Peringkasan Teks Ekstraktif pada Dokumen Tunggal Menggunakan Metode *Restricted Boltzmann Machine*

Rully Widiastutik, *Teknik Informatika AKNS*, Lukman Zaman P.C.S.W., *Teknologi Informasi iSTTS*, dan Joan Santoso, *Teknologi Informasi iSTTS*

***Abstrak*—Penelitian yang dilakukan yaitu menghasilkan peringkasan teks ekstratif secara otomatis yang dapat membantu menghasilkan dokumen yang lebih pendek dari dokumen aslinya dengan cara mengambil kalimat penting dari dokumen sehingga pembaca dapat memahami isi dokumen dengan cepat tanpa membaca secara keseluruhan. Dataset yang digunakan sebanyak 30 dokumen tunggal teks berita berbahasa Indonesia yang diperoleh dari www.kompas.com pada kategori tekno. Dalam penelitian ini, digunakan sepuluh fitur yaitu posisi kalimat, panjang kalimat, data numerik, bobot kalimat, kesamaan antara kalimat dan centroid, bi-gram, tri-gram, kata benda yang tepat, kemiripan antar kalimat, huruf besar. Nilai fitur setiap kalimat dihitung. Nilai fitur yang dihasilkan ditingkatkan dengan menggunakan metode *Restricted Boltzmann Machine* (RBM) agar ringkasan yang dihasilkan lebih akurat. Untuk proses pengujian dalam penelitian ini menggunakan ROUGE-1. Hasil yang diperoleh dalam penelitian yaitu dengan menggunakan *learning rate* 0.06 menghasilkan *recall*, *precision* dan *f-measure* tertinggi yakni 0.744, 0.611 dan 0.669. Selain itu, semakin besar nilai *compression rate* yang digunakan maka hasil *recall*, *precision* dan *f-measure* yang dihasilkan akan semakin tinggi. Hasil peringkasan teks dengan menggunakan RBM memiliki nilai *recall* lebih tinggi 2.1%, *precision* lebih tinggi 1.6% dan *f-measure* lebih tinggi 1.8% daripada hasil peringkasan teks tanpa RBM. Hal ini menunjukkan bahwa peringkasan teks dengan menggunakan RBM hasilnya lebih baik daripada peringkasan teks tanpa RBM.**

***Kata Kunci*—Natural Language Processing, Peringkasan Teks, Restricted Boltzmann Machine, Teks Berita Berbahasa Indonesia.**

I. PENDAHULUAN

Dengan perkembangan teknologi internet menyebabkan jumlah data terus meningkat sehingga menyebabkan informasi yang tersedia semakin banyak. Salah satu informasi yang banyak ditemui di media online adalah artikel berita. Artikel berita merupakan salah satu dokumen berbasis teks. Dengan membaca artikel berita, pembaca dapat mengetahui informasi mengenai sesuatu yang sedang terjadi atau yang telah terjadi. Pembaca dapat mengalami kesulitan serta membutuhkan waktu yang lama dalam memahami isi dokumen apabila dokumen yang akan dibaca banyak dan panjang. Hal ini dapat menyebabkan berkurangnya minat seseorang dalam membaca. Namun apabila terdapat ringkasan dari sebuah teks atau dokumen akan membantu memahami isi dokumen dengan cepat tanpa harus membaca secara keseluruhan. Ringkasan adalah dokumen yang berisi informasi penting atau intisari dari dokumen asli yang dihasilkan dari satu atau lebih dokumen. Ringkasan dokumen yang dibuat secara manual akan membutuhkan waktu lama apabila dokumen yang akan diringkas banyak dan panjang. Oleh karena itu dalam penelitian ini diterapkan suatu sistem peringkasan teks secara otomatis yang dapat membantu dalam penyusunan kalimat mengenai intisari dari dokumen secara cepat[1].

Pada peringkasan teks terdapat dua pendekatan, yang pertama adalah pendekatan ekstraktif. Pendekatan ini menghasilkan ringkasan dengan cara mengambil kalimat-kalimat penting dari dokumen asli dan menyusunnya menjadi dokumen yang lebih pendek. Yang kedua adalah pendekatan abstraktif. Pendekatan ini menghasilkan ringkasan dimana kalimat yang tersusun tidak ada dalam dokumen aslinya[1]. Terdapat dua kategori peringkasan teks yaitu peringkasan dokumen tunggal, dimana peringkasan diekstrak dari satu dokumen dan peringkasan multi dokumen, dimana peringkasan diekstrak dari beberapa dokumen yang saling berhubungan[2].

II. PENELITIAN PENUNJANG

Bab ini terdapat penjelasan dari beberapa jurnal ilmiah yang digunakan untuk perbandingan serta sebagai penunjang pada penelitian ini.

Penelitian yang dilakukan oleh Suputra (2017) yaitu menghasilkan ringkasan berdasarkan skor fitur-fitur penting dari kalimat pada sebuah dokumen. Fitur yang digunakan sebanyak tiga fitur, pertama adalah fitur *keyword* positif, kedua adalah fitur kemiripan antar kalimat, dan ketiga adalah fitur kemiripan kalimat dengan judul dokumen. Dari ketiga fitur tersebut yang memberikan pengaruh yang dominan adalah fitur kemiripan antar kalimat. Data yang digunakan pada saat percobaan yaitu artikel Bahasa Bali yang didapat dari berbagai sumber[3].

Penelitian yang dilakukan oleh Gotami, dkk (2018) yaitu menghasilkan ringkasan yang memiliki makna umum atau luas dengan menggunakan metode *Latent Semantic Analysis*

Rully Widiastutik, Teknik Informatika, Akademi Komunitas Negeri Sumenep, Sumenep, Jawa Timur, Indonesia (e-mail: rullywidiastutik@gmail.com)

Lukman Zaman P.C.S.W, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: lz@stts.edu)

Joan Santoso, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: joan@stts.edu)

(LSA). Digunakannya metode tersebut karena mampu mengekstrak struktur semantik atau makna yang tersembunyi pada sebuah kalimat. Untuk penyusunan urutan ringkasan pada tahap ekstraksi ringkasan menggunakan *Cross method LSA*. Data yang digunakan untuk pengujian yaitu 10 artikel berita kesehatan berbahasa Indonesia[4].

