TextRank* tidak memerlukan data *training* untuk proses pengolahan dokumen (Mihalcea, R., & Tarau, P., 2004).

STUDI LITERATUR

Dalam melakukan proses peringkasan, algoritma *TextRank* terdiri dari beberapa tahap yaitu tahap *preprocessing*, tahap representasi graf dengan melakukan perhitungan nilai *content overlap similarity* dan nilai *TextRank*, serta tahap pemeringkatan kalimat. Skema mengenai tahapan yang dilakukan dalam perancangan sistem peringkasan otomatis ini dapat dilihat pada **Gambar 1**.

Pada **Gambar 1**, Pertama-tama, dokumen akan dimasukkan ke dalam sistem. Dokumen yang dimasukkan merupakan *single-document*. Kemudian di dalam sistem, dokumen tersebut dipecah menjadi kalimat-kalimat tunggal yang berdiri sendiri. Setiap kalimat akan direpresentasikan sebagai *vertex* dan kumpulan *vertex* inilah yang membangun sebuah graf. *Vertex* di dalam graf tersebut akan terhubung oleh *edges*. *Edges* ini didapatkan dari perhitungan nilai *similarity* antarkalimat.

Selanjutnya akan dicari nilai dari semua *vertex* dengan menggunakan algoritma *TextRank*. Setelah didapatkan nilai dari semua *vertex*, maka diurutkanlah nilai tersebut dari nilai yang tertinggi ke nilai yang terendah untuk menghasilkan daftar kalimat *top-rank*.



Gambar 1 Skema sistem peringkasan otomatis dengan Algoritma *TextRank*

Kalimat yang berada di daftar urutan teratas akan dipilih menjadi ringkasan ekstraktif untuk dokumen tersebut. Banyaknya kalimat yang akan menjadi hasil ringkasan tergantung dari persentase nilai

kompresi yang ditentukan oleh *user*. Dalam penelitian ini, akan dihasilkan ringkasan yang masih mengandung setengah kalimat yang ada dalam dokumen dan tiga perempat kalimat dari dokumen. Informasi dokumen semakin banyak yang hilang apabila hasil ringkasan hanya berisi seperempat kalimat dari seluruh dokumen. Sehingga *user* dapat memilih nilai kompresi 50% atau 75%.

A. *Preprocessing*

Preprocessing terdiri dari 2 tahap yaitu segmentasi dan tokenisasi. Pada tahap segmentasi, kalimat-kalimat dalam dokumen dipecah menjadi kalimat-kalimat tunggal. Pemecahan dokumen ini dilakukan dengan menggunakan *splitter*, yaitu berupa tanda baca titik (“.”), tanda seru (“!”), tanda tanya (“?”), dan *newline* (Pinandhita, 2013). Dalam proses segmentasi ini, gelar seseorang dan suatu singkatan harus dapat dikenali. Misalnya gelar pada nama seorang presiden Indonesia “Ir. H. Joko Widodo” dan pada nama spesies bahasa latin “*E .coli*”. Kemudian pada tahap tokenisasi, tiap kalimat akan dipecah menjadi kata-kata/frasa yang berdiri sendiri dan terpisah oleh spasi.

B. *TextRank* dan *Content Overlap Similarity*

TextRank merupakan *graph-based ranking algorithm* (graf dengan model pemeringkatan) untuk pemrosesan teks dari dokumen bahasa alami atau manusia. Dokumen yang diolah berupa dokumen tunggal (*single-document*) (Pinandhita, 2013). Terdapat dua jenis pengolahan bahasa dalam *TextRank*, yaitu *TextRank for keyword extraction* (ekstraksi kata kunci) dan *TextRank for sentence extraction* (ekstraksi kalimat).

Pada *TextRank for sentence extraction* akan dibangun sebuah graf yang berisi hubungan antarkalimat dalam dokumen. *Vertex* di dalam graf ini direpresentasikan sebagai unit satuan yang akan diberikan peringkat. *Vertex* ini mempunyai *similarity* yang dihubungkan oleh *edges*. Jenis *similarity* yang digunakan adalah *content overlap*. *Similarity* disini juga dapat ditentukan dengan menggunakan *cosine similarity*, tergantung dari kebutuhan sistem yang akan dibangun (Purwasih, 2008).

