BAB 2
LANDASAN TEORI

2.1. Peringkas Teks Otomatis

2.1.1. Definisi Peringkas Teks Otomatis

Dalam Kamus Besar Bahasa Indonesia Departemen Pendidikan Nasional Balai Pustaka (2008) mendefinisikan ringkasan, yaitu hasil meringkaskan, ikhtisar, singkatan cerita, dan kependekan. Sedangkan definisi otomatis dalam Kamus Besar Bahasa Indonesia Departemen Pendidikan Nasional Balai Pustaka (2008) yaitu dengan sendirinya, secara otomat. Ringkasan merupakan ekspresi yang ketat dari isi utama suatu artikel, tujuannya untuk memberitahu pembaca inti dari suatu pikiran utama[22]. Ringkasan adalah mengambil isi yang paling penting dari sumber informasi yang kemudian menyajikannya kembali dalam bentuk yang lebih ringkas bagi penggunanya.[23]

peringkas teks otomatis adalah teknik pembuatan ringkasan dari sebuah teks secara otomatis dengan memanfaatkan aplikasi yang dijalankan pada komputer untuk menghasilkan informasi yang paling penting dari dokumen aslinya[24].

2.1.2. Pendekatan Peringkasan Teks

Menurut Zaman B. dan E Winarko terdapat 2 pendekatan peringkas teks[24], yaitu:

1. Ekstraksi (*extractive summary*)

   Pada teknik ekstraksi, sistem menyalin unit-unit teks yang dianggap paling penting dari teks sumber menjadi ringkasan. Unit-unit teks yang disalin dapat berupa klausa utama, kalimat utama, atau paragraf utama tanpa ada penambahkan kalimat-kalimat baru yang tidak terdapat pada dokumen aslinya.

2. Abstraksi (*abstractive summary)*

   Teknik abstraksi menggunakan metode linguistik untuk memeriksa dan menafsirkan teks dokumen menjadi ringkasan. Ringkasan teks tersebut

dihasilkan dengan cara menambahkan kalimat-kalimat baru yang merepresentasikan intisari teks sumber ke dalam bentuk yang berbeda dengan kalimat-kalimat yang ada pada teks sumber.

Pada penelitian ini metode pendekatan yang digunakan untuk peringkasan teks otomatis yaitu teknik ekstraktif dengan memanfaatkan aplikasi yang dijalankan pada komputer untuk menghasilkan sebuah ringkasan dari dokumen aslinya.

2.1.3. Tahapan Membuat Ringkasan

Menurut Juhara, E,dkk ada beberapa tahapan untuk membuat ringkasan[25], yaitu sebagai berikut:

1. Membaca naskah asli secara menyeluruh untuk mengetahui kesan umum, gagasan pengarang, dan sudut pandangnya.
2. Mencatat semua gagasan utama atau gagasan penting
3. Menyusun kembali suatu karangan singkat berdasarkan gagasan tersebut.

Untuk membuat dan mendapatkan ringkasan yang baik, yaitu dengan cara membaca dengan seksama teks bacaan dan mencatat hal-hal pokok dari bacaan tersebut[25]. Ciri-ciri sebuah ringkasan yang baik itu harus mudah dipahami, bentuknya lebih singkat atau lebih ringkas, dan terdiri dari gagasan-gagasan utama (*expert judgement*).

2.1.4. Tujuan Ringkasan

Menurut Juhara, E,dkk tujuan dari peringkasan teks dapat dikategorikan berdasarkan maksud, fokus, dan cakupannya [25], yaitu sebagai berikut:

a. *Informatif* Tujuannya ringkasan ini adalah untuk menyatakan informasi–informasi penting yang terdapat pada dokumen asal.
b. *Indicatif* Tujuannya ringkasan ini adalah untuk dijadikan referensi yang membantu pembaca mengetahui isi dari teks daripada membaca keseluruhan teks yang ada. Ringkasan ini meliputi topik dari teks asal.
c. *Evaluatif* Yaitu melibatkan pembuatan sebuah pertimbangan pada teks asal, seperti suatu tinjauan ulang atau opini.

d. *User-Focused* Yaitu ringkasan yang dibuat berdasarkan topik yang dipilih oleh user, sering yang merupakan jawaban dari query yang dimiliki oleh user.
e. *Generic* Yaitu sifatnya lebih umum dan berdasarkan pada teks aslinya.
f. *Single Document* (Dokumen tunggal) yaitu ringkasan yang dihasilkan merupakan ringkasan dari satu dokumen.
g. *Multi Document* Pada peringkasan *multi document*, ringkasan merupakan hasil ringkasan dari banyak dokumen.