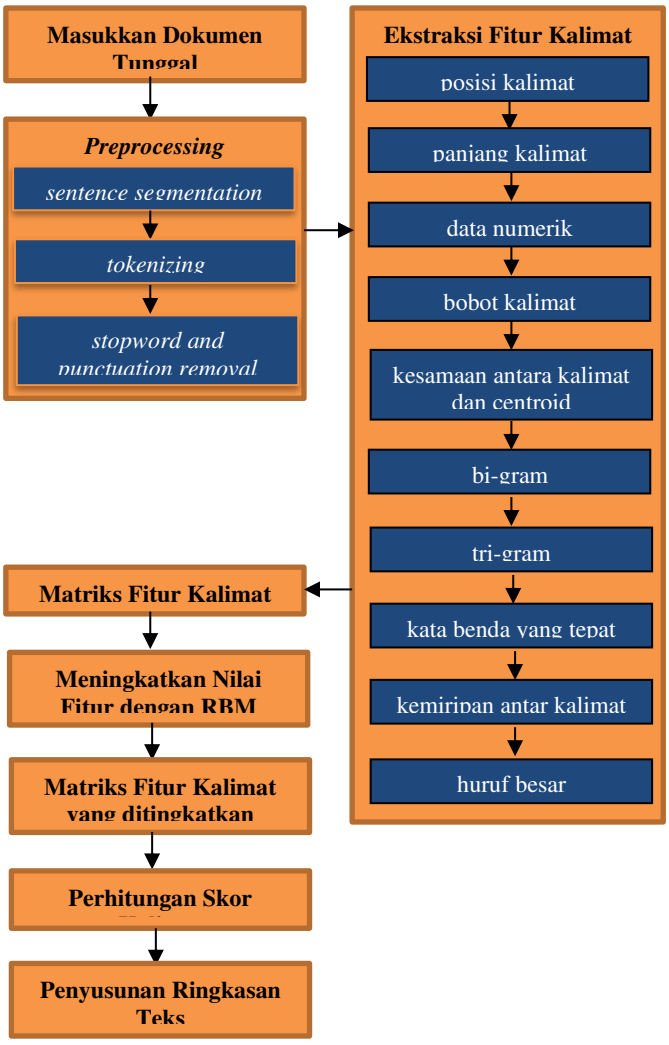
Penelitian yang dilakukan oleh Ambekar, dkk (2018) yaitu menghasilkan peringkasan teks menggunakan *Restricted Boltzmann Machine* (RBM). Terdapat lima fitur yang digunakan yaitu kesamaan judul, posisi kalimat, bobot *term*, panjang kalimat, skor kata benda yang tepat. Dataset yang digunakan merupakan data input yang tidak berlabel. RBM merupakan algoritma *deep learning* tanpa pengawasan. Dalam penelitiannya menjelaskan bahwa ringkasan dengan menggunakan algoritma *deep learning* tanpa pengawasan hasilnya lebih baik daripada ringkasan dengan menggunakan teknik pembelajaran terawasi karena tidak ada dataset pelatihan yang diperlukan. Sehingga metode tersebut lebih efisien[5].

Penelitian yang dilakukan oleh Elgamal, dkk (2019) yaitu menghasilkan sistem peringkasan teks otomatis pada teks arab. Dalam penelitiannya melakukan perbandingan antara algoritma *deep learning* menggunakan RBM dan algoritma *clustering* menggunakan LSA. Setelah dilakukan pengujian menunjukkan bahwa dengan algoritma *deep learning* menggunakan RBM memberikan hasil yang lebih baik dalam peringkasan teks arab[6].

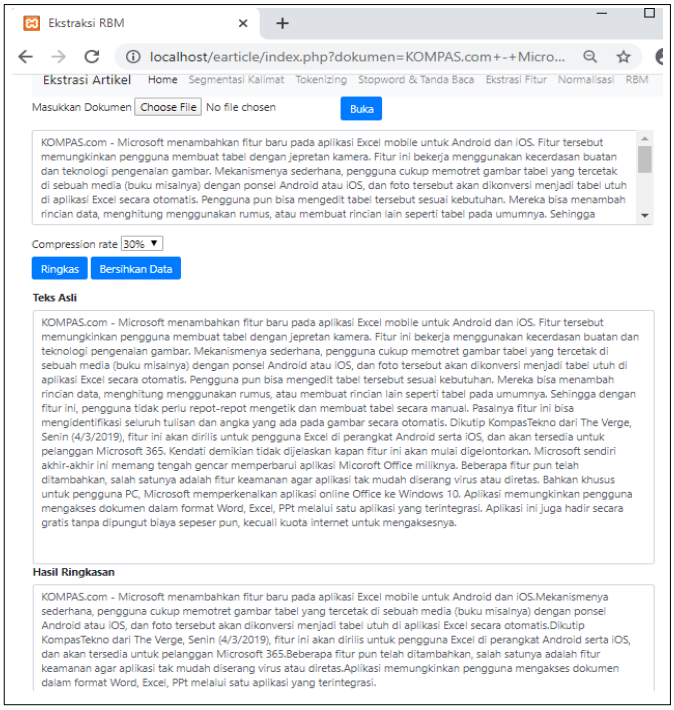
Dari penelitian-penelitian tersebut maka dalam penelitian ini menghasilkan peringkasan teks ekstraktif secara otomatis pada dokumen tunggal menggunakan metode *Restricted Boltzmann Machine* (RBM). Peringkasan teks secara otomatis yang dihasilkan mampu menyusun dokumen dalam bentuk yang lebih pendek dari dokumen asli dengan cara mengambil kalimat penting dokumen asli tersebut. Terdapat sepuluh fitur yang digunakan dalam penelitian ini. Digunakan metode RBM karena dapat memberikan hasil yang lebih baik dalam peringkasan. Dalam penelitian ini, metode RBM digunakan untuk meningkatkan nilai fitur setiap kalimat sehingga ringkasan yang dihasilkan lebih akurat[7].

Gambar 1 merupakan arsitektur sistem dalam penelitian ini. Dokumen teks yang diinputkan dilakukan *preprocessing* untuk menghasilkan *Term Index* yang dapat digunakan pada tahap selanjutnya. Tahap selanjutnya adalah perhitungan nilai fitur setiap kalimat. Terdapat sepuluh fitur yang digunakan. Matriks fitur kalimat dibentuk. Nilai fitur kalimat yang telah dihitung ditingkatkan menggunakan metode RBM. Matriks fitur kalimat yang ditingkatkan dibentuk. Dilakukan perhitungan skor setiap kalimat pada dokumen. Kalimat yang termasuk dalam ringkasan yaitu kalimat yang memiliki urutan teratas berdasarkan skor sesuai *compression rate* yang ditentukan.

