

---

# Rancang Bangun Portal Berita Dewan Legislatif Dengan Fitur Peringkasan Teks Otomatis Menerapkan Term Frequency Inverse Document Frequency

Ivonne Yusriputri Alition, Universitas Ciputra, UC Town, Surabaya  
David Boy Tonara, Universitas Ciputra, UC Town, Surabaya

---

## ABSTRAK

Pemilu adalah salah satu perwujudan asas demokrasi negara Indonesia. Salah satunya adalah pemilu Calon Legislatif (Caleg). Namun, setelah berakhirnya pemilu masyarakat tidak dapat dengan mudah mendapatkan berita mengenai anggota legislatif yang telah terpilih. Padahal, rekam jejak anggota legislatif sangat dibutuhkan sebagai pertimbangan pada pemilu berikutnya. Oleh karena itu, akan dilakukan rancang bangun aplikasi berita caleg berbasis *mobile* yang memuat berita tokoh anggota legislatif yang diringkas menggunakan metode *Term Frequency – Inverse Document Frequency*. Rancang bangun ini dibuat agar aplikasi yang dapat diakses dimana saja ini dapat meningkatkan kesadaran masyarakat untuk memperoleh informasi mengenai anggota legislatif yang dipilih apakah telah memiliki rekam jejak yang baik. Hasil penelitian menunjukkan tingkat akurasi pemahaman pengguna atas hasil ringkasan sebesar 76,8% dan tingkat akurasi stuktur hasil ringkasan sebesar 73,4% dari 30 data *sampling* yang digunakan.

Kata kunci: Berita anggota legislatif, Ringkasan Teks Otomatis, Sistem Temu Balik Informasi, *Term Frequency – Inverse Document Frequency*,

---

## 1. Pendahuluan

### 1.1. Latar belakang

Indonesia merupakan salah satu negara yang sangat menjunjung demokrasi, dimana pemilu menjadi salah satu sarana dalam perwujudan asas demokrasi tersebut. Dalam pasal 1 ayat (1) Undang-Undang Republik Indonesia Nomor 8 Tahun 2012 menyatakan bahwa Pemilihan Umum, selanjutnya disebut Pemilu, adalah sarana pelaksanaan kedaulatan rakyat yang dilaksanakan secara langsung, umum, bebas, rahasia, jujur, dan adil dalam Negara Kesatuan Republik Indonesia berdasarkan Pancasila dan Undang-Undang Dasar Negara Republik Indonesia Tahun 1945.

Beberapa pemilu yang dilaksanakan di Indonesia adalah pemilu legislatif DPR dan DPRD. Tetapi setelah pemilihan umum berakhir, kebanyakan masyarakat tidak dapat dengan mudah serta tidak memiliki kesadaran yang cukup untuk mengetahui rekam jejak dari anggota legislatif yang telah terpilih. Padahal, beberapa tokoh anggota legislatif yang telah terpilih tersebut biasanya kembali mencalonkan diri pada pemilihan umum di periode

berikutnya atau biasa disebut dengan caleg *incumbent*.

Selain itu, minat baca masyarakat Indonesia yang rendah juga berpengaruh dalam kurangnya kesadaran masyarakat untuk mencari tahu mengenai rekam jejak anggota legislatif yang tengah menjabat. Menurut data UNESCO pada tahun 2011, indeks minat baca masyarakat Indonesia baru mencapai 0,001 yang berarti dari setiap 1.000 penduduk hanya satu orang saja yang memiliki minat baca (Republika, 2013).

Dengan merujuk pada permasalahan kurangnya kemudahan dan kesadaran masyarakat akan rekam jejak anggota legislatif, maka salah satu solusinya adalah dengan membuat aplikasi *mobile* yang menyediakan berita dari situs-situs berita yang ada mengenai tokoh anggota legislatif serta memberikan fitur peringkasan teks berita menjadi sebuah deskripsi singkat yang merangkum keseluruhan isi *headline*. Sehingga, masyarakat di pemilihan umum berikutnya dapat memilih dengan informasi yang baik mengenai rekam jejak anggota legislatif berdasarkan kinerja yang telah dilakukan oleh anggota legislatif tersebut selama menjabat.