Content overlap antara dua kalimat didefinisikan sebagai jumlah kata yang sama (*word overlap*) antara kedua kata dan dinormalisasi dengan membagi jumlah *word overlap* dengan panjang tiap kalimat. Pembobotan tidak dilakukan karena nilai kesamaan antarkalimat langsung dihitung berdasarkan banyaknya kata yang sama

antarkalimat. (Pinandhita, 2013) Rumusnya adalah sebagai berikut: (Pinandhita, 2013)

$$Similarity(S_i, S_j) = \frac{|W_k| |W_k \in S_i \& W_k \in S_j|}{\log(|S_i|) + \log(|S_j|)} \dots\dots\dots(1)$$

Keterangan :

W_k = Jumlah kata (*term*) yang sama antara kalimat S_i dan S_j .

S_i = Panjang kalimat S_i .

S_j = Panjang kalimat S_j .

TextRank sendiri merupakan rumus yang berasal dari metode PageRank. Rumus pada metode PageRank ini telah diubah/dimodifikasi untuk kebutuhan meringkas suatu dokumen. Rumus dari PageRank adalah sebagai berikut: (Mihalcea, R., & Tarau, P., 2004)

$$s(v_i) = (1 - d) + d * \sum_{j \in in(v_i)} \frac{1}{|out(v_j)|} s(v_j) \dots\dots\dots(2)$$

Keterangan :

V_i = *vertex* yang dihitung skor-nya.

V_j = *vertex* yang bertetangga dengan V_i .

d = damping factor yang nilainya antara 0 dan 1, biasanya 0.85.

Graf yang dihasilkan oleh *TextRank* merupakan graf yang tidak mempunyai arah (*undirected*) dan berbobot (*weighted*) atau *undirected weighted graph*. Rumus dari *TextRank* adalah: (Mihalcea, R., & Tarau, P., 2004)

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in Adj(V_i)} \frac{w_{ji}}{\sum_{V_k \in Adj(V_j)} w_{jk}} WS(V_j) \dots\dots\dots(3)$$

Keterangan :

WS = *Weight Sentence* (Bobot Kalimat).

w = nilai *content overlap similarity*.

V_i = *vertex* yang dihitung skor-nya.

V_j = *vertex* yang bertetangga dengan V_i .

V_k = *vertex* yang bertetangga dengan V_j .

d = *damping factor* yang nilainya antara 0 dan 1, biasanya 0.85.

Dengan demikian, langkah-langkah untuk melakukan peringkasan dalam *TextRank*: (Purwasih, 2008)

1. Lakukan ekstraksi kalimat dengan menjadikan seluruh kalimat sebagai *vertex* dalam graf.
2. Identifikasi hubungan antarkalimat dengan membuat *edges* antara *vertex-vertex*. Dapat digunakan *content overlap similarity* untuk mengidentifikasinya.

3. Beri skor awal *vertex* untuk menentukan iterasi.
4. Lakukan iterasi algoritma *TextRank* sampai *error rate* tiap *vertex* konvergen di bawah *threshold*. *Error Rate* disini adalah perbedaan antar dua skor *vertex* yang dihitung pada iterasi yang berurutan dengan rumus:

$$S^{k+1}(V_i) - S^k(V_i) \dots\dots\dots(4)$$

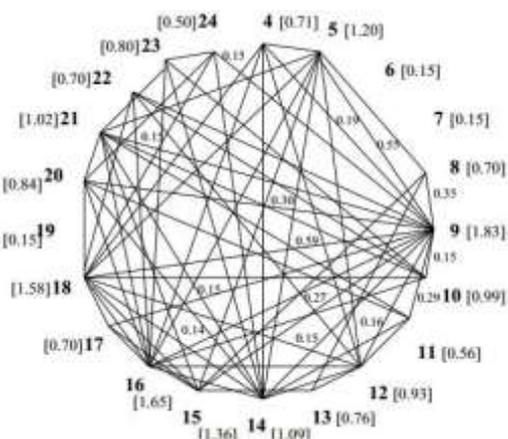
Keterangan :

V_i = *Vertex* yang dihitung skornya

k = pada iterasi ke- k

5. Setelah graf terbentuk, kemudia *vertex* diurutkan berdasarkan skor akhirnya. Lalu diambil *top-rank* sebagai hasil ekstraksi ringkasannya.