Gambar 2 merupakan sistem peringkasan teks ekstraktif pada dokumen artikel berita berbahasa Indonesia menggunakan metode *Restricted Boltzmann Machine*. Pada sistem tersebut user menginput dokumen berupa isi dari artikel berita. Kemudian sistem akan mengolah dokumen tersebut dan akan menampilkan hasil ringkasan sesuai panjang ringkasan yang ditentukan oleh user.



Gambar. 1. Arsitektur Sistem



Gambar. 2. Sistem Peringkasan Dokumen Berita Menggunakan RBM

III. METODOLOGI

A. *Preprocessing*

Tahap awal pada pemrosesan peringkasan teks yaitu *preprocessing*. Tujuan dilakukan *preprocessing* yaitu untuk menghasilkan *Term Index* dari dokumen teks sehingga dapat digunakan pada tahap selanjutnya[4]. Tahapan *preprocessing* yang digunakan pada penelitian ini meliputi tahapan *sentence segmentation* adalah proses pemisahan kalimat dengan pembatas karakter titik yang diikuti oleh karakter spasi menjadi komponen terpisah. Tahapan selanjutnya adalah *tokenizing*, pada tahapan ini kumpulan kalimat hasil segmentasi dilakukan proses pemisahan setiap kata untuk menjadi data tunggal. Yang terakhir adalah tahapan *stopword and punctuation removal* adalah proses penghapusan kata umum untuk mengurangi jumlah kemunculan kata yang kurang mempunyai arti seperti "dan", "atau", "adalah", "di", "lalu" dan lain-lain serta proses penghapusan tanda baca kecuali titik karena sebagai pembatas akhir kalimat.

B. *Ekstraksi Fitur kalimat*

Setelah dilakukan tahap *preprocessing*, tahap selanjutnya adalah perhitungan nilai fitur setiap kalimat. Pada penelitian ini digunakan sebanyak sepuluh fitur yaitu :

1. Fitur Posisi Kalimat

Posisi kalimat adalah letak kalimat dalam sebuah dokumen. Relevansi kalimat dapat diketahui berdasarkan posisinya dalam dokumen. Pada kalimat pertama atau terakhir dari dokumen selalu penting dan memiliki informasi maksimal. Sehingga untuk kalimat pertama atau terakhir, nilai fiturnya yaitu bernilai 1. Sedangkan untuk kalimat lainnya dapat dihitung menggunakan persamaan (1)[8].

$$PK = \frac{N-P}{N} \quad (1)$$

Pada persamaan (1), variabel *PK* adalah hasil perhitungan nilai fitur posisi kalimat untuk kalimat selain kalimat pertama atau terakhir. *N* adalah total jumlah kalimat dalam dokumen dan *P* adalah posisi kalimat dalam dokumen.

2. Fitur Panjang Kalimat

Kalimat yang sangat singkat tidak mengandung banyak informasi. Apabila jumlah kata dalam kalimat kurang dari 3 maka nilai fiturnya bernilai 0. Sedangkan untuk yang lainnya dalam menentukan kalimat penting berdasarkan panjang kalimatnya dapat dihitung menggunakan persamaan (2)[7],[8].

$$Panjang\_Kalimat_i = \frac{jumlah\ kata\ pada\ kalimat\ i}{jumlah\ kata\ pada\ semua\ kalimat} \quad (2)$$

Pada persamaan (2), variabel $Panjang\_Kalimat_i$ adalah hasil perhitungan nilai fitur panjang kalimat pada kalimat ke-*i*. Untuk menghitung nilai fitur panjang kalimat pada kalimat ke-*i* yaitu jumlah kata pada kalimat ke-*i* dibagi dengan jumlah kata pada semua kalimat.

3. Fitur Data Numerik

Kalimat yang berisi data berupa bilangan juga dipertimbangkan dalam ringkasan karena data berupa bilangan dapat mempresentasikan suatu nilai penting pada dokumen. Fitur data numerik dapat dihitung menggunakan persamaan (3)[8].

$$Datanumerik_i = \frac{jumlah\ data\ numerik\ pada\ kalimat\ i}{jumlah\ kata\ pada\ kalimat\ i} \quad (3)$$

Pada persamaan (3), variabel $DataNumerik_i$ adalah hasil perhitungan nilai fitur data numerik pada kalimat ke-*i*. Untuk menghitung nilai fitur data numerik pada kalimat ke-*i* yaitu jumlah data numerik pada kalimat ke-*i* dibagi dengan jumlah kata pada kalimat ke-*i*. Data numerik yaitu data yang berupa bilangan.

4. Fitur Bobot Kalimat

Pada bobot kalimat menggunakan perhitungan tf-isf (*term frequency-inverse sentence frequency*). Tf-isf didapat dari frekuensi kemunculan kata pada kalimat beserta dengan jumlah dari frekuensi banyak kata yang muncul dalam sebuah dokumen. Frekuensi kemunculan kata pada kalimat dinyatakan dengan tf, jumlah kalimat yang terdapat dalam sebuah dokumen dinyatakan dengan N, dan jumlah kalimat yang mengandung kata dinyatakan dengan n. Fitur bobot kalimat dapat dihitung menggunakan persamaan (4)[9].

$$Bobot\_Kalimat_i = tf_i\ x\ isf_i = tf_i\ x\ \left(log\ \frac{N}{n_i}\right) \quad (4)$$

Pada persamaan (4), variabel $Bobot\_Kalimat_i$ adalah hasil perhitungan nilai fitur bobot kalimat pada kalimat ke-*i*. tf_i adalah frekuensi kemunculan kata pada kalimat. *N* adalah jumlah kalimat dalam satu dokumen. n_i adalah jumlah kalimat dimana kata tersebut muncul.