Berdasarkan latar belakang masalah yang telah

diuraikan, maka tujuan dari penelitian ini adalah untuk membangun aplikasi portal berita anggota legislatif berbasis *mobile* dengan fitur otomatisasi peringkasan teks otomatis yang akan menghasilkan struktur ringkasan berita yang baik dari tokoh anggota legislatif.

## 2. Landasan Teori

Landasan teori ini berisikan penjelasan teori yang digunakan untuk membantu pelaksanaan penelitian rancang bangun portal berita dewan legislatif dengan fitur peringkasan teks otomatis menggunakan algoritma *Term Frequency-Inverse Document Frequency*.

### 2.1. Web mining

*Web Mining* disebut sebagai penerapan teknologi data mining untuk data web. Terdapat tiga pengkategorian *web mining* berdasarkan bagian yang akan di *mining* yaitu *content mining*, *structure mining*, dan *usage mining*. *Content mining* merupakan proses untuk mendapatkan informasi yang berguna dari konten di *website* (*text, image, audio, video*). *Structure mining* merupakan proses untuk mengkategorikan halaman *website* berdasarkan kesamaan antar situs. Sedangkan, *usage mining* merupakan proses untuk menganalisa kebiasaan pengguna saat mengakses *website* untuk menghasilkan konten *website* yang lebih diinginkan oleh pengguna (Lalani, 2003).

#### 2.1.1 Google websearch query parameter

*Request* dari *Google Search* menggunakan perintah standar HTTP GET. Kebanyakan dari *request* pencarian terdiri dari satu atau lebih *term*. *Term* pencarian tersebut tampil sebagai parameter *value* pada *request* pencarian. ID Parameter yang ditulis dalam url sebagai pasangan nama=value dipisahkan menggunakan karakter '&'. Berikut adalah penjelasan dari *query parameter* yang digunakan dalam penelitian ini (Google Developers, 2013):

- 1) q: parameter ini merupakan *term* pencarian, dimana tiap spasi akan diganti dengan karakter '+'. Contoh: q=horses+cows, artinya hasil pencarian yang ditampilkan adalah hasil dari kata 'horses' dan 'cow'.
- 2) start: parameter ini mengindikasikan hasil pencocokan pertama yang harus masuk dalam hasil pencarian. Contoh: start=10, artinya hasil pencarian yang diberikan dimulai dari halaman ke-2.
- 3) inurl: parameter ini membatasi hasil pencarian untuk kata tertentu yang berada dalam dokumen. Contoh: inurl:google+search, artinya hasil pencarian yang diberikan adalah hasil yang di urlnya harus terdapat kata 'google' dan kata 'search' harus muncul antara di judul, url, *link* ataupun *body* dari dokumen.

### 2.2. Sistem temu balik informasi (information retrieval system)

Sistem temu balik informasi merupakan sistem informasi yang berfungsi untuk menemukan informasi yang relevan

dengan kebutuhan pemakai. Pada dasarnya sistem temu balik informasi adalah suatu proses untuk mengidentifikasi, kemudian memanggil (*retrieve*) suatu dokumen dari suatu simpanan (*file*), sebagai jawaban atas permintaan informasi (Hasugian, 2006).

*Information retrieval* yang digunakan dalam penelitian ini adalah proses untuk mengambil *url* dari portal berita yang telah ditentukan serta proses untuk melakukan pemotongan pada DOM untuk mengambil konten yang dibutuhkan, yaitu berita, judul, dan tanggal artikel. Penentuan *url* menggunakan *term* pencarian dengan parameter tokoh anggota legislatif, alamat situs berita (kompas.com dan detik.com), dan jumlah halaman hasil pencarian (hanya hingga tiga halaman hasil pencarian).

### 2.3. Peringkasan teks dokumen otomatis

Peringkasan teks dokumen otomatis adalah bentuk ringkasan dari dokumen yang bertujuan untuk menghilangkan term yang dianggap tidak relevan atau redundant dengan menjaga inti makna dari dokumen. Sehingga, dokumen yang memiliki volume yang besar tetap dapat dipahami inti maknanya dengan cepat dan benar oleh user dokumen (Luthfiarta, dkk., 2013).