Pada Gambar 2 adalah contoh *graph-based ranking algorithm* yang dibangun dari dokumen yang terdiri dari 24 kalimat. Kalimat tersebut dipecah menjadi 24 kalimat tunggal dan tiap kalimat tunggal menjadi suatu *vertex* dalam graf. Selanjutnya akan dibentuk *edges* dengan melakukan perhitungan *similarity* untuk menghubungkan *vertex* sehingga terdapat nilai pada semua *edges* tersebut. *Edges* yang menghubungkan *vertex* menunjukkan hubungan *similarity*. *Similarity* digunakan untuk menyatakan hubungan antar-*vertex*, antarkata, dan antara kalimat yang satu dengan kalimat lainnya. Setelah dilakukan perhitungan *similarity*, maka akan dilakukan perhitungan menggunakan rumus *graph-based ranking algorithm* untuk mendapatkan nilai bobot tiap *vertex*. Setelah didapatkan nilai untuk setiap *vertex* dan *edges*, maka graf telah berhasil dibangun.



Gambar 2 Graf yang telah dibangun dari algoritma *TextRank*

Sumber : Rada Mihalcea, and Paul Tarau, *TextRank: Bringing order into texts*,
http://digital.library.unt.edu/ark:/67531/metadc30962/m2/1/high_res_d/Mihalcea-2004-TextRank-Bringing_Order_into_Texts.pdf, 20 Februari 2016

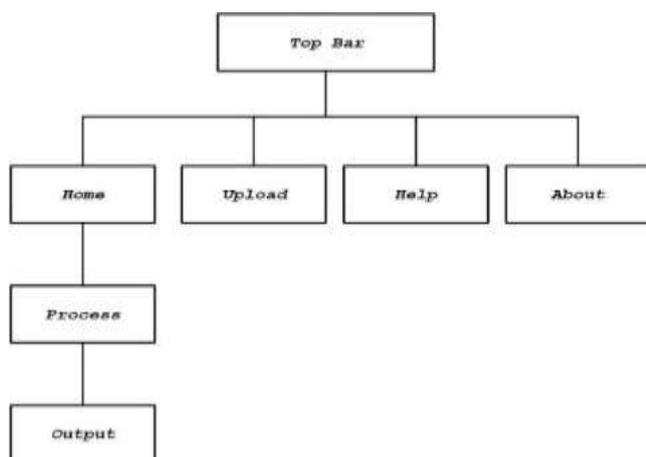
METHODOLOGI

A. Rancangan Sistem

Sistem yang dirancang adalah sistem peringkasan otomatis dokumen berbahasa Indonesia dengan algoritma *TextRank* berbasis *website*. Sistem menerima data berupa dokumen tunggal (*single-document*) dan menghasilkan ringkasan ekstraktif dari dokumen tersebut. Tahapan dari algoritma

TextRank sendiri adalah dengan membangun sebuah graf. Untuk membangun sebuah graf, sistem menerima *input* berupa dokumen teks. Graf dibangun dengan memecah dokumen menjadi kalimat-kalimat sebagai *vertex-vertex* yang saling terhubung oleh *edges*. *Edges* inilah yang merepresentasikan hubungan relasi antarkalimat. Pada tahap ini juga dilakukan perancangan struktur menu.

Perancangan struktur menu dilakukan dengan membuat diagram hirarki. Diagram hirarki menunjukkan hubungan hirarki modul (halaman web) pada suatu sistem. Tujuan dibuatnya diagram ini adalah untuk memudahkan dalam memberikan struktur gambaran proses yang terjadi pada sistem ini. Rancangan diagram hirarki ini dapat dilihat pada **Gambar 3**.



Gambar 3 Diagram hirarki

B. Pembuatan Sistem

Tahap pembuatan sistem dilakukan untuk merealisasikan rancangan yang sudah dibuat dengan membuat halaman-halaman situs web. Halaman-halaman situs web tersebut terdiri dari 6 halaman, yaitu:

1. Halaman *Home* (Awal)

Halaman *home* (awal) yang merupakan halaman utama (*homepage*) dari web ini yang memiliki hubungan dengan halaman-halaman lainnya. Halaman yang dibuat berisi 2 *textbox* (*textbox* judul dan *textbox* isi dokumen), 3 tombol (tombol *summarize*, tombol *clear*, tombol *process*, tombol *help*), dan 2 *radio button* (50% dan 75%) yang digunakan untuk menentukan panjangnya ringkasan. Tampilan dari halaman *home* ini dapat dilihat pada **Gambar 4**.