5. Fitur Kesamaan Antara Kalimat dan Centroid

Kalimat yang fitur tf-isf maksimal digunakan sebagai kalimat centroid. Kesamaan antara kalimat dan centroid dapat dihitung menggunakan *cosine similarity* seperti pada persamaan (5)[8].

$$Cos_{sim_i} = cos(\text{kalimat}_i\ , \text{centroid}) = \frac{kalimat_i\ .\ centroid}{||kalimat_i||\ ||centroid||} \quad (5)$$

Pada persamaan (5), variabel Cos_{sim_i} adalah hasil perhitungan nilai fitur kesamaan antara kalimat ke-*i* dengan kalimat centroid. Variabel $kalimat_i$ adalah vektor kalimat ke-*i*, yang akan dibandingkan kemiripannya. Variabel *centroid* adalah vektor kalimat centroid, yang akan dibandingkan kemiripannya. $kalimat_i\ .\ centroid$ adalah *dot product* antara vektor kalimat ke-*i* dan vektor kalimat centroid. $||kalimat_i||$ adalah panjang vektor kalimat ke-*i*. $||centroid||$ adalah panjang vektor kalimat centroid. $||kalimat_i||\ ||centroid||$ adalah *cross product* antara $||kalimat_i||$ dan $||centroid||$.

6. Fitur Bi-gram

Bi-gram adalah dua kata yang berdekatan pada setiap kalimat dalam dokumen. Nilai fitur Bi-gram dapat dihasilkan dengan cara menghitung jumlah total Bi-gram pada setiap kalimat[8].

7. Fitur Tri-gram

Tri-gram adalah tiga kata yang berdekatan pada setiap kalimat dalam dokumen. Nilai fitur Tri-gram dapat dihasilkan dengan cara menghitung jumlah total Tri-gram pada setiap kalimat[8].

8. Fitur Kata Benda yang Tepat

Kalimat memberi arti penting jika kalimat tersebut memiliki kata benda yang tepat. Kata benda yang tepat adalah kata yang digunakan untuk mengklasifikasikan orang, tempat, atau benda. Fitur kata benda yang tepat dapat dihitung menggunakan persamaan (6)[8].

$$KB_i = \frac{jumlah\ kata\ benda\ yang\ tepat\ pada\ kalimat\ i}{jumlah\ kata\ pada\ kalimat\ i} \quad (6)$$

Pada persamaan (6), variabel KB_i adalah hasil perhitungan nilai fitur kata benda yang tepat pada kalimat ke-*i*. Untuk

menghitung nilai fitur kata benda yang tepat pada kalimat ke-*i* yaitu jumlah kata benda yang tepat pada kalimat ke-*i* dibagi dengan jumlah kata pada kalimat ke-*i*.

9. Fitur Kemiripan Antar Kalimat

Kemiripan antar kalimat yaitu kata yang muncul dalam kalimat sama dengan kata yang muncul dalam kalimat lain. Kemiripan antar kalimat didapat dari jumlah jarak kedua kalimat dibagi jumlah maksimum jarak kedua kalimat. Fitur kemiripan antar kalimat dapat dihitung menggunakan persamaan (7)[9].

$$KK = \frac{\sum Sim(Si,Sj)}{Max(\sum Sim(Si,Sj))} \quad (7)$$

Pada persamaan (7), variabel *KK* adalah hasil perhitungan nilai fitur kemiripan antar kalimat. $Sim(S_i,S_j)$ adalah kemiripan antar kalimat ke-*i* dan kalimat ke-*j*. $\sum Sim(Si,Sj)$ adalah jumlah kemiripan antar kalimat ke-*i* dan kalimat ke-*j*. Dengan S_i,S_j diperoleh dari persamaan (8) jarak antar kalimat.

$$S_i,S_j = \frac{\sum_{t=1}^{n} w_{ti} \, x \, w_{tj}}{\sqrt{\sum_{t=1}^{n} w_{ti}^2} \; x \; \sqrt{\sum_{t=1}^{n} w_{tj}^2}} \quad (8)$$

Pada persamanan (8), w_{ti} adalah bobot kata t pada kalimat *i*. w_{tj} adalah bobot kata t pada kalimat *j*. *n* adalah jumlah kata dalam kalimat. $\sum_{t=1}^{n} w_{ti} \, x \, w_{tj}$ adalah jumlah perhitungan skalar dari pembobotan kalimat. $\sqrt{\sum_{t=1}^{n} w_{ti}^2} \; x \; \sqrt{\sum_{t=1}^{n} w_{tj}^2}$ adalah jumlah perhitungan vektor dari pembobotan kalimat.

10. Fitur Huruf Besar

Kalimat yang berisi kata yang mengandung huruf besar mempunyai peluang dalam ringkasan karena kata yang mengandung huruf besar termasuk dalam kata yang dipentingkan pada dokumen. Fitur huruf besar dapat dihitung menggunakan persamaan (9)[10].

$$HB = \frac{banyak\ kata\ yang\ terdapat\ huruf\ besar}{banyak\ kata\ dalam\ kalimat} \quad (9)$$

Pada persamaan (9), variabel *HB* adalah hasil perhitungan nilai fitur huruf besar pada setiap kalimat. Untuk menghitung nilai fitur huruf besar pada setiap kalimat yaitu banyak kata yang terdapat huruf besar dalam kalimat dibagi dengan banyak kata dalam kalimat.

C. *Matriks Fitur Kalimat*

Setelah setiap kalimat nilai fiturnya dihitung, selanjutnya matriks fitur kalimat dibentuk yang berupa matriks 2-d yang terdiri dari S = (s_1, s_2, s_3, …, s_N) adalah kalimat dalam dokumen dan S_i = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}) adalah vektor fitur.

D. *Restricted Boltzmann Machine (RBM)*

Restricted Boltzmann Machine (RBM) merupakan model generatif probabilistik yang mampu secara otomatis mengekstrak fitur input data dengan menggunakan algoritma pembelajaran tanpa pengawasan. RBM adalah jaringan saraf yang bersifat *stochastic*. Jaringan saraf berarti memiliki unit neuron berupa aktivasi biner yang bergantung pada neuron-neuron yang saling terhubung. Sedangkan *stochastic* berarti aktivasi yang memiliki unsur probabilistik. RBM terdiri dari dua binary unit yaitu *visible layer* dan *hidden layer* serta unit bias. *Visible layer* merupakan state yang akan diobservasi dan *hidden layer* merupakan *feature detectors.* Masing-masing *visible* unit terhubung ke semua *hidden* unit yang diwakili oleh array bobot, sehingga masing-masing *hidden* unit juga terhubung ke semua *visible* unit dan unit bias terhubung ke semua *visible* unit dan semua *hidden* unit. Untuk memudahkan proses pembelajaran, jaringan dibatasi sehingga tidak ada *visible* unit terhubung ke *visible* unit lain dan *hidden* unit terhubung ke *hidden* unit lain.