Dalam sistem temu balik informasi terdapat beberapa karakteristik, salah satunya adalah proses pencocokan yang dilakukan secara parsial (*partial match*) dan hanya mencari hasil temu balik yang terbaik yaitu hasil temu balik yang relevan (*relevant matching*). Dimana, model dalam sistem temu balik informasi yang diambil adalah model yang bersifat probabilistik dan proses inferensi yang dilakukan menggunakan metode induksi. Selain itu, karakteristik lainnya adalah *query* diberikan dalam bahasa alami (*natural language*) dan dalam bentuk yang tidak lengkap (Wibowo, 2011).

Ada lima titik perhatian atau fokus dalam kajian tentang temu kembali informasi yaitu perpindahan informasi dalam sistem komunikasi, pemikiran tentang informasi yang diinginkan, efektifitas sistem dan perpindahan informasi, hubungan antara informasi dengan penciptanya, serta hubungan antara informasi dengan pemakai (Belkin, 1985). Tujuannya adalah untuk mempelajari proses temu kembali, membentuk, membangun dan mengevaluasi sistem temu kembali yang dapat memberikan informasi yang diinginkan secara efektif antara pengarang dan pemakai.

### 2.4. Term Frequency – Inversed Document Frequency (TF-IDF)

Algoritma TF-IDF merupakan algoritma pembobotan yang sering digunakan dalam pengumpulan informasi. Perhitungan bobot digunakan untuk melakukan penyaringan atas kata yang sering muncul dan seberapa penting suatu kata bagi sebuah dokumen. Rumus perhitungan pembobotan TF-IDF dapat dituliskan dalam persamaan (1) berikut:

$$TFIDF(t) = TF * \text{Log} \frac{D}{DF} \quad (1)$$

Dimana TF adalah jumlah munculnya suatu *term* dalam suatu dokumen, sedangkan *IDF* adalah perhitungan logaritma pembagian jumlah dokumen(D) dengan frekuensi dokumen(DF) yang memuat suatu *term*. *TFIDF* adalah hasil perkalian nilai *TF* dengan *IDF* untuk sebuah *term* dalam dokumen (Lutfiartha, 2013).

Setelah didapatkan bobot TF-IDFnya maka diambil beberapa bobot tertinggi per kalimat sebagai hasil ringkasan. Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen (Grossman, 1998).

### 3. Desain dan Analisis Sistem

Bagian ini akan menjelaskan mengenai desain dan analisis sistem yang digunakan dalam pembuatan rancang bangun anggota dewan legislatif dengan fitur peringkasan teks otomatis menerapkan algoritma *TFIDF(Term Frequency Inverse Document Frequency)*.

#### 3.1. Analisis kebutuhan

Hal pertama yang perlu dilakukan dalam analisis kebutuhan sistem adalah menentukan dan mengungkapkan kebutuhan sistem. Kebutuhan sistem terbagi menjadi dua yaitu: kebutuhan sistem fungsional dan kebutuhan sistem nonfungsional.

##### 3.1.1. Kebutuhan fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung dengan sistem. Kebutuhan fungsional dari aplikasi ini meliputi kebutuhan *user* untuk memilih berita (berdasarkan region, partai dan tokoh), melihat *summary* berita tokoh, memberikan laporan mengenai berita yang salah, melihat data berita asli, serta kebutuhan administrator untuk memanipulasi data legislatif, memanipulasi data berita, melihat data legislatif, melihat data berita yang di laporkan salah.

##### 3.1.2. Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak secara langsung terkait dengan fitur tertentu di dalam sistem. Dalam membangun aplikasi ini kebutuhan perangkat keras dibagi menjadi perangkat keras administrator sistem dan perangkat keras *user*. Perangkat keras administrator berupa komputer server.