2.2.Proses Peringkasan Teks

Secara umum terdapat tiga tahapan dalam proses peringkasan teks, yaitu sebagai berikut[22]:

1. *Topic Identification*

   Tahapan ini meliputi identifikasi faktor yang sangat penting tentang apa yang dibicarakan dala teks tersebut. Ada beberapa teknik untuk melakukannya, diantaranya:

   i. Dalam beberapa tipe teks, informasi penting terdapat dalam bagianbagian tertentu dalam teks tersebut, seperti dalam judul, kalimat pertama, kalimat terakhir dan lain sebagainya.
   ii. Beberapa kata atau frase mengidentifikasi intisari dari suatu teks.
   iii. Beberapa kata cenderung muncul lebih. Ini dijadikan faktor penentu topik dari suatu teks (*word frequency*).
   iv. Ada juga topik diidentifikasi dari jumlah pengertiannya atau makna dibandingkan kata.

2. *Interpretation*

   Interpretasi peringkas yang ekstraktif berdasarkan pada metode yang digunakan, sedangkan pada peringkas yang abstraktif interpretasi ditunjukkan dengan cara penggabungan pengertian yang serupa menjadi satu, penghilangan redudansi dan lain sebagainya.

3. *Generating*

Tahapan ini adalah pembangkitan atau pembentukan hasil akhir. Terdiri dari penggabungan frase, pencetakan kata atau frase dan pembangkitan kalimat. Terdapat beberapa metode yang dapat digunakan, diantaranya :

a. *Extraction*, hasil akhir berisi kalimat atau frase yang dihasilkan setelah semua tahap proses pada *text summarization* selesai dilakukan.
b. *Topic list*, hasil akhir berisi kata-kata yang sering muncul atau penggabungan pengertian yang telah diinterpretasi.

2.3.Kata

Kata adalah kesatuan terkecil yang diperoleh sesudah sebuah kalimat dibagi atas bagian-bagiannya dan mengandung suatu ide. kata adalah kumpulan huruf atau bunyi yang mengandung arti[26].

Kategori kata berdasarkan sintaksisnya terdiri dari lima kelas kata yaitu :

1. Kata Benda (nomina)
   Adalah kata yang mengacu pada manusia, binatang, benda dan konsep atau pengertian.
2. Kata Kerja (verba)
   Adalaha kata yang menyatakan tindakan
3. Kata Sifat (adjektiva)
   Adalah kata yang memberi keterangan yang lebih khusus tentang sesuatu yang dinyatakan oleh manusia dalam kalimat.
4. Kata Keterangan (adverbial)
   Adalah kategori yang mendampingi adjektiva, numerilia atau preposisi dalam kontruksi sintaksis.
5. Kata Tugas
   Adalah kata yang hanya memiliki arti gramatikal dan tidak memiliki arti leksikal.

2.4.Kalimat

Kalimat adalah satuan terkecil dalam wujud lisan atau tulisan, yang mengungkapkan pikiran yang utuh. Sebagaimana menurut Sakri, Adjat[27] yang

menyatakan, bahwa kalimat dalam tulisan terdiri atas deret kata yang dimulai dengan huruf kapital dan diakhiri dengan tanda titik(.), tanda tanya(?), tanda seru(!).

Unsur-unsur kalimat terdiri dari kata, kelompok dan lagu kalimat. Pada kalimat terdapat pengaturan hubungan kedudukan antara bagian-bagiannya. Berdasarkan jabatannya kalimat terdiri dari :

1. Subyek, yaitu bagian yang menjadi pangkal atau pokok permasalahan.
2. Predikat, yaitu bagian yang menerangkan subyek, biasanya berdiri sesudah subyek
3. Obyek, yaitu bagian yang menjadi tujuan.
4. Keterangan, yaitu bagian yang menunjukkan waktu (keterangan waktu), tempat (keterangan tempat), alat (keterangan alat), dan sebagainya.