Untuk melatih RBM, sampel dari *training set* yang digunakan sebagai input untuk RBM melalui neuron visible, dan kemudian jaringan sampel bolak balik antara neuron *visible* dan *hidden*. Tujuan dari pelatihan yaitu untuk pembelajaran koneksi bobot pada *visible* atau *hidden* dan bias aktivasi neuron sehingga RBM belajar untuk merekonstruksi data input selama fase dimana sampel neuron *visible* dari neuron *hidden*. Setiap proses sampling pada dasarnya berupa perkalian matriks antara sekumpulan sampel pelatihan dan matriks bobot, diikuti dengan fungsi aktivasi neuron yaitu fungsi sigmoid. Sampling antara lapisan *hidden* dan *visible* diikuti oleh modifikasi parameter (dikontrol oleh *learning rate*) dan diulang untuk setiap kelompok data dalam *training set*, dan untuk state sebanyak yang diperlukan untuk mencapai konvergensi[11].

Unit *hidden* diinisialisasi dan diperbaharui menggunakan persamaan (10), dimana H_j dari setiap unit *hidden* j dengan nilai V (binari state unit visible) diatur satu dengan probabilitas :

$$p(H_j = 1 \mid V) = \sigma(b_j + \sum_i V_i W_{ij}) \quad (10)$$

Dimana $\sigma(x)$ adalah fungsi sigmoid

$$\sigma(x) = \frac{1}{1+exp^{(-x)}} \quad (11)$$

Pada persamaan (10), b_j adalah bias dari unit *hidden*. V_i adalah binari state dari *visible* unit i. W_{ij} adalah bobot antara unit *visible* dan unit *hidden*. i (i = 1,2,3,..,n) untuk mewakili jumlah dari *visible* neuron. Sedangkan pada persamaan (11), x merupakan hasil perhitungan dari $b_j + \sum_i V_i W_{ij}$.

Unit *visible* diinisialisasi dan diperbaharui menggunakan persamaan (12), dimana V_i dari setiap unit *visible* i dengan nilai H (binari state unit hidden) diatur satu dengan probabilitas :

$$p(V_i = 1 \mid H) = \sigma(a_i + \sum_j H_j W_{ij}) \quad (12)$$

Pada persamaan (12), a_i adalah bias dari unit *visible*. H_j adalah binari state dari *hidden* unit j. W_{ij} adalah bobot antara unit *visible* dan unit *hidden*. j (j = 1,2,3,..,n) untuk mewakili jumlah dari *hidden* neuron. σ adalah fungsi sigmoid yang dihitung dengan menggunaan persamaan (11).

Metode RBM digunakan dalam penelitian ini karena mampu meningkatkan nilai fitur sehingga ringkasan yang dihasilkan lebih akurat. Matriks fitur kalimat yang diperoleh dari proses sebelumnya dilakukan normalisasi dengan membagi setiap elemennya dengan elemen tertinggi, selanjutnya digunakan sebagai input RBM. Berikut adalah tahapan-tahapan menggunakan RBM[11] :

1. Inisialisasi data
   a. Lakukan proses pencarian nilai bobot dan bias dengan nilai random yang kecil.
   b. Tentukan *Learning Rate* dan maksimum Epoch yang akan digunakan.
   c. Selama Epoch < MaksimumEpoch, lakukan langkah dibawah ini.

d. Selama datasampel < maksimumdatasampel, lakukan langkah dibawah ini.

2. Fase Positif
Setelah tahap inisialisasi data, selanjutnya adalah tahap fase positif. Fase positif yaitu mengambil data dan sampel dari *hidden* unit.

a. Menghitung energi aktivasi, probabilitas dan state dari unit *hidden* dengan menggunakan persamaan (10).

b. Menghitung positif_assosiatif dengan menggunakan persamaan (13). Untuk menghitung positif assosiatif yaitu matriks data sampel dari *visible* neuron ditranspose dikali dengan probabilitas yang dihasilkan dari langkah 2(a).

$$\text{Pos\_Asso} = (data)^T * P(H_j) \quad (13)$$

Pada persamaan (13), $(data)^T$ merupakan matriks data sampel dari *visible* neuron yang ditranspose. $P(H_j)$ merupakan probabilitas unit *hidden* yang dihasilkan dari langkah 2(a).

3. Fase Negatif
Tahap selanjutnya adalah fase negatif, yaitu merekonstruksi *visible* unit dan data sampel dari *hidden* unit.

a. Menghitung energi aktivasi dan probabilitas dari unit *visible* dengan menggunakan persamaan (12).

b. Melakukan langkah 2(a) untuk update *hidden* unit.

c. Menghitung negatif_assosiatif dengan menggunakan persamaan (14). Untuk menghitung negatif assosiatif yaitu matriks data (probabilitas dari unit *visible* yang diperoleh dari langkah 3(a)) yang ditranspose dikali dengan probabilitas dari unit *hidden* yang dihasilkan dari langkah 3(b).

$$\text{Neg\_Asso} = (data)^T * P(H_j) \quad (14)$$

Pada persamaan (14), $(data)^T$ merupakan matriks data (probabilitas dari unit *visible* yang diperoleh dari langkah 3(a)) yang ditranspose. $P(H_j)$ merupakan probabilitas unit *hidden* yang dihasilkan dari langkah 3(b).

4. Update Bobot
Tahap selanjutnya adalah update bobot, dengan menggunakan persamaan (15).

$$W_{ij}\,(\text{baru}) = W_{ij}(\text{lama}) + \Delta W_{ij} \quad (15)$$

$$\Delta Wij = \varepsilon\,(\text{Pos\_Asso} - \text{Neg\_Asso}) \quad (16)$$

Pada persamaan (15), W_{ij} adalah bobot antara unit *visible* dan unit *hidden.* ΔW_{ij} adalah perubahan bobot. Untuk menghitung bobot yang baru yaitu bobot lama ditambah dengan perubahan bobot yang dihasilkan. Sedangkan pada persamaan (16), merupakan perhitungan perubahan bobot. ε adalah *learning rate*. Perhitungan perubahan bobot ΔW_{ij} diperoleh dari *learning rate* dikali dengan selisih antara positif assosiatif dan negatif assosiatif.