Gambar 4 Halaman *home*

2. Halaman *Upload*

Halaman *upload* merupakan halaman yang digunakan *user* untuk meringkas dokumen yang ada di dalam memori internal komputernya. *User* dapat mengunggah dokumen yang ingin diringkaskannya dalam bentuk format *.txt* atau *.word*. Di halaman ini juga terdapat tombol *summarize* dan tombol *process*. Terdapat pula 2 *radio button* (50% dan 75%) yang digunakan untuk menentukan panjangnya ringkasan. Di bawah *radio button* tersebut juga terdapat *checkbox* untuk menentukan apakah dokumen yang akan diringkaskan terdapat judul di dalam isi dokumen tersebut ataupun tidak. Tampilan dari halaman *upload* ini dapat dilihat pada **Gambar 5**.



Gambar 5 Halaman *upload*



Gambar 7 Halaman *output*

3. Halaman *Process*

Halaman *process* merupakan halaman yang digunakan untuk menampilkan proses langkah-langkah sistem dalam melakukan peringkasan. Di halaman ini, proses tersebut anatar lain, proses *input*, proses *preprocessing*, proses perhitungan *content overlap similarity*, proses perhitungan *TextRank*, proses pemeringkatan kalimat, dan proses hasil ringkasan. Terdapat pula 2 tombol *Next* pada bagian atas dan bagian halaman web yang digunakan untuk ke halaman selanjutnya, yaitu halaman *output*. Tampilan dari halaman *process* ini dapat dilihat pada **Gambar 6**.



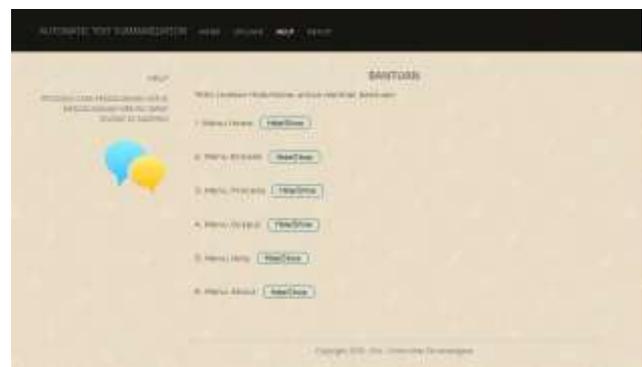
Gambar 6 Halaman *process*

4. Halaman *Output*

Halaman *output* merupakan halaman yang digunakan untuk menampilkan hasil ringkasan. Di halaman ini terdapat 2 *textbox* (*textbox* untuk menampilkan hasil ringkasan dan *textbox* untuk menampilkan dokumen yang belum diringkaskan), 4 tombol (tombol *save to pdf*, tombol *save to txt*, tombol *back to homepage* dan tombol *show/hide document*). Tampilan dari halaman *output* ini dapat dilihat pada **Gambar 7**.

5. Halaman *Help*

Halaman *help* berisi petunjuk cara penggunaan website ini bagi *user*. Tampilan dari halaman *help* ini dapat dilihat pada **Gambar 8**.



Gambar 8 Halaman *help*

6. Halaman *About*

Halaman *about* berisi tentang segala keterangan mengenai informasi pembuat dan pembuatan *website*. Tampilan dari halaman *about* ini dapat dilihat pada **Gambar 9**.



Gambar 9 Halaman *about*

HASIL DAN DISKUSI

Untuk mengetahui seberapa efektif sistem peringkasan otomatis ini dalam menghasilkan peringkasan, maka dilakukanlah pengujian. Pada pengujian sistem

peringkasan ini digunakan 50 dokumen tentang teknologi informasi yang diambil dari situs *tekno.kompas.com*. Masing-masing dokumen diringkaskan sehingga menghasilkan 2 macam ringkasan yaitu ringkasan dengan kompresi 50% dan ringkasan dengan kompresi 75%. Sehingga jumlah ringkasan yang perlu dievaluasi adalah sebanyak 100 ringkasan dokumen.

Tahapan pengujian ini dilakukan dengan memberikan kuisioner yang berisi hasil ringkasan dan daftar pertanyaan serta pilihan jawabannya kepada responden melalui sebuah tautan yang diberikan. Dengan kuisioner yang diberikan ini diharapkan dapat mengetahui seberapa akurat dan tepat sistem peringkasan yang telah dibangun untuk menghasilkan ringkasan ekstraktif.