Sedangkan kalimat berdasarkan fungsinya dapat dikategorikan sebagai berikut:

1. Kalimat pernyataan
2. Kalimat pertanyaan
3. Kalimat perintah
4. Kalimat seruan

2.5.Paragraf

Paragraf disebut juga alinea. Paragraf merupakan inti penuangan pikiran dalam sebuah karangan. Kalimat-kalimat dalam paragraf memperlihatkan kesatuan pikiran atau mempunya keterkaitan dalam membentuk gagasan atau topic. Terdapat dua syarat dalam membentuk paragraf [27].

1. Menulis pernyataan (kalimat) tentang pokok bahasan dengan baik.
2. Menggali pola susunan rincian dengan patut.

2.6.Berita

Berita adalah laporan mengenai fakta atau ide terbaru yang benar, dan penting bagi sebagian besar khalayak, melalui media berkala seperti surat kabar, radio, televise, atau media internet.

Berita atau news adalah laporan mengenai suatu peristiwa atau kejadia yang terbaru (aktaul), laporan mengenai fakta-fakta yang actual, menarik perhatian, dinilai penting, atau luar biasa[28].

Susunan berita umumnya terdiri dari empat bagian, yaitu:

1. Headline, kepala berita atau judul berita.
2. Dateline, yaitu waktu dan nama tempat berita dibuat atau diperoleh.
3. Teras berita (Lead)

   Adalah bagian berita yang terletak dialinea atau paragraf pertama, setelah head dan dateline sebelum badan atau isi berita. Biasanya berisi fakta penting dengan mengedepankan unsur 5W+1H (what, who, when,where,why,dan how).
4. Tubuh berita (News Body)

   Berisi penjelasan atau uraian rinci unsur 5W+1H, baik yang sudah dikemukakan dalam teras berita maupun yang belum diungkapkan. Penulisan tubuh berita untuk melanjutkan apa yang sudan tertuang dalam teras yang mencerminkan pokok-pokok terpenting isi berita. Biasanya berupa kutipan dari isi berita atau kutipan isi pembicaraan nara sumber yang paling menarik.

2.7. Algoritma *Text Summarization*

2.7.1. Pra Proses (*Preprocessing)*

Preprocessing adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya. Inputan awal pada proses ini adalah berupa dokumen. Pada umumnya *preprocessing* memiliki beberapa tahapan yaitu *case folding, tokenizing, stop word removal, stemming,* dan lain-lain. *Preprocessing* pada penelitian ini terdiri dari beberapa tahapan, yaitu: proses pemecahan kalimat, proses case folding, proses tokenizing kata, dan proses stop word removal

2.7.1.1. Pemecahan Kalimat

Memecah dokumen menjadi kalimat-kalimat merupakan langkah awal tahapan *preprocessing*. Pemecahan kalimat yaitu proses memecah string teks dokumen yang panjang menjadi kumpulan kalimat-kalimat. Dalam memecah

dokumen menjadi kalimat-kalimat menggunakan fungsi **split(),** dengan tanda titik ".", tanda tanya "?" dan tanda tanya "!" sebagai delimiter untuk memotong string dokumen [22].

Tabel 2. 1 Contoh pemecahan kalimat

Kalimat	Hasil pemecahan kalimat
Manajemen transaksi elektronik. Pengetahuan antar individu. Dalam manajemen pengetahuan, terdapat transfer pengetahuan elektronik	Manajemen transaksi elektronik Pengetahuan antar individu Dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik

2.7.1.2. Case *Folding*

Dokumen mengandung berbagai variasi dari bentuk huruf sampai tanda baca. Variasi huruf harus diseragamkan (menjadi huruf besar saja atau huruf kecil saja) dan tanda baca dihilangkan untuk menghilangkan *noise* pada saat pengambilan informasi. Hal ini dapat dilakukan dengan *case folding. Case folding* adalah tahapan proses mengubah semua huruf dalam teks dokumen menjadi huruf kecil, serta menghilangkan karakter selain a-z.[22].

Tabel 2. 2 Contoh *case folding*

Kalimat	Hasil *case folding*
Manajemen transaksi elektronik Pengetahuan antar individu Dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik	manajemen transaksi elektronik pengetahuan antar individu dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik

2.7.1.3. Tokenizing

Tokenizing adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan men-scan kalimat dengan pemisah (*delimiter*) *white space* (spasi, tab, dan *newline*)[22].