*Komputer server* digunakan untuk menempatkan

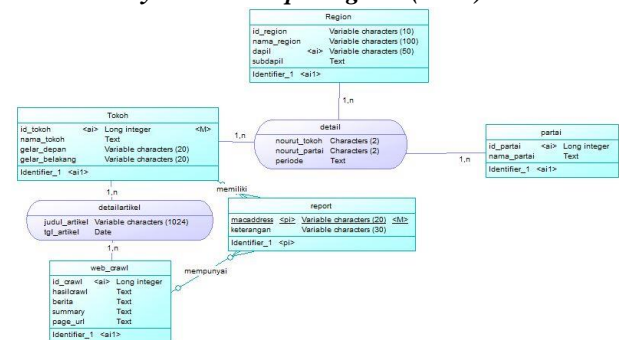
program dan database aplikasi peringkasan berita legislatif secara online. Sedangkan, perangkat keras *user* berupa handphone dengan *Operating System* Android. Adapun perangkat lunak yang digunakan dalam membangun aplikasi ini adalah sebagai berikut:

- Aplikasi yang digunakan untuk melakukan perancangan terhadap desain sistem dan database ke dalam sebuah model. Dalam hal ini, penulis menggunakan aplikasi *Enterprise Architect* untuk perancangan UML (*Unified Modelling Language*) dan *Sybase* untuk ERD (*Entity Relationship Diagram*).
- Aplikasi pembangun sistem/bahasa pemrograman, dimana bahasa pemrograman merupakan bahasa yang digunakan untuk mendefinisikan program komputer. Bahasa pemrograman yang digunakan adalah *Visual Basic 2010* dan *PHP versi 5.2.8*
- Aplikasi web server yang berfungsi sebagai web server dan pengolah basis data. Aplikasi yang digunakan adalah WAMP.

#### 3.2. Desain Sistem

Setelah melakukan analisis terhadap sistem, maka hal selanjutnya yang dilakukan adalah membuat rancangan atau desain sistem. Dalam pengembangan aplikasi ini beberapa rancangan yang dibuat adalah *Entity Relationship Diagram* (ERD) dan *Data Flow Diagram* (DFD) level 1.

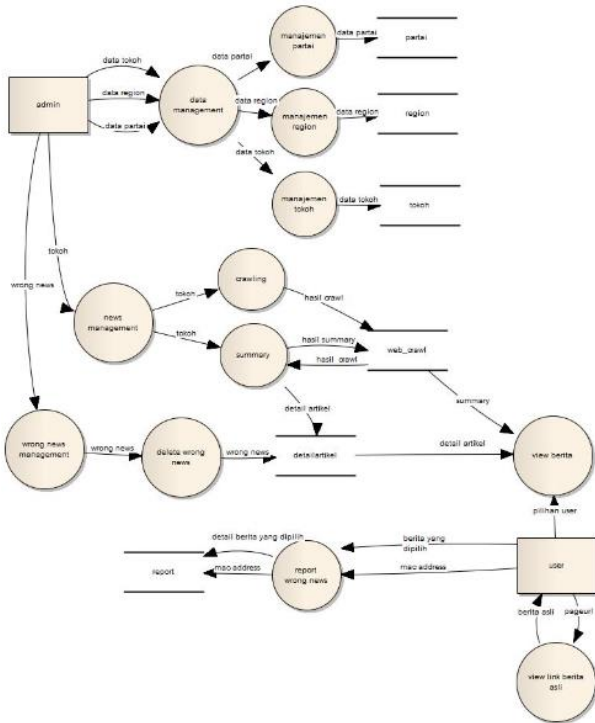
##### 3.2.1. Entity Relationship Diagram (ERD)



Gambar 1. Entity Relationship Diagram (ERD)

Pada Gambar 1, terlihat bahwa entiti tokoh berisikan data tokoh, entiti region berisikan data region, dan entiti partai berisikan data partai. Entiti web\_crawl adalah entiti yang digunakan untuk menampung hasil dari proses *crawl* maupun *summary* dalam sistem. Entiti detail merupakan entiti yang berisikan detail yang lebih lengkap dari tokoh legislatif. Sedangkan, entiti detailartikel merupakan entiti yang berisikan detail dari artikel tokoh- tokoh calon legislatif.