5. Hitung Error
Untuk menghitung error menggunakan persamaan (17).

$$\text{Error} = \frac{1}{2}\sum_{i=1}^{p}(O_i - t_i)^2 \quad (17)$$

Pada persamaan (17), Error dihitung dengan pengurangan O_i merupakan data sampel dan t_i merupakan *visible* probabilitas yang dihasilkan dari fase negatif pada langkah 3(a). p adalah jumlah data.

Dalam penelitian ini, pada *visible* layer terdapat sepuluh neuron karena terdapat sepuluh fitur. *Learning rate* yang digunakan yaitu 0.01, 0.03, 0.06 dan 0.09 serta pembatasan epoch sebanyak 50 epoch.

E. Matriks Fitur Kalimat yang ditingkatkan

Setelah setiap kalimat nilai fiturnya ditingkatkan dengan menggunakan metode RBM, maka diperoleh matriks fitur kalimat yang ditingkatkan.

F. Perhitungan Skor Kalimat

Setelah dihasilkan matriks fitur kalimat yang ditingkatkan, selanjutnya adalah menghitung skor setiap kalimat pada dokumen. Skor kalimat diperoleh dengan cara menghitung nilai total dari nilai fitur yang telah ditingkatkan pada setiap kalimat.

G. Penyusunan Ringkasan Teks

Kalimat diurut secara *descending* berdasarkan skor kalimat. Untuk menentukan kalimat dalam ringkasan yaitu kalimat yang memiliki urutan teratas sesuai *compression rate* yang ditentukan. *Compression rate* yang digunakan dalam penelitian ini yaitu 20%, 30% dan 40%. Kalimat dalam ringkasan diurutkan sesuai dengan posisi asli dalam dokumen.

IV. Dataset Artikel Berita Berbahasa Indonesia

Dataset yang digunakan dalam penelitian ini yaitu dokumen tunggal teks berita berbahasa Indonesia yang diperoleh dari www.kompas.com pada kategori tekno. Artikel berita yang digunakan sebanyak 30 artikel yang diambil dari postingan tanggal 1 Januari 2019 sampai 31 Maret 2019. Dataset yang digunakan dalam penelitian ini berupa isi dari artikel berita. Situs berita online kompas.com digunakan dalam penelitian ini karena kompas.com merupakan salah satu situs berita online di Indonesia yang sering dikunjungi[12]. Sedangkan kategori tekno yaitu mengulas tentang kabar berita terkini dunia IT meliputi gadget terbaru, games, apps, smartphone, review produk, internet, software dan hardware.

Dalam penelitian ini melibatkan seorang pakar yang dapat membantu dalam menentukan hasil ringkasan secara manual. Dimana nantinya hasil ringkasan manual tersebut digunakan sebagai pembanding pada saat melakukan pengujian. Hasil ringkasan manual dari pakar berupa data kuisioner yang terdiri dari dokumen-dokumen kalimat berita, selanjutnya pakar melakukan ceklist pada dokumen kalimat yang dianggap mengandung informasi penting yang mampu mewakili isi berita secara keseluruhan. Seorang pakar yang terlibat dalam penelitian ini yaitu guru Bahasa Indonesia. Guru Bahasa Indonesia dipilih karena dalam penelitian ini menggunakan dokumen teks berbahasa Indonesia.

V. Pengujian

Metode pengujian ringkasan yang digunakan dalam penelitian ini yaitu intrinsik. Metode intrinsik merupakan metode evaluasi dimana kualitas ringkasan diukur berdasarkan hasil ringkasan yang dihasilkan. Dalam evaluasinya, dilakukan perbandingan antara hasil ringkasan sistem dengan hasil ringkasan manual yang dibuat oleh pakar.

Recall Oriented Understudy for Gisting Evaluation (ROUGE) adalah proses pengujian dengan menghitung jumlah *n-gram* yang sama antara ringkasan sistem dengan ringkasan manual. Dalam penelitian ini menggunakan ROUGE-1 karena memiliki tes signifikan *recall* yang tinggi. ROUGE-1 yaitu menghitung jumlah unigram kata yang sama antara ringkasan sistem dengan ringkasan manual. Parameter untuk evaluasi kenerja adalah *Recall*, *Precision* dan *F-Measure* dengan menggunakan persamaan 18, 19, dan 20[13].

$$\text{ROUGE-N } Recall = \frac{Common\ N-grams(Peer,Reference)}{N-grams(Reference)} \quad (18)$$

$$\text{ROUGE-N } Precision = \frac{Common\ N-grams(Peer,Reference)}{N-grams(Peer)} \quad (19)$$

$$\text{ROUGE-N } F\text{-}Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (20)$$

Pada persamaan (18), persamaan (19), dan persamaan (20) terdapat tiga komponen penting yaitu *Common N-grams(Peer,Reference)* adalah jumlah *n-gram* dalam ringkasan sistem dan ringkasan manual. *N-grams(reference)* adalah jumlah *n-gram* dalam ringkasan manual. *N-grams(peer)* adalah jumlah *n-gram* dalam ringkasan sistem.

Proses pengujian yang dilakukan dalam penelitian ini yaitu pengujian berdasarkan *learning rate*, pengujian berdasarkan *compression rate* dan melakukan perbandingan antara hasil ringkasan menggunakan RBM dan tanpa RBM.

