Analisis Metoda *Latent Dirichlet Allocation* untuk Klasifikasi Dokumen Laporan Tugas Akhir Berdasarkan Pemodelan Topik

Urip T. Setijohatmo[1], Setiadi Rachmat[2], Tati Susilawati[3], Yuda Rahman[4]

[1,2,3,4]*Jurusan Teknik Komputer, Politeknik Negeri Bandung*
Jl. Gegerkalong Hilir, Ds. Ciwaruga, Bandung 40012

ABSTRAK

Mahasiswa tingkat akhir Jurusan Teknik Komputer setiap tahunnya mengerjakan tugas akhir. Tugas akhir tersebut merupakan salah satu syarat kelulusan. Untuk mengerjakan tugas akhir dibutuhkan referensi-referensi, salah satunya adalah dokumen tugas akhir tahun-tahun sebelumnya. Untuk mencari dokumen tugas akhir tersebut Jurusan Teknik Komputer hanya memperlihatkan katalog yang berisi judul-judul tugas akhir. Permasalahannya adalah tidak semua judul yang diberikan menggambarkan isi dari dokumen tersebut. Salah satu cara dalam mengatasi masalah tersebut adalah dengan pemodelan topik. Penelitian ini akan menggunakan Perluasan PLSA dari pendekatan lain yang disebut LDA (*Latent Dirichlet Allocation*), spesifiknya menggunakan algoritma *Gibbs Sampling*, dan dilakukan pada studi kasus pencarian dokumen laporan tugas akhir. Eksperimen menggunakan sekumpulan laporan tugas akhir yang telah diberi label. Selanjutnya hasil eksperimen akan diukur tingkat kerelevanannya jika dibandingkan dengan *judgement* manusia dalam bentuk laporan tugas akhir berlabel

Kata Kunci
Topic Modeling, Latent Dirichlet Allocation, Gibbs Sampling

1. PENDAHULUAN

Pada jaman sekarang informasi yang sudah terdigitalisasi semakin bertambah dan menyebabkan kesulitan dalam mencari informasi yang sesuai dengan kebutuhan, sehingga dibutuhkan teknologi yang bisa membantu untuk mencari, memahami, dan mengorganisasi informasi tersebut. Mahasiswa tingkat akhir Jurusan Teknik Komputer setiap tahunnya mengerjakan tugas akhir. Tugas akhir tersebut merupakan salah satu syarat kelulusan. Untuk mengerjakan tugas akhir dibutuhkan referensi-referensi, salah satunya adalah dokumen tugas akhir tahun-tahun sebelumnya.

1.1. Latar Belakang

Untuk mencari dokumen tugas akhir tersebut Jurusan Teknik Komputer hanya memperlihatkan katalog yang berisi judul-judul tugas akhir. Permasalahannya adalah tidak semua judul yang diberikan menggambarkan isi dari dokumen tersebut. Salah satu cara dalam mengatasi masalah tersebut adalah dengan pemodelan topik. Pemodelan topik bertujuan untuk menemukan topik dari kumpulan dokumen. Ada beberapa metoda untuk pemodelan topik yaitu LSA, PLSA, LDA. Pada penelitian sebelumnya [2] telah dilakukan penelitian pemodelan topik dengan metode RPLSA (*Robust Probabilistic Latent Semantic Analysis*) yang merupakan perluasan PLSA (*Probabilistic Latent Semantic Analysis*) dengan penanganan *overfitting*. Untuk menangani *overfitting* selain perluasan PLSA terdapat pendekatan lain yang disebut LDA (*Latent Dirichlet Allocation*) [1]. Namun pada pendekatan LDA tidak bisa secara langsung diimplementasikan karena ada kesulitan dalam perhitungannya sehingga mengestimasi parameter untuk variabel tersembunyi tersebut salah satunya digunakan *Gibbs Sampling*.