3.2.2. Data Flow Diagram (DFD) level 1



Gambar 2. Data Flow Diagram (DFD) level 1

Gambar 2 menunjukkan dalam proses *data management*, sistem menerima data tokoh, data region, data partai dari admin dan data tersebut akan dimasukkan dalam database tokoh region, dan partai. Dalam proses *news management*, admin memberikan data tokoh yang akan digunakan sistem dalam proses *crawling* dan *summary*. Hasil *crawling* dan hasil *summary* dimasukkan ke dalam Tabel ‘web\_crawl’, sedangkan detail artikel dari proses *summary* dimasukkan pada Tabel ‘detailartikel’. Untuk proses manajemen berita yang salah, admin akan memberikan data berita yang salah pada sistem dan akan menghapusnya pada Tabel ‘detailartikel’.

Pada proses *view berita*, user memberikan data pilihan user dan untuk menampilkan berita, user akan mengambil data *summary* dan detail artikel dari Tabel ‘web\_crawl’ dan ‘detailartikel’. Sedangkan pada proses *report wrong news*, user akan memberikan data berita yang dipilih beserta *macaddress* dan data tersebut akan dimasukkan oleh sistem kedalam Tabel ‘report’. Untuk melihat url berita asli, sistem menerima data *pageurl* dari user dan sistem akan memberikan data berita asli.

4. Implementasi dan Pengujian

Implementasi ini terbagi atas 2 proses, yaitu proses *crawling* dan proses *information retrieval*. Proses *crawling* dilakukan untuk mengambil *data training*, sedangkan

proses *information retrieval* mencakup proses *pre-processing* dan algoritma TF-IDF.

4.1.1. Sistem web crawling dan information retrieval

Objek pengujian pada penelitian ini menggunakan dua portal berita, yaitu *detik.com* dan *kompas.com*. *Website* yang digunakan untuk pengambilan data *training* adalah <http://news.detik.com/read/>, <http://news.detik.com/surabaya/>, dan <http://megapolitan.kompas.com> dan <http://regional.kompas.com>. Proses *web crawler* ini dimulai dengan melakukan *query* melalui *web browser* pada *visual basic* ke *google search*. Parameter *query* yang digunakan yaitu “www.google.com/search?q=[Nama+tokoh]+inurl:[detik.com atau kompas.com] &start=[0-20]”. Batasan pengambilan url hingga halaman ke tiga pada hasil *google search* atau *start=20* pada *google query*. Setelah itu, dilakukan analisa struktur DOM untuk mendapatkan letak dari alamat *url*, dan letak konten *data training* sehingga didapatkan *url*, berita dan judul konten.

```

1: Dim client As WebClient = New WebClient()
2: Dim isikompas As String = client.DownloadString(alamatweb)
3: // cut html till got judul
4: If isikompas.IndexOf("orange breadcrumb") > 0 Then
5:   Dim cutkompas = isikompas.Substring(isikompas.IndexOf("orange breadcrumb"))
6: // get judul
7:   Dim judulkompas = cutkompas.Substring(cutkompas.IndexOf("<h2>")+4)
8:   judulkompas = judulkompas.Substring(0, judulkompas.IndexOf("</h2>"))
9: // get artikel
10:  berita = cutkompas.Substring(cutkompas.IndexOf("<strong>")+8)
11:  berita = berita.Substring(0, berita.IndexOf("<hr class=""orange""/>") - 17)
12: End if
13: // get tanggal artikel
14: Dim tglartikel = alamatweb.Substring(alamatweb.IndexOf("read/") + 5).Substring(0, 10).Replace("/", "-")

```

Segmen 1. Contoh mengambil artikel dan judul berita (kompas.com)

Segmen 1 menunjukkan *code* yang digunakan untuk mendapatkan artikel dan judul berita dimulai dengan menggunakan *WebClient* untuk mendownload keseluruhan isi dari url dan ditampung pada variabel ‘*isikompas*’ menggunakan fungsi *client.DownloadString()*. Kemudian dilakukan pemotongan menggunakan fungsi *substring()* dimulai dari *class* “orange breadcrumb” yang ditampung pada variabel ‘*cutkompas*’. Judul artikel diambil menggunakan fungsi *substring()* dimulai dari tag “<h2>” hingga tag “</h2>” dan disimpan pada variabel ‘*judulkompas*’.