Pengujian hasil ringkasan ini menggunakan metode *Q&A Evaluation*. Langkah pertama yang dilakukan oleh penguji dalam pengujian ini adalah membuat daftar pertanyaan yang dibuat dari dokumen asli yang belum diringkaskan dan memberikan pilihan jawabannya dengan pilihan benar atau salah. Pertanyaan yang dibuat untuk tiap dokumen sebanyak 5 pertanyaan.

Selanjutnya dokumen asli diringkaskan oleh sistem peringkasan. Dokumen yang diringkaskan pada pengujian ini adalah 50 dokumen berita teknologi informasi dan menghasilkan ringkasan sebanyak 100 buah dokumen dengan jumlah 50 buah dokumen dengan kompresi panjang ringkasan sebesar 50% dan 50 buah dokumen dengan kompresi panjang ringkasan sebesar 75%. Setiap ringkasan dokumen akan dibaca paling tidak oleh 5 orang responden untuk mendapatkan hasil yang lebih objektif.

Hasil ringkasan tersebut diberikan kepada para responden. Kemudian responden membaca hasil ringkasan yang telah diberikan dan selanjutnya baru menjawab pertanyaan yang diberikan sesuai dengan isi hasil ringkasan yang telah dibaca oleh responden. Responden menjawab pertanyaan dengan mengisi pada pilihan kotak benar dan salah yang telah disediakan.

Contoh dokumen berita yang digunakan untuk pengujian dapat dilihat pada **Gambar 10**. Dokumen berita tersebut terdiri dari 15 kalimat. Setelah dilakukan pemeringkatan menggunakan algoritma *TextRank*, didapatkan nilai *top-rank* dari setiap kalimat yang dapat dilihat pada **Tabel 1**.

Drone ini Bisa Terbang dan Menyelam

Drone atau pesawat tanpa awak yang diberi nama Cracuns ini punya kemampuan unik. Drone itu tak hanya sanggup terbang di udara, melainkan juga bisa menyelam di dalam air. Cracuns sendiri merupakan singkatan dari Corrosion Resistant Aerial Covert Unmanned Nautical System. Penciptanya adalah sekelompok peneliti dari John Hopkins University. Bodi Cracuns tidak dicetak dengan cara biasa. Para peneliti itu membuatnya menggunakan printer 3D agar murah dan mendapatkan bobot yang ringan. Sementara itu komponen-komponen elektronik yang jadi inti drone Cracuns diletakkan dalam kotak penyimpanan terpisah. Kotak ini terlindungi dari air meski drone dibawa menyelam. Dilansir KompasTekno dari The Verge, Rabu (23/3/2016), komponen lain yang ada di bagian luar mendapatkan lapisan khusus antikorosi. Lapisan ini mengamankannya dari korosi ketika digunakan di air laut. Kendat Cracuns masih berupa purwarupa, peneliti telah membuktikan bahwa drone tersebut bisa bertahan di bawah permukaan air selama dua bulan. "Teknisi di laboratorium Fisika Terapan John Hopkins University sudah lama mengerjakan proyek pesawat tanpa awak serta sistem milik kapal selam angkata laut," terang manajer proyek tersebut, Jason Stipes. Mereka tidak menjanjikan Cracuns bisa dipakai untuk berbagai jenis keperluan. Satu hal yang pasti, kemampuannya bertahan di bawah permukaan air membuat drone ini cocok untuk misi pengintaian. Berikut ini video singkat mengenai drone yang bisa menyelam sekaligus terbang di udara tersebut.

Gambar 10 Dokumen berita bersumber dari *kompas.com*

Sumber: Kompas.com, [Drone ini Bisa Terbang dan Menyelam](http://tekno.kompas.com/read/2016/03/24/07461397/Drone.ini.Bisa.Terbang.dan.Menyelam),
<http://tekno.kompas.com/read/2016/03/24/07461397/Drone.ini.Bisa.Terbang.dan.Menyelam>, 30 Mei 2016.