A. *Pengujian berdasarkan Learning Rate*

TABEL I
HASIL PENGUJIAN BERDASARKAN LEARNING RATE

Learning Rate (ε)	Recall	Precision	F-Measure
0.01	0.728	0.604	0.657
0.03	0.737	0.610	0.664
0.06	0.744	0.611	0.669
0.09	0.736	0.603	0.660

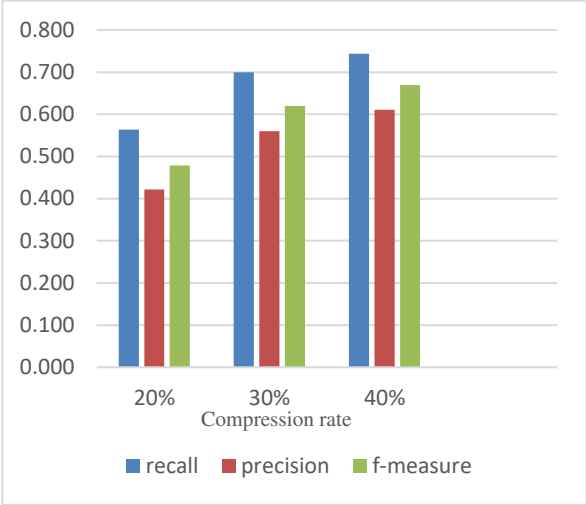
Terdapat empat *learning rate* yang digunakan dalam pengujian ini yaitu 0.01, 0.03, 0.06 dan 0.09. Dalam pengujian digunakan 30 dokumen artikel berita berbahasa Indonesia dengan *compression rate* 40%. Pembatasan epoch yang digunakan sebanyak 50 epoch. Pengujian ini dilakukan untuk mengetahui hasil *recall*, *precision* dan *f-measure* berdasarkan *learning rate* yang diuji. Dari hasil pengujian, tampak bahwa *learning rate* dengan 0.06 menghasilkan *recall*, *precision* dan *f-measure* tertinggi seperti yang terlihat pada tabel I.

B. *Pengujian berdasarkan Compression Rate*

Dalam pengujian ini, *compression rate* yang digunakan yaitu 20%, 30% dan 40% dari setiap data dokumen. Sebanyak 30 dokumen artikel berita berbahasa Indonesia yang digunakan. Pengujian ini dilakukan untuk mengetahui hasil *recall*, *precision* dan *f-measure* berdasarkan *compression rate* yang diuji. Pada gambar 3 tampak bahwa hasil *recall*, *precision* dan *f-measure* tertinggi pada saat pengujian dengan menggunakan *compression rate* 40%.

C. *Pengujian dengan Membandingkan antara Hasil Peringkasan menggunakan RBM dan tanpa RBM*

Pengujian ini dilakukan untuk mengetahui perbandingan antara hasil peringkasan dengan menggunakan RBM dan hasil peringkasan tanpa RBM. Dari hasil pengujian, tampak bahwa terdapat perbedaan hasil rata-rata *recall*, *precision* dan *f-measure* pada peringkasan menggunakan RBM dengan peringkasan tanpa RBM. Nilai rata-rata *recall*, *precision* dan *f-measure* pada peringkasan menggunakan RBM lebih tinggi daripada nilai rata-rata *recall*, *precision* dan *f-measure* pada peringkasan tanpa RBM seperti yang terlihat pada tabel II.



Gambar. 3. Grafik Hasil Pengujian berdasarkan *Compression Rate*

TABEL II
HASIL PENGUJIAN BERDASARKAN PERBANDINGAN ANTARA MENGGUNAKAN RBM DAN TANPA RBM

Artikel	Dengan RBM			Tanpa RBM		
	R	P	F	R	P	F
1	0.4953	0.3926	0.4380	0.4953	0.3926	0.4380
2	0.6879	0.6136	0.6486	0.6879	0.6136	0.6486
3	0.5773	0.4786	0.5234	0.5773	0.4786	0.5234
4	0.4588	0.4382	0.4483	0.4588	0.4382	0.4483
5	0.8107	0.6009	0.6902	0.6627	0.5068	0.5744
6	0.8065	0.7813	0.7937	0.8065	0.7813	0.7937
7	0.9130	0.6087	0.7304	0.9130	0.6087	0.7304
8	0.7822	0.6870	0.7315	0.7822	0.6870	0.7315
9	1.0000	0.7248	0.8404	1.0000	0.7248	0.8404
10	0.8049	0.5593	0.6600	0.8049	0.5593	0.6600
11	0.7216	0.5983	0.6542	0.7216	0.5983	0.6542
12	0.5833	0.3203	0.4135	0.5476	0.3108	0.3966
13	0.5446	0.4766	0.5083	0.5446	0.4766	0.5083
14	0.6835	0.5243	0.5934	0.6835	0.5243	0.5934
15	0.6458	0.4189	0.5082	0.6563	0.4286	0.5185
16	0.6364	0.5469	0.5882	0.6364	0.5469	0.5882
17	0.9070	0.5821	0.7091	0.9070	0.5821	0.7091
18	0.5960	0.5728	0.5842	0.5960	0.5728	0.5842
19	0.7092	0.5917	0.6452	0.7092	0.5917	0.6452
20	0.6463	0.5521	0.5955	0.5244	0.4624	0.4914
21	0.6981	0.5606	0.6218	0.6981	0.5606	0.6218
22	0.8169	0.7073	0.7582	0.8169	0.7073	0.7582
23	0.8721	0.6466	0.7426	0.8721	0.6466	0.7426
24	0.8542	0.7736	0.8119	0.7292	0.6863	0.7071
25	0.7015	0.4845	0.5732	0.6418	0.4526	0.5309
26	0.5765	0.5385	0.5568	0.5765	0.5385	0.5568
27	0.5301	0.3385	0.4131	0.5301	0.3385	0.4131
28	0.6742	0.6122	0.6417	0.5393	0.4528	0.4923
29	0.5783	0.4948	0.5333	0.5783	0.4948	0.5333
30	0.7000	0.5698	0.6282	0.6857	0.5714	0.6234
Rerata	**0.700**	**0.560**	**0.620**	**0.679**	**0.544**	**0.602**

VI. KESIMPULAN

Dalam melakukan peringkasan teks ekstraktif secara otomatis dengan menggunakan metode *Restricted Boltzmann Machine* (RBM), salah satu yang perlu diperhatikan yaitu penentuan *learning rate* yang akan digunakan agar ringkasan yang dihasilkan baik. Berdasarkan pengujian yang dilakukan dalam penelitian ini, dengan menggunakan *learning rate* 0.06 menghasilkan *recall, precision dan f-measure* tertinggi yakni 0.744, 0.611 dan 0.669.

Compression rate digunakan dalam peringkasan untuk menentukan panjang ringkasan yang dihasilkan. Berdasarkan pengujian yang dilakukan, semakin besar nilai *compression rate* yang digunakan maka hasil *recall*, *precision* dan *f-measure* yang dihasilkan akan semakin tinggi.