1.2. Rumusan Masalah

Berdasarkan latar belakang, permasalahan yang muncul ialah apakah metoda *Latent Dirichlet Analysis* dapat digunakan untuk memunculkan topik yang tersembunyi pada dokumen laporan tugas akhir dengan mengimplementasi *Gibbs Sampling* dan mengetahui seberapa baik kerelevanan klasifikasi topik yang dihasilkan.

1.3. Batasan Masalah

Beberapa batasan masalah yaitu:

- Menggunakan dokumen tugas akhir yang berbahasa Indonesia
- Menggunakan dokumen tugas akhir yang sudah berbentuk pdf

- Sumber dokumen laporan tugas akhir berasal dari Jurusan Teknik Komputer

1.4. Research Question

Seberapa besar tingkat kerelevanan metode *Latent Dirichlet Allocation* untuk pencarian dokumen laporan tugas akhir berdasarkan pemodelan topik jika dibandingkan dengan judgement manusia

1.5. Hipotesis

Dengan menggunakan metode *Latent Dirichlet Allocation* dapat dilakukan pencarian dokumen laporan tugas akhir berdasarkan topik dengan tingkat relevansi 75% dari *expert judgement* manusia.

2. STUDI PUSTAKA

2.1. Penelitian Sejenis

Penelitian sejenis yang dibahas di sini adalah tentang kategorisasi teks untuk dua hal yaitu kinerja *speech recognition* dan *information retrieval*.

2.1.1 Kategorisasi untuk Speech Recognition

Penelitian [3] bertujuan membentuk subkorpora teks dari corpora teks Slovakia yang berisi dokumen teks yang mirip dimana mereka menggunakan teks subkorpora yang terorganisasi lebih baik tsb untuk sistem *speech recognition*. Bahwa hasil untuk *speech recognition* akan lebih baik jika teks sudah dikategorisasi. Karena itulah Penelitian [3] fokus pada kategorisasi teks, yang dicobakan menggunakan LDA. Initial corpus dibagi menjadi 2, 5, 10, 20, 100 subcorpora. Model bahasa dibangun pada subcorpora tsb dan diadaptasi dengan *linear interpolation* pada domain hukum. Hasil eksperimen memperlihatkan bahwa kategorisasi text menggunakan LDA memperbaiki *automatic speech recognition*.

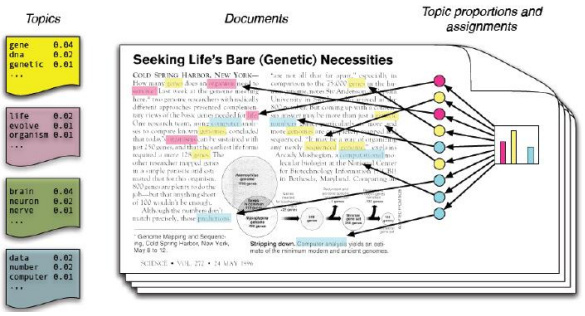
2.1.2 LDA untuk Perbaikan Kategorisasi BOW

Hampir semua algoritma kategorisasi teks merepresentasikan suatu koleksi dokumen sebagai suatu *Bag of Words* (BOW). Representasi BOW ini tidak mampu mengenali sinonim dari sekumpulan term dan tidak mampu mengenali keterhubungan semantik diantara terms. Penelitian [4] menerapkan pemodelan topik dengan berdasarkan pendapat bahwa term pada topik yang sama adalah secara semantik berkaitan sehingga berpotensi lebih baik dibandingkan dengan hanya berdasarkan BOW. Hasil eksperimen menunjukkan pendekatan pemodelan topik untuk mewakili dokumen menghasilkan klasifikasi yang lebih baik. Penelitian [4] menggunakan metode LDA dalam pemodelan topik.