Tabel 1 Hasil pemeringkatan kalimat

Peringkat No.	Kalimat ke- (<i>Vertex</i>)	Nilai akhir <i>TextRank</i>
1	WS(V11)	1.519
2	WS(V14)	1.422
3	WS(V15)	1.412
4	WS(V2)	1.27
5	WS(V1)	1.244
6	WS(V8)	1.152
7	WS(V10)	1.127
8	WS(V7)	1.103
9	WS(V9)	0.92

10	WS(V12)	0.782
11	WS(V13)	0.685
12	WS(V3)	0.629
13	WS(V4)	0.622
14	WS(V6)	0.611
15	WS(V5)	0.503

Hasil ringkasan pengujian dari dokumen tersebut dengan panjang ringkasan 50% dan 75% dapat dilihat pada **Gambar 11** dan **Gambar 12**. Hasil ringkasan dengan panjang ringkasan 50% menghasilkan jumlah kalimat sebanyak 8 kalimat, sedangkan untuk panjang ringkasan 75% menghasilkan 12 kalimat.

Drone ini Bisa Terbang dan Menyelam

Drone atau pesawat tanpa awak yang diberi nama Cracuns ini punya kemampuan unik. Drone itu tak hanya sanggup terbang di udara, melainkan juga bisa menyelam di dalam air. Sementara itu komponen-komponen elektronik yang jadi inti drone Cracuns diletakkan dalam kotak penyimpanan terpisah. Kotak ini terlindungi dari air meski drone dibawa menyelam. Lapisan ini mengamankannya dari korosi ketika digunakan di air laut. Kendat Cracuns masih berupa purwarupa, peneliti telah membuktikan bahwa drone tersebut bisa bertahan di bawah permukaan air selama dua bulan. Satu hal yang pasti, kemampuannya bertahan di bawah permukaan air membuat drone ini cocok untuk misi pengintaian. Berikut ini video singkat mengenai drone yang bisa meyelam sekaligus terbang di udara tersebut.

Gambar 11 Contoh hasil ringkasan (*output*) 50% yang telah diproses

Drone ini Bisa Terbang dan Menyelam

Drone atau pesawat tanpa awak yang diberi nama Cracuns ini punya kemampuan unik. Drone itu tak hanya sanggup terbang di udara, melainkan juga bisa menyelam di dalam air. Cracuns sendiri merupakan singkatan dari Corrosion Resistant Aerial Covert Unmanned Nautical System. Sementara itu komponen-komponen elektronik yang jadi inti drone Cracuns diletakkan dalam kotak penyimpanan terpisah. Kotak ini terlindungi dari air meski drone dibawa menyelam. Dilansir KompasTekno dari The Verge, Rabu (23/3/2016), komponen lain yang ada di bagian luar mendapatkan lapisan khusus antikorosi. Lapisan ini mengamankannya dari korosi ketika digunakan di air laut. Kendat Cracuns masih berupa purwarupa, peneliti telah membuktikan bahwa drone tersebut bisa bertahan di bawah permukaan air selama dua bulan. "Teknisi di laboratorium Fisika Terapan John Hopkins University

sudah lama mengerjakan proyek pesawat tanpa awak serta sistem milik kapal selam angkata laut," terang manajer proyek tersebut, Jason Stipes. Mereka tidak menjanjikan Cracuns bisa dipakai untuk berbagai jenis keperluan. Satu hal yang pasti, kemampuannya bertahan di bawah permukaan air membuat drone ini cocok untuk misi pengintaian. Berikut ini video singkat mengenai drone yang bisa meyelam sekaligus terbang di udara tersebut.

Gambar 12 Contoh hasil ringkasan (*output*) 75% yang telah diproses

Rata-rata pertanyaan yang berhasil dijawab oleh responden dari daftar pertanyaan yang diberikan dapat dilihat pada **Tabel 2**.

Tabel 2 Hasil pengujian *Q&A Evaluation*

No. Responden	50%	75%
Responden ke-1	84.4%	94%
Responden ke-2	82.8%	93.6%
Responden ke-3	84.8	93.2%
Responden ke-4	81.2%	92.4%
Responden ke-5	79.2%	95.6%
Rata-rata persentase akhir	82.48%	93.76%

Hasil pengujian ringkasan yang dihasilkan oleh sistem menggunakan metode Q&A Evaluation adalah sebagai berikut.