Tabel 2. 3 Contoh *Tokenizing*

Case folding kalimat	Hasil *Tokenizing*
manajemen transaksi elektronik pengetahuan antar individu dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik	Manajemen transaksi elektronik pengetahuan antar

	individu dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik

2.7.1.4. Stoplist / *Stop Word Removal*

Penghapusan *Stopword* merupakan proses penghilangan kata *stopword*. *Stopword* adalah kata - kata yang sering kali muncul dalam dokumen namun arti dari kata-kata tersebut tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagainya [22].

Tabel 2. 4 Contoh *Stop Word Removal*

Tokenizing kalimat	Hasil *Stop Word Removal*
manajemen transaksi elektronik pengetahuan antar individu dalam manajemen pengetahuan terdapat transfer pengetahuan elektronik	manajemen transaksi elektronik pengetahuan individu manajemen transfer

2.7.1.5. *Stemming*

Menurut Zaman B. dan E Winarko [24] *stemming* adalah proses pemetaan dari penguraian berbagai bentuk kata baik itu *prefix*, *sufix*, maupun gabungan antara *prefix* dan *sufix* (*confix*), menjadi bentuk kata dasarnya.

Algoritma *stemmer* yang diperkenalkan Nazief dan Adriani didefinisikan sebagai berikut (Andita, 2010):

1. Di awal proses *stemming* dan setiap langkah yang selanjutnya dilakukan, lakukan pengecekan hasil proses stemming kata yang di-input-kan pada

langkah tersebut ke kamus kata dasar. Jika kata ditemukan, berarti kata tersebut sudah berbentuk kata dasar dan proses stemming dihentikan. Jika tidak ditemukan, maka langkah selanjutnya dilakukan.

2. Hilangkan *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”). Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
    a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hilangkan *derivation prefixes*.
    a. Langkah 4 berhenti jika :
        i. Terjadi kombinasi awalan dan akhiran yang terlarang.
        ii. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
        iii. Tiga awalan telah dihilangkan.
    b. Identifikasikan tipe awalan dan hilangkan. Awalan terdiri dari dua tipe :
        i. Standar (“di-”, “ke-”, “se-”) yang dapat langsung dihilangkan dari kata.
        ii. Kompleks (“me-”, “be-”, “pe”, “te-”) adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 2.1 untuk mendapatkan hasil pemenggalan yang tepat.

c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus kata dasar. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses *recoding* dilakukan dengan mengacu pada aturan pada Tabel 2.1. *Recoding* dilakukan dengan menambahkan karakter *recoding* di awal kata yang dipenggal. Pada Tabel 2.1, karakter *recoding* adalah huruf kecil setelah tanda hubung („-“) dan terkadang berada sebelum tanda kurung. Sebagai contoh, kata “menangkap” (aturan 15), setelah dipenggal menjadi “nangkap”. Karena tidak valid, maka recoding dilakukan dan menghasilkan kata “tangkap”.
6. Jika semua langkah gagal, maka input kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

Tabel 2.1 Aturan pemenggalan Awalan Stemmer Nazief dan Adriani

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V... | be-rV...
2	berCAP...	ber-CAP... dimana C!=“r“ & P!=“er“
3	berCAerV...	ber-CaerV... dimana C!=“r“
4	Belajar	bel-ajar
5	beC1erC2...	be-C1erC2... dimana C1!={“r“|“l“}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV... dimana C!=“r“
8	terCP...	ter-CP... dimana C!=“r“ dan P!=“er“
9	teC1erC2...	te-C1erC2... dimana C1!=“r”
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe{r|l}...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q}...	meng-{g|h|q}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV…
19	mempV...	mem-pV... dimana V!=„e“
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
22	perCAP	per-CAP... dimana C!=“r“danP!=“er“

23	perCAerV...	per-CAerV... dimana C!="r"
24	pem{b|f|V}...	pem-{b|f|V}...
25	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
26	pen{c|d|j|z}...	pen-{c|d|j|z}...
27	penV...	pe-nV... | pe-tV...
28	peng{g|h|q}...	peng-{g|h|q}...
29	pengV...	peng-V... | peng-kV...
30	penyV...	peny-sV…
31	pelV...	pe-lV... kecuali "pelajar" yang menghasilkan "ajar"
32	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
33	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!="er"

Keterangan simbol huruf :

C: huruf konsonan

V: huruf vokal

A: huruf vokal atau konsonan

P: partikel atau fragmen dari suatu kata, misalnya "er"

2.7.2. Algoritma *TF-IDF*

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *Term frequency* (TF) merupakan frekuensi kemunculan kata (*t*) pada kalimat (*d*). *Document frequency* (DF) adalah banyaknya kalimat dimana suatu kata (*t*) muncul.

Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen[22]. Pada aloritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu :

$$W_{dt} = tf_{dt} * IDF_t \quad \textbf{(2.1)}$$

Dimana:

d = dokumen ke-d

t = kata ke-t dari kata kunci

W= bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = Inversed Document Frequency

IDF = log2 (D/df)

D = total dokumen

df = banyak dokumen yang mengandung kata yang dicari

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses sorting/pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, demikian sebaliknya.

Inverse Document Frequency memperhatikan kemunculan *term* pada kumpulan dokumen. Pada metode ini, *term* yang dianggap bernilai/berharga adalah *term* yang jarang muncul pada koleksi/ kumpulan dokumen[20]. Persamaan IDF adalah sebagai berikut:

$$IDF(t) = log \frac{N}{df(t)} \quad \textbf{(2.2)}$$

Dimana *df(t)* adalah banyak dokumen yang mengandung *term t*.

TF*IDF merupakan kombinasi metode TF dengan metode IDF. Sehingga persamaan TF*IDF adalah sebagai berikut:

$$TF * IDF(d,t) = TF(d,t) * IDF(t) \quad \textbf{(2.3)}$$

Perhitungan bobot query relevance merupakan bobot hasil perbandingan kemiripan (similaritas) antara query yang dimasukkan oleh user terhadap keseluruhan kalimat. Sedangkan bobot similarity kalimat, merupakan bobot hasil perbandingan kemiripan antar kalimat.

2.7.3. Vector *Space Model*

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan *(similarity)* term dengan cara pembobotan *term*. Dokumen dipandang sebagi sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model,* sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query*[20].

Dokumen dan query direpresentasikan sebagai vektor.

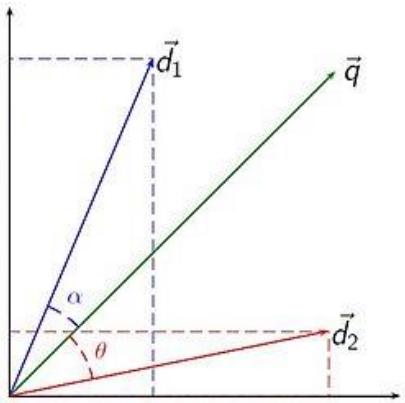
$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}) \quad \textbf{(2.4)}$$

$$q = (w_{1,q}, w_{2,q}, \ldots, w_{n,q}) \quad \textbf{(2.5)}$$

Setiap dimensi sesuai dengan istilah yang terpisah. Jika jangka terjadi dalam dokumen, nilainya dalam vektor adalah non-nol. Beberapa cara yang berbeda dari komputasi nilai-nilai, juga dikenal sebagai (istilah) berat, telah dikembangkan. Salah satu skema yang paling dikenal adalah tf-idf bobot (lihat contoh di bawah).

Definisi *istilah* tergantung pada aplikasi. Biasanya istilah yang satu kata, kata kunci , atau frase lagi. Jika kata-kata yang dipilih untuk menjadi persyaratan, dimensi dari vektor adalah jumlah kata dalam kosa kata (jumlah kata yang berbeda terjadi di corpus). Operasi vektor dapat digunakan untuk membandingkan dokumen dengan query.

Relevansi peringkat dokumen dalam pencarian kata kunci dapat dihitung dengan menggunakan asumsi dari dokumen kesamaan teori, dengan membandingkan deviasi dari sudut antara masing-masing vektor dokumen dan query vektor asli di mana query direpresentasikan sebagai jenis yang sama vektor sebagai dokumen.