2.2. Konsep

2.2.1 Metode *Latent Dirichlet Allocation*

Latent Dirichlet Allocation adalah sebuah *generative probabilistic model* untuk menemukan *latent semantic topic* di dalam kumpulan data text. Intuisi dasar dari LDA adalah sebuah dokumen memuat berbagai topik [1]. LDA dapat mengatasi permasalahan *overfitting* [1] yang dialami metode PLSA. *Overfitting* menggambarkan suatu kondisi di mana model memiliki terlalu banyak parameter yang mengarahkan tingkat kecocokan yang tinggi untuk sampel tersebut, namun dengan sampel baru tingkat kecocokan tersebut menjadi rendah.



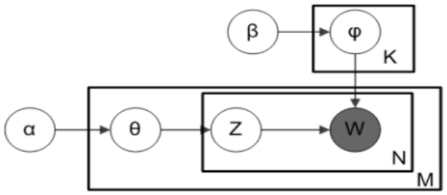
Gambar 1. Ilustrasi LDA

Cara kerja dari model LDA yaitu pertama-tama mengasumsikan topik telah dispesifikasikan sebelum didapatkan dokumen pada gambar tersebut adalah deretan topik yang ada di sebelah kiri.

Untuk setiap dokumen di dalam koleksi dilakukan:

1. Secara acak dipilih distribusi atas topik (pada gambar ditunjukan sebagai grafik distribusi topik di sebelah kanan)
2. Untuk setiap kata didalam dokumen:
   a. Secara acak dipilih sebuah topik dari distribusi atas topik, pada langkah 1 (pada gambar ditunjukan pada hubungan grafik dengan lingkaran).
   b. Secara acak dipilih distribusi sebuah kata dari distribusi yang sesuai ,atas kosakata. (pada gambar ditunjukan dengan cara memilih warna pada lingkaran)

LDA digambarkan dengan model grafis menggunakan *plate notation* seperti pada Gambar-2.



Gambar 2. *Plate Notation* LDA

dimana

- β adalah *dirichlet parameter* atas distribusi kata terhadap topik
- φ adalah distribusi kata terhadap topik dalam *corpus*
- K adalah kumpulan topik
- W adalah kata
- N adalah kumpulan kata

- M adalah kumpulan dokumen
- Z adalah topik *index assignment*
- θ adalah dokumen, dan
- α adalah *dirichlet parameter* atas distribusi topik terhadap dokumen. LDA dirumuskan sebagai berikut:

$$p(w,z,\vartheta,\varphi|\alpha,\beta) = p(\varphi|\beta)p(\vartheta|\alpha)p(z|\vartheta)p(w|\varphi_k)$$

Pada Implementasinya LDA tidak bisa diterapkan karena variabel Z, θ, dan φ yang tersembunyi/ tidak diketahui dan juga untuk mencari asosiasi antara z dan w sulit karena satu kata bisa mengandung 2 topik atau lebih, dan jika kita mencoba untuk mengestimasi $p(Z|W)$ yang menghasilkan :

$$p(\vec{z}|\vec{w}) = \frac{p(\vec{z},\vec{w})}{p(\vec{w})} = \frac{\prod_{i=1}^{W} p(zi,wi)}{\prod_{i=1}^{W}\sum_{k=1}^{K} p(zi=k,wi)}$$

Pada bagian pembagi *intractable* atau tidak bisa dikelola karena harus mengkomputasikan penjumlahan atas KW term. Untuk mengatasi masalah tersebut salah satunya dengan menggunakan metode *Collapsed Gibbs Sampling* [5].

2.2.2. *Collapsed Gibbs Sampling*

Collapsed Gibbs Sampling adalah salah satu algoritma yang berasal dari *Markov Chain Monte Carlo* (MCMC). Gibbs sampling merupakan cara yang terbaik untuk mengestimasi p(z|w) [6]. Berikut pada Gambar 4 adalah algoritma *collapsed gibbs sampling.* Cara kerjanya adalah pertama tama dengan insialisasi z dengan merandom topik untuk setiap kata , lalu mensampel z dari semua kata atas distribusi kata dengan topik dari seluruh dokumen dan topik dengan dokumen lalu mengassign topik hasil sample untuk setiap kata. :

```
Data: words w ∈ documents d
Result: Topic assignments z
Initialize z randomly
foreach iteration do
 for each w do
   for each topik k do
     θ_dw,k = (n^w_-w,k + β) / (n_-w,k + Wβ) * (n^dw_-w,k + α) / (n^dw_-w + Kα)
   end
  topik ←sample from mult(θ_dw)
  Z[w] ← topik
  Update counts according to new assignment
 end
end
Return z
```

dimana:

- $\theta_{dw,k}$: p(w|z) atau probabilitas kata terhadap topik
- $n^{w}_{-w,k}$: berapa kali kata w di assign ke topik k di setiap dokumen
- β : *dirichlet parameter* atas distribusi kata terhadap topik di corpus (semua dokumen)
- $n^{d_w}_{-w,k}$: berapa kali topik k di assign ke dokumen d
- α adalah *dirichlet parameter* atas distribusi topik terhadap dokumen
- $n_{-w,k}$: berapa kali kata selain w di *assign* ke topik k di setiap dokumen
- W: jumlah variasi kata di dalam corpus
- $n^{d_w}_{-w}$: berapa kali topik selain k di *assign* ke d.

2.2.3 *Text Preprocessing*

Proses ini digunakan untuk mempersiapkan masukan dalam rangka komputasi dari dokumen dengan melakukan *case folding*, *tokenizing*, *filtering term*, *stopword removal*, dan *stemming*

2.2.4. *Matrix Inverted Index*

Matrix Inverted Index digunakan untuk menampung hasil dari *preprocessing*. *Inverted index* berisikan pemetaan dari kata(*term*) ke dokumen dimana kata tersebut muncul, dengan struktur baris adalah kata dan kolom adalah dokumen, dan sel berisi TF dan TF-IDF.

2.2.5. TF-IDF

TF-IDF ini digunakan untuk mengetahui frekuensi kata didalam dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut. Untuk menghitung idf digunakan perhitungan sebagai berikut:

$$W_{ij} = tf_{ij} \times ((\log\frac{N}{n}) + 1) \quad \text{dimana:}$$

- W_{ij} : bobot kata term t_j terhadap dokumen d_i
- tf_{ij} : jumlah kemunculan kata / term t_j dalam d_i
- N : jumlah semua dokumen yang ada
- n : jumlah dokumen mengandung kata/term t_j

Perhitungan dilakukan pada matriks *inverted index*, dan hasilnya digunakan untuk perhitungan LDA.

2.2.6. *Precision Measurement*

Precision Measurement adalah salah satu cara untuk mengukur efektifitas *information retrieval*. Yaitu dengan membandingkan *relevan item* yang diterima dengan item yang diterima.

$$\text{Precision} = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)}$$

$$= P(relevant|retrieved)$$

3. METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan yaitu penelitian eksperimental dengan pendekatan kuantitatif, sebab pada penelitian ini akan fokus mengukur kerelevanan topik yang dihasilkan pada implementasi pencarian dokumen tugas akhir berdasarkan abstrak dengan menggunakan LDA. Hasil pengukuran tersebut lalu

dibandingkan dengan judgement manusia untuk mengetahui kerelevanan topik yang didapat

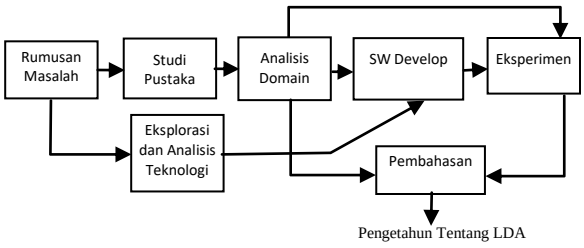
3.2 Data Penelitian

Data penelitian berupa data kata didalam dokumen, dimana dokumen adalah laporan tugas akhir pada bagian tertentu sesuai hasil analisis. Data penelitian ini akan didapatkan melalui mekanisme *text preprocessing* pada kumpulan dokumen laporan tugas akhir yang berupa pdf.