Pada peringkasan teks tanpa RBM diperoleh nilai rata-rata *recall* 0.679, *precision* 0.544 dan *f-measure* 0.602. Sedangkan pada peringkasan teks menggunakan metode RBM diperoleh nilai rata-rata *recall* 0.700, *precision* 0.560 dan *f-measure* 0.620. Peringkasan teks dengan menggunakan metode RBM menghasilkan nilai rata-rata *recall*, *precision* dan *f-measure* lebih tinggi daripada peringkasan teks tanpa menggunakan RBM, walaupun perbedaan hasil yang diperoleh tidak terlalu jauh. Peringkasan teks menggunakan metode RBM memiliki nilai *recall* lebih tinggi 2.1%, *precision* lebih tinggi 1.6% dan *f-measure* lebih tinggi 1.8% daripada peringkasan teks tanpa RBM. Hal ini menunjukkan bahwa peringkasan teks dengan menggunakan RBM hasilnya lebih baik daripada peringkasan teks tanpa RBM.

Pada metode RBM yang digunakan dalam penelitian ini masih perlu dilakukan improvisasi dalam penentuan nilai bobot dan bias yang akan digunakan agar ringkasan yang dihasilkan lebih baik lagi. Untuk penelitian selanjutnya dapat menambahkan fitur selain yang ada pada penelitian ini serta metode ini dapat dikombinasikan dengan metode lain untuk meningkatkan hasil peringkasan.

DAFTAR PUSTAKA

[1] J. S. Saputra, M. Fachrurrozi, Yunita, "Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode Latent Semantic Analysis (LSA) dan Teknik Steinberger & Jezek," in *Prosiding Annual Research Seminar Computer Science and ICT*, 2017.

[2] S. Irawan, Hermawan, Samsuryadi, "Studi Awal Peringkasan Dokumen Bahasa Indonesia Menggunakan Metode Latent Semantik Analysis dan Maximum Marginal Relevance," in *Prosiding Annual Research Seminar*, 6 Desember 2016.

[3] I P. G. H. Suputra. (2017, April). Peringkasan Teks Otomatis Untuk Dokumen Bahasa Bali Berbasis Metode Ektraktif. *Jurnal Ilmu Komputer*. X(1), pp. 33-38. Available: https://ojs.unud.ac.id/index.php/jik/article/view/39775/24171

[4] N. S. W. Gotami, Indriati, R. K. Dewi. (2018, September). Peringkasan Teks Otomatis Secara Ekstraktif Pada Artikel Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Latent Semantic Analysis. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 2(9), pp. 2821-2828. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2430/905

[5] A. Ambekar, K. Shah, M. Agrawal, S. Pawar, A. Shaikh. (2018, June). Text Summarization Using Restricted Boltzmann Machine: Unsupervised Deep Learning Approach. *IJSART*. 4(6), pp. 103-107. Available: http://ijsart.com/Content/PDFDocuments/IJSARTV4I623858.pdf

[6] M. Elgamal, Prof. Dr S. Hamada, Prof. Dr R. Aboelezz and Dr M. Abou-Kreisha. (2019, August). Better Results in Automatic Arabic Text Summarization System Using Deep Learning based RBM than by Using Clustering Algorithm based LSA. *International Journal of Scientific & Engineering Research*. 10(8), pp. 781-786. Available: https://www.ijser.org/researchpaper/Better-Results-in-Automatic-Arabic-Text-Summarization-System-Using-Deep-Learning-based-RBM-than-by-Using-Clustering-Algorithm-based-LSA.pdf

[7] S. P. Singh, A. Kumar, A. Mangal, S. Singhal. (2016). Bilingual Automatic Text Summarization Using Unsupervised Deep learning. *IEEE International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. pp. 1195-1200. Available: https://ieeexplore.ieee.org/document/7754874

[8] N. S. Shirwandkar, Dr. S. Kulkarni. (2018). Extractive Text Summarization using Deep Learning. *IEEE Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. Available: https://ieeexplore.ieee.org/document/8697465

[9] N. I. Widiastuti, W. K. Afnan. (2017). Fuzzy Logic dan Lexical Chains untuk Peringkasan Teks Otomatis. *Jurnal Sistem* Komputer. 7(1), pp. 5-12. Available: https://docplayer.info/91149501-Fuzzy-logic-dan-lexical-chains-untuk-peringkasan-teks-otomatis.html

[10] A. Ridok, T. C. Romadhona, "Peringkasan Dokumen Otomatis Menggunakan Metode Fuzzy Model Sistem Inferensi Mamdani," in *Seminar Nasional Teknologi Informasi dan Multimedia*, Yogyakarta, 19 Januari 2013, pp. 19-24.

[11] Susilawati, "Algoritma Restricted Boltzmann Machines (RBM) untuk Pengenalan Tulisan Tangan Angka," in *Seminar Nasional Teknologi Informatika, "The Future of Computer Vision"*, 2017, pp. 140-148.

[12] D. Branding (2019) "Ini Dia 7 Situs Berita Online di Indonesia yang Sering di Kunjungi," [Online]. https://www.nataconnexindo.com/blog/ini-dia-7-situs-berita-online-di-indonesia-yang-sering-di-kunjungi, tanggal akses: 21-Mei-2020.

[13] J. Yadav, Dr. Y. K. Meena. (2016, Sept). Use of Fuzzy Logic and WordNet for Improving Performance of Extractive Automatic Text Sumarization. *IEEE Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI)*. pp. 2071-2077. Available: https://ieeexplore.ieee.org/document/7732356

Rully Widiastutik lahir di Sumenep, Jawa Timur, Indonesia, pada tahun 1988. Menyelesaikan studi S1 di program studi Teknik Informatika Universitas Trunojoyo Madura pada tahun 2010, dan berkarir sebagai dosen di program studi Teknik Informatika Akademi Komunitas Negeri Sumenep. Minat penelitiannya adalah bidang text mining.

Lukman Zaman P. C. S. W berkarir sebagai dosen di program studi Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya.

Joan Santoso lahir di Surabaya, Jawa Timur, Indonesia. Telah menyelesaikan pendidikan S1 pada tahun 2011 dan S2 pada tahun 2013 dari Sekolah Tinggi Teknik Surabaya. Minat penelitiannya ialah computational linguistic, information extraction, machine learning, dan big data processing.