Gambar 2.1 Ruang Vektor

Dalam prakteknya, lebih mudah untuk menghitung cosinus dari sudut antara vektor, bukan sudut itu sendiri:

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\|\mathbf{d_2}\| \, \|\mathbf{q}\|} \quad \textbf{(2.6)}$$

Dimana $\mathbf{d_2} \cdot \mathbf{q}$ adalah persimpangan (yaitu dot produk) dari dokumen (d $_2$ pada gambar di sebelah kanan) dan query (q pada gambar) vektor, $\|\mathbf{d_2}\|$ adalah norma vektor d $_2$, dan $\|\mathbf{q}\|$ adalah norma vektor q. norma dari vektor dihitung seperti:

$$\|\mathbf{q}\| = \sqrt{\sum_{i=1}^{n} q_i^2} \quad \textbf{(2.7)}$$

Karena semua vektor sedang dipertimbangkan oleh model ini adalah elementwise nonnegatif, nilai cosinus dari nol berarti bahwa permintaan dan vektor dokumen yang orthogonal dan tidak memiliki pertandingan (yaitu istilah permintaan tidak ada dalam dokumen yang sedang dipertimbangkan). Lihat kesamaan kosinus untuk informasi lebih lanjut.

2.8. Teknik Evaluasi Peringkasan Teks

Adapun Teknik yang digunakan untuk mengevaluasi hasil suatu ringkasan teks merupakan topik yang cukup sulit, baik evaluasi terhadap ringkasan yang

dihasilkan dari mesin peringkas otomatis ataupun ringkasan yang dihasilkan secara manual yang dibuat manusia, dikarenakan tidak terdapat definisi ringkasan ideal. Menurut Zaman B. dan E Winarko [24] metode untuk melakukan evaluasi terhadap hasil dari ringkasan secara umum dibagi 2, yaitu:

1. *Ekstrinsik*

   Metode evaluasi ekstrinsik adalah menghitung efektivitas dan akseptabilitas dari hasil ringkasan untuk tugas-tugas tertentu, misalnya *assessment* terhadap hasil ringkasan.

2. *Intrinsik*

   Metode evaluasi intrinsik adalah evaluasi yang dilakukan oleh sistem peringkas itu sendiri, misalnya menggunakan *F-Measures*. Evaluasi ini difokuskan pada tingkat koheren dan informatif dari hasil ringkasan.

Dalam penelitian ini, metode evaluasi yang digunakan adalah metode intrinsik penghitungan *F-Measure* berdasarkan perhitungan *Precision* dan *Recall* yang menurut Zaman B. dan E Winarko [24] merupakan standar evaluasi dalam penghitungan *information retrieval*. evaluasi perhitungan *information retrieval* dengan menggunakan *Precision* dan *Recall* juga dapat digunakan dalam evaluasi perhitungan peringkas teks otomatis.

$$Recall = \frac{\sum Kalimat\ Relevan\ yang\ dihasilkan\ Sistem}{\sum Kalimat\ ringkasan\ relevan} \quad \textbf{(2.8)}$$

$$Precision = \frac{\sum Kalimat\ Relevan\ yang\ dihasilkan\ Sistem}{\sum Kalimat\ yang\ dihasilkan\ sistem} \quad \textbf{(2.9)}$$

Kombinasi antara nilai *recall* dan *precision menghasilkan nilai* f-measure.

$$F - measure = \frac{2 * precision * recall}{recall + precision} \quad \textbf{(2.10)}$$

Dalam metode intrinsik, *precision* dan *recall* digunakan untuk mengukur kualitas ringkasan otomatis dengan cara membandingkan ringkasan otomatis dengan ringkasan manual (buatan manusia). Kemudian hasil akhir akan didapatkan dengan

cara penggabungan nilai *recall* (2.8) dan *precision* (2.9) yang disebut dengan nilai *F-measures* (2.10). Masalah dalam metode ini adalah dalam menentukan kalimat relevan karena pasti terdapat perbedaan pendapat antar subyek pembuat ringkasan dalam memilih kalimat. Misal subyek A menganggap kalimat *x* sebagai kalimat utama tetapi bisa saja subyek B berpendapat sebaliknya bahwa kalimat *x* kurang begitu penting. Untuk mengatasi hal ini digunakan beberapa metode seperti suara terbanyak (*majority opinion*), gabungan (*union*), atau irisan (*intersection*)[24].