Bagian dari dokumen laporan tugas akhir hanya pada bagian cover memiliki informasi mengenai judul dan abstraksi karena memuat intisari dokumen yang dapat mencerminkan isi dari dokumen laporan tugas akhir.

3.3 Langkah Proses Penelitian

Berikut adalah langkah-langkah proses pelaksanaan penelitian yang terdiri dari Rumusan Masalah; Studi Pustaka; Analisis Domain Permasalahan; Pengembangan Perangkat Lunak; Eksperiment; Pembahasan. Diagram di bawah ini merepresentasikan langkah proses pelaksanaan penelitian berikut pengaruhnya suatu langkah terhadap langkah proses lain.



Gambar 3. Rancangan Langkah Proses Penelitian

4. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil analisis terhadap penelitian dimulai dari analisis domain problem, Pengembangan Perangkat Lunak, Eksperimen serta Pembahasan Hasil Eksperimen

4.1 Analisis Domain Masalah

Analisis problem domain bertujuan untuk mengetahui domain permasalahan yang ada. Permasalahan untuk penelitian ini berkisar pada dokumen tugas akhir yang terdiri dari deskripsi dokumen tugas akhir, mengapa menggunakan pemodelan 405opic LDA, dan analisis penggunaan LDA dengan menggunakan *collapsed gibbs sampling*.

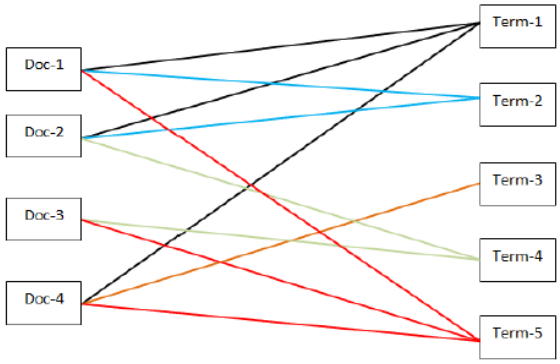
4.1.1 Dokumen Tugas Akhir

Dokumen laporan tugas akhir terkomposisi dari beberapa bab seperti kata pengantar, pendahuluan, studi pustaka, analisis, metodologi penelitian, hasil dan pembahasan, serta kesimpulan dan saran. Isi dari dokumen terdiri dari kalimat bahasa Indonesia mengandung pengertian/semantic umum dan khusus. Pemilihan data bagian-bagian dokumen mana saja yang mempengaruhi presisi hasil klasifikasi. Dari bagian-bagian yang merupakan komponen isi dokumen, bagian yang paling lengkap dan padat berpengertian khusus adalah abstraksi.

4.1.2 Pemodelan Topik

Berdasarkan uraian sebelumnya permasalahan pencarian dokumen berdasarkan topik adalah cara yang menantang apakah lebih baik dibandingkan dengan berdasarkan judul atau keyword. Pendekatan ini menggunakan pemodelan topik yang bertujuan untuk memunculkan topik tersembunyi dari kumpulan dokumen.

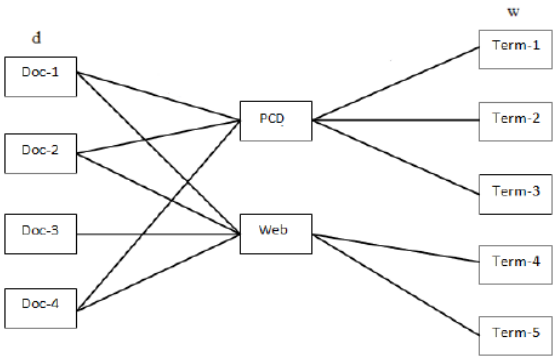
Pada gambar 4 terdapat himpunan dokumen dan himpunan kata/term. Secara intrinsik kata-kata mewakili topik, dimana pada kenyataannya terdapat



Gambar 4. Topik Tersembunyi dari Dokumen

tema/inti/topik dari dokumen-dokumen tersebut. Adalah mungkin suatu dokumen mengandung lebih dari satu topik dengan kekentalan masing-masing.