1. Rata-rata pertanyaan yang dapat dijawab oleh responden untuk hasil ringkasan dengan panjang 50% adalah 84.4%. Dengan hanya membaca setengah dari dokumen awal, pengguna sistem ini mampu mengetahui informasi-informasi utama yang terdapat di dalam dokumen tersebut. Hal ini tentu akan menghemat waktu pengguna untuk membaca banyak dokumen.
2. Rata-rata pertanyaan yang dapat dijawab oleh responden untuk hasil ringkasan dengan panjang 75% adalah 93.76%. Hasil ringkasan dengan panjang 75% mempunyai informasi yang lebih banyak dibandingkan kompresi panjang ringkasan sebesar 50%. Hal ini dikarenakan kalimat yang dibuang pada dokumen asli lebih sedikit sehingga kandungan informasi yang dimuat lebih banyak.
3. Algoritma *TextRank* dapat membuat ringkasan dari dokumen berbahasa Indonesia. Kandungan informasi yang ada pada hasil ringkasan dapat tersajikan tanpa perlu melakukan *training* sehingga dari segi pembuatan sistem ini mampu menyajikan hasil ringkasan ekstraktif yang bersifat informatif. Informatif disini dapat diartikan sebagai hasil ringkasan yang disajikan

mengandung informasi-informasi utama yang terdapat dalam dokumen asli.

KESIMPULAN

Setelah melakukan percobaan, maka dapat disimpulkan bahwa:

1. Algoritma *TextRank* dapat mengambil kalimat menjadi hasil ringkasan jika kalimat tersebut mempunyai nilai *content overlap similarity* yang tinggi dibandingkan dengan kalimat-kalimat yang lainnya sehingga kalimat yang direpresentasikan sebagai *vertex* tersebut mempunyai banyak *edge* dan bernilai tinggi.
2. Hasil ringkasan dengan panjang 50% mampu menghasilkan informasi sebesar 82.48% sehingga dengan membaca setengah dari dokumen asli pengguna dapat mengetahui informasi yang terkandung dari dokumen tersebut sebanyak 82.48%. Hal ini tentu dapat menghemat banyak waktu pengguna dalam membaca suatu dokumen.
3. Hasil ringkasan dengan panjang 75% dapat menghasilkan informasi sebesar 93.76%, yang berarti hampir mendekati informasi keseluruhan yang terkandung dalam dokumen asli tersebut.

Saran kepada pihak yang ingin mengembangkan sistem peringkasan otomatis menggunakan algoritma *TextRank* ini lebih lanjut adalah:

1. Menggunakan metode *similarity* yang lain untuk mendapatkan nilai hubungan antar kalimat (nilai *edges*) seperti menggantikan *content overlap similarity* dengan *cosine similarity*.
2. Melakukan proses penghilangan *stopword* agar kata-kata yang tidak bermakna seperti kata penghubung, tidak diproses dalam perhitungan *content overlap similarity*.
3. Penggunaan evaluasi yang berbeda, seperti menggantikan *Q&A Evaluation* dengan *Kappa measure* (responden dengan kesepakatan beberapa ahli).

DAFTAR PUSTAKA

- Aristoteles, A. (2013). Penerapan Algoritma Genetika pada Peringkasan Teks Dokumen Bahasa Indonesia. *Prosiding SEMIRATA 2013*, 1(1).
- Mihalcea, R., & Tarau, P. (2004, July). *TextRank: Bringing order into texts*. Association for Computational Linguistics.
- Pinandhita, R. R. (2013). Peringkasan dokumen berbahasa Indonesia berbasis kata benda dengan BM25. Diakses 11 Februari 2016 dari <http://repository.ipb.ac.id/bitstream/handle/123456789/63836/G13rrr.pdf?sequence=1&isAllowed=y>.
- Pratama, I. S., Alam, G., & Tinaliah, T. Penerapan Algoritma Centroid-Based Summarization untuk Sistem Peringkasan Dokumen Berbahasa Indonesia. Diakses 02 Februari 2016 dari <http://eprints.mdp.ac.id/1248/1/Penerapan%20Algoritma%20CentroidBased%20Summarization2.pdf>
- Purwasih, N. (2008). Sistem Peringkasan Teks Otomatis untuk Dokumen Tunggal Berita Berbahasa Indonesia dengan Menggunakan Graph-based Summarization Algorithm dan Similarity. *Departemen Teknik Informatika, Institut Teknologi Telkom Bandung*. Diakses 02 Februari 2016, <https://repository.telkomuniversity.ac.id/pustaka/94907/peringkasan-teks-otomatis-dokumen-tunggal-berbahasa-indonesia-menggunakan-graph-based-summarization-algorithm-dansimilarity-studi-kasus-artikel-berita-.html>