Untuk mendapatkan artikel berita menggunakan fungsi *substring()* pada variabel “*cutkompas*” dimulai dari index “<strong>”+8 hingga index pada tag “<hr class=“orange”/>” kemudian disimpan pada variabel ‘*berita*’. Sedangkan, untuk mengambil tanggal artikel pemotongan dimulai dari index “*read/*”+5, kemudian mengambil hingga 10 index berikutnya sehingga didapatkan tanggal artikel yang disimpan pada variabel ‘*tglartikel*’.

#### 4.1.2. Sistem peringkasan teks otomatis

Pada tahap ini proses awal yang dilakukan adalah proses *pre-processing*, yaitu tahapan untuk membuang karakter dalam data *training*.

```

1: // CASE FOLDING
2: // pisah per kalimat
3: Dim kalimat As String = Regex.Split(main.cleannews, "\s+")
4: Dim a1 = kalimat.Length - 1
5: Dim a2 = 0
6: Dim token(a1, 0) As String
7: For i = 0 To a1
8: // ubah kecil semua
9: kalimat(i) = kalimat(i).ToLower()
10: //hapus karakter selain huruf
11: kalimat(i) = Regex.Replace(kalimat(i), "[^a-z]", "")
12: // pisahkan per kata
13: Dim words() As String = kalimat(i).Split(New Char() {" "c})
14: For j = 0 To words.Length - 1
15: If a2 = 0 Or a2 < words.Length - 1 Then
16: a2 = words.Length - 1
17: ReDim Preserve token(a1, a2)
18: End If
19: // TOKENIZING
20: token(i, j) = words(j)
21: Next
22: Next

```

Segmen 2. Proses pre-processing

Segmen 2 menunjukkan proses *case folding*, dimana dalam penelitian ini karakter selain “a-z” akan dihilangkan serta mengubah keseluruhan huruf dalam data *training* menjadi huruf kecil. Data *training* tersebut akan dimasukkan dalam token-token dalam proses *tokenizing*. Setelah itu dilakukan proses *stopword* yaitu melakukan penghapusan *token* yang memiliki *string* yang terdapat pada daftar *stopword*. Proses berikutnya yang dilakukan adalah *stemming*, yaitu proses untuk mendapatkan kata dasar dari tiap-tiap token dengan menggunakan algoritma *porter stemmer*. Proses terakhir adalah pembuatan hasil ringkasan menggunakan algoritma peringkasan menggunakan *Term Frequency-Inverse Document Frequency*.

```

1: For i = 0 To kata.Count - 1
2: Dim df As Integer = 0
3: Dim idf As Double = 0.0
4: For j = 0 To tokens.Count - 1
5: //hitung ada berapa kemunculan 'kata'
6: Dim ex As New System.Text.RegularExpressions.Regex(kata(i))
7: Dim m As System.Text.RegularExpressions.MatchCollection
8: m = ex.Matches(tokens(j))
9: Dim row As DataRow = table.NewRow
10: table.Rows(i)(j + 1) = m.Count.ToString()
11: df = df + m.Count()
12: Next
13: idf = Math.Log10(df / tokens.Count)
14: table.Rows(i)(tokens.Count + 1) = df
15: table.Rows(i)(tokens.Count + 2) = idf
16: Next

```

Segmen 3. Proses TF-IDF (1)

Pada Segmen 3, variabel ‘df’ dibuat untuk menampung banyaknya jumlah kata yang muncul dengan mencocokkan *string* pada array ‘kata’ dengan *string* pada array ‘tokens’ dan akan disimpan pada tabel bernama ‘table’. Variabel ‘idf’ merupakan variabel yang dibuat

untuk menampung hasil perhitungan idf dengan menggunakan `math.log()`.

```

1: tableW.Rows.Add("Total")
2: For i = 0 To kata.Count - 1
3: Dim W As Double = 0.0
4: For j = 1 To tableW.Columns.Count - 1
5: Dim row As DataRow = tableW.NewRow
6: W = table(i)(j) * table(i)(tokens.Count + 2)
7: tableW.Rows(i)(j) = W
8: Next
9: Next
10: Dim max = New ArrayList()
11: For i = 1 To tableW.Columns.Count - 1
12: Dim total As Double = 0.0
13: For j = 0 To kata.Count - 1
14: total = total + tableW(j)(i)
15: Next
16: tableW.Rows(kata.Count)(i) = total
17: // array untuk tampung total
18: max.Add(total)
19: Next
20:

```

Segmen 4. Proses TF-IDF (2)

Pada Segmen 4, TableW dibuat untuk menampung hasil dari perhitungan TF-IDF, dan variabel ‘W’ dibuat untuk menampung hasil perkalian hasil TF dengan IDF pada ‘table’. Untuk dapat melihat nilai tertinggi maka dilakukan penjumlahan pada nilai W per kalimat dan disimpan pada variabel ‘total’ dan array ‘max’.

Penentuan kalimat yang paling penting untuk diambil sebagai ringkasan adalah dengan mengurutkan kalimat berdasarkan hasil bobot algoritma TF-IDF pada kalimat yang tertinggi yang terdapat pada array ‘max’ dengan fungsi `sort()`, dan kemudian mengambil jumlah kalimat sesuai dengan jumlah kalimat yang ditentukan sebagai hasil ringkasan.

Pada penelitian ini, jumlah kalimat yang diambil sebagai hasil ringkasan adalah 30% dari total keseluruhan kalimat dalam satu dokumen (berita). Sebagai contoh, total kalimat dalam satu dokumen adalah 10 kalimat sehingga hasil ringkasan yang dihasilkan nantinya akan berjumlah 3 kalimat dari kalimat yang memiliki bobot tertinggi.

#### 4.2. Pengujian Sistem

Bagian ini akan membahas pengujian sistem, yang terbagi atas tiga pengujian, yaitu dari pengujian *fungsional* aplikasi, pengujian *information retrieval*, dan pengujian akurasi ringkasan berita.

##### 4.2.1. Pengujian fungsional aplikasi

Pengujian ini dilakukan untuk melihat apakah responden dapat menggunakan aplikasi ini dengan baik. Metode pengujian yang dilakukan adalah dengan memberikan skenario pengujian seperti pada Tabel 1 kepada lima responden tanpa memberitahu cara untuk melakukannya, dan kemudian penulis melakukan pencatatan atas waktu yang digunakan responden untuk menyelesaikan tiap skenario pengujian.

**Tabel 1. Skenario Pengujian Fungsional Aplikasi**

1)	Melihat berita tokoh Sahat Tua P. Simanjuntak yang berasal dari region DPRD Jawa Timur dan partai ‘Golongan Karya’.
2)	Melihat link berita asli dari artikel “Ditunjuk Jadi Menpora, Imam Nahrawi: Ini Beban Buat Saya” dari tokoh ‘H.Imam Nahrawi’, region DPR RI, dan Partai Kebangkitan Bangsa.
3)	Melaporkan salah satu berita salah dari tokoh ‘H. Budi Leksono S.H.’ dari region ‘DPRD Surabaya’ dan partai ‘PDI Perjuangan’.

#### 4.2.2. Pengujian information retrieval

Pengujian *Information Retrieval* ini terbagi atas dua pengujian, yaitu pengujian akurasi ketepatan tokoh pada berita dan pengujian akurasi pengambilan berita dari proses *web crawling*. Pengujian akurasi ketepatan tokoh dilakukan dengan mengambil 30 data *sampling* dari proses *crawl*, kemudian dilakukan pencocokan apakah data berita tersebut merupakan berita dari tokoh yang tepat atau tidak. Sedangkan, pengujian akurasi pengambilan berita dilakukan dengan mengambil 30 data *sampling* yaitu data hasil *crawl* kemudian dilakukan pencocokan apakah data tersebut sudah mencakup keseluruhan berita pada *website* asli.

#### 4.2.3. Pengujian akurasi ringkasan berita

Pengujian akurasi ringkasan berita ini bertujuan untuk mengetahui apakah pengguna memahami hasil ringkasan yang dihasilkan dan apakah hasil ringkasan tersebut telah memiliki struktur yang baik. Pengujian ini dilakukan dengan menggunakan lima(5) hasil *summary* yang dihasilkan oleh aplikasi dan diujikan menggunakan kuisioner kepada 20 responden. Metode perhitungan pengujian menggunakan skala *likert* dengan skala angka 5 hingga angka 1. Angka 5 memiliki maksud bahwa responden sangat setuju dengan pernyataan tersebut, angka 4 dengan maksud setuju, angka