Pada kasus nyata manusia dapat mengetahui topik dari dokumen tersebut karena mempunyai pengetahuan dari kata-kata yang muncul. Namun tidak demikian jika dilakukan komputasi oleh mesin komputer, dimana hanya dimiliki dokumen dengan himpunan kata-kata saja.



Gambar 5. Memunculkan Topik dari Dokumen

Sehingga digunakanlah pemodelan topik untuk memunculkan topik tersembunyi dari dokumen-dokumen tersebut. Pada Gambar 5 manusia mengenali konsep sering diwakili oleh kata atau terdapat kata-kata spesifik mewakili konsep, sehingga diantara

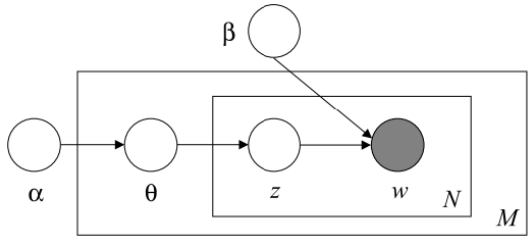
dokumen terdapat topik yang menghubungkan Dokumen dengan kata.
Dokumen-1 mempunyai topik Web dengan kata-kata yang mewakili topik ini. Topik-topik ini adalah bernama dan hanya manusia yang dapat melakukannya karena manusia mempunyai pengetahuan, atau dengan perkataan lain topik-topik yang dihasilkan mesin tidak akan bernama (berlabel). Masalah pelabelan adalah diluar lingkup dari penelitian ini.

4.1.3 Analisis Penggunaan LDA

Bagian ini ini adalah untuk mengetahui penggunaan LDA dan bagaimana cara untuk mengimplementasikan LDA dan simulasi penggunaan *collapsed gibbs sampling*.

4.1.3.1. LDA

Pada [1] diketahui bahwa *Latent Dirichlet Allocation* (LDA) dapat mengatasi masalah *overfitting*. LDA merupakan pengembangan dari PLSA namun pada LDA terdapat parameter *dirichlet*, direpresentasikan dengan gambar berikut:



Gambar 6. Memunculkan Topik dari Dokumen

Pada gambar tersebut terlihat representasi yang serupa dengan PLSA namun LDA terdapat 2 parameter dirichlet yaitu α sebagai parameter dirichlet dari distribusi dokumen terhadap topik dan β sebagai parameter dirichlet dari distribusi topik terhadap kata. Pada perhitungan LDA digunakkan algoritma Collapsed Gibbs Sampling untuk mengetahui probabilitas topik terhadap dokumen dan topik terhadap kata. Perhitungan ini juga akan diulang terus menerus hingga selisih iterasi yang dilakukan sudah konvergen atau mendekati 0. Masalah *overfitting* ditangani LDA dengan menambahkan parameter *dirichlet* saat melakukan perhitungan, yang digunakan sebagai faktor normalisasi.

4.1.3.2. Collapsed Gibbs Sampling

Teknik ini digunakan mengestimasi probabilitas topik terhadap kata dan dokumen terhadap topik. Cara Kerja dari Collapsed Gibbs Sampling :

1. Inisialisasi kata terhadap dokumen dengan merandom topik untuk setiap kata
2. Mensampel topik dari semua kata atas distribusi kata dengan melihat distribusi topik dari seluruh dokumen dan topik dengan dokumen
3. Lalu mengassign topik yang dari hasil sample untuk setiap kata tadi lebih besar.

Berikut adalah simulasi untuk 3 dokumen (D1, D2, dan D3) dan 2 buah topik tersembunyi (T1, dan T2).

Pertama menginisialisasi topik untuk semua kata di dalam dokumen secara *random* (1 adalah T1, dst).

Tabel 1. Inisialisasi Topik secara Random

D1	Money	Bank	Loan	Bank	Money	Money	Bank	Loan
	1	2	1	2	1	2	1	2
D2	Money	Bank	Bank	River	Loan	Stream	Bank	Money
	1	2	1	2	1	2	1	2
D3	River	Bank	Stream	Bank	River	River	Stream	Bank
	1	2	1	2	1	2	1	2

Dari Tabel 1 di atas dapat diketahui distribusi topik terhadap kata pada semua dokumen dan distribusi dokumen terhadap topik, diperlihatkan pada Tabel 2:

Tabel 2. Distribusi Topik

	T1	T2
Money	3	2
Bank	3	6
Loan	2	1
River	2	2
Stream	2	1

	D1	D2	D3
T1	4	4	4
T2	4	4	4

Pada tahap selanjutnya dilakukan sample dengan cara pertama melakukan pengurangan terlebih dahulu jumlah distribusi kata terhadap topik dan dokumen terhadap topik untuk topik yang terpilih, lalu sampling dengan menggunakan rumus

$$P(z_i = j \mid z_{-i}, w_i, d_i, .) \propto \frac{C_{wij}^{WT} + \beta}{\sum_{w=1}^{W} C_{wij}^{WT} + W\beta} \frac{C_{dij}^{DT} + \alpha}{\sum_{t=1}^{T} C_{dit}^{DT} + T\alpha}$$

C_{wij}^{WT} berapa kali w di-*assign* ke topik k di setiap dokumen
C_{dij}^{DT} berapa kali topik k di-*assign* ke dokumen d

C_{wj}^{WT} berapa kali w di-*assign* ke topik selain k di setiap dokumen
C_{dj}^{DT} berapa kali topik k di-*assign* ke selain d

α adalah *dirichlet* parameter atas distribusi dokumen terhadap topik, diambil dari 50/T, nilai tersebut adalah nilai standar untuk distribusi *dirichlet*
β adalah *dirichlet* parameter atas distribusi topik terhadap kata di semua dokumen yaitu 0.001, nilai tersebut adalah nilai standar untuk distribusi *dirichlet*
W adalah jumlah kata untuk setiap dokumen
T adalah jumlah topik untuk setiap dokumen

Pada simulasi ini akan mensample kata pertama pada dokumen 1 yaitu "Money". Kata tersebut diinisial-isasi sebagai topik 1.

$$P(z_i = T1 \mid z_{-i}, \text{Money}, d_i, .) = \frac{2+0.01}{9+5*0.01} * \frac{3+25}{4+2*25} = 0.10$$

$$P(z_i = T2 \mid z_{-i}, \text{Money}, d_i, .) = \frac{2+0.01}{10+5*0.01} * \frac{4+25}{3+2*25} = 0.11$$

Dapat dilihat bahwa topik 2 adalah hasil yang lebih besar, sehingga topik yang dipilih menjadi topik 2.

Tabel 3. Hasil *Gibbs Sampling* Kata Pertama Dokumen D1

D1	Money	Bank	Loan	Bank	Money	Money	Bank	Loan
	2	2	1	2	1	2	1	2
D2	Money	Bank	Bank	River	Loan	Stream	Bank	Money
	1	2	1	2	1	2	1	2
D3	River	Bank	Stream	Bank	River	River	Stream	Bank
	1	2	1	2	1	2	1	2

Jumlah distribusi topik terhadap kata dan dokumen terhadap topik disesuaikan untuk topik yang dipilih.

Tabel 4. Hasil Penyesuaian Jumlah Distribusi Topik

	T1	T2
Money	**2**	**3**
Bank	3	6
Loan	2	1
River	2	2
Stream	2	1

	D1	D2	D3
T1	3	4	4
T2	5	4	4

Sampling tersebut dilakukan berulang-ulang hingga selisih hasil sebelum dan sesudah sampling konvergen atau mendekati 0.

4.2 Eksperimen dan Pembahasan

Pada tahap ini akan dijelaskan skenario yang digunakan untuk melakukan eksperimen dan hasil dari eksperimen.

4.2.1 Skenario

Pada eksperimen ini akan dilakukan pengamatan terhadap hasil LDA yang menggunakan label(*judgement manusia*) dan hasil LDA yang tanpa menggunakan label. Untuk mengukur seberapa baik hasil dari metoda Latent Dirichlet Allocation digunakan juga beberapa skenario-skenario dari faktor-faktor yang mempengaruhi hasil metoda tersebut. Faktor faktor tersebut adalah jumlah dokumen atau jumlah kata dan jumlah topik. Berikut skenario yang akan dilakukan eksperimen :

4.2.2 Hasil

Hasil eksperimen ini terbagi tiga sesuai dengan jumlah skenario yang digunakan pada percobaan.

Tabel 5. Skenario Eksperimen

Skenario	Jumlah Dokumen	Jumlah Topik
S1	4	2
S2	60	3
S3	80	4

Hasil skenario 1

Hasil scenario 1 adalah bahwa Dok-1 adalah bertopik T1 karena probabilitasnya 75.17% lebih besar dari T2 yang 24.83%. Kesimpulan tersebut ditandai dengan warna kuning.

Tabel 6. Hasil Skenario S1

Dok	Topik				Presisi
	T1		T2		
	Frek	Prob	Frek	Prob	
D1	112	75.17%	37	24.83%	92.55%
D2	26	12.26%	186	87.74%	97.16%
D3	112	69.57%	49	30.43%	92.55%
D4	27	14.75%	156	85.25%	97.16%

Presisi T1 (dokumen D1 dan D3)= 92.55%, dan
Presisi T2 (dokumen D2 dan D4)= 97.16%,
Nilai presisi akhir merupakan rata2 presisi semua jenis dokumen bertopik berbeda, sehingga presisi LDA =(92.55%+97.16%)/2=94.86%

Cara yang sama untuk skenario S2 dan S3, hasilnya :

Tabel 6. Hasil Skenario S2 dan S3

Skenario	Tingkat Relevansi
S1	94.86 %
S2	91.80 %
S3	54.38 %
Rata-rata	80.35%

Dapat disimpulkan tingkat relevansi sesuai dengan hipotesa lebih besar dari 75%.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksperimen pada ketiga skenario, dapat disimpulkan bahwa:

- Hasil Probabilitas sebuah kata dipengaruhi oleh banyaknya jumlah topik dan jumlah dokumen.
- Metode LDA sensitive terhadap komposisi kata dimana ketika data yang digunakan mengandung banyak kata umum akan secara signifikan mengurangi tingkat presisi
- LDA dapat mengelompokan dokumen dengan topik tertentu namun tidak berlabel.

5.2 Saran

Saran pengembangan terhadap penelitian adalah

- perlu dilakukan kajian terhadap dirichlet parameter alpha dan beta terhadap hasil
- perlu dilakukan eksperimen terkait pengaruh jumlah dokumen dan atau jumlah topik terhadap hasil

DAFTAR PUSTAKA

[1] Blei, D. Probabilistic Topic Models, Communi-cations of the ACM, 2012, Vol 55, No.4.

[2] Rahmadita, O., Penerapan Metoda Robust Probabilistic Latent Semantic Analysis untuk Pencarian Dokumen Laporan Tugas akhir Berdasarkan Pemodelan Topik, 2014.

[3] Zlacky Daniel, Stas Ján, Juhar Jozef, Cizmar Anton, Text Categorization with Latent Dirichlet

Allocation, Journal of Electrical and Electronics Engineering 7(1):161-164, May 2014

[4] Wongkot Sriurai, Improving Text Categorization by Using a Topic Model, Advanced Computing: An International Journal (ACIJ), Vol.2, No.6, November 2011.

[5] William M. Darling. A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling. Technical Report, School of Computer Science University of Guelph, December, 2011.

[6] Wang, Y. Distributed Gibbs Sampling of Latent Topic Models: The Gritty Details, Course Notes, Agustus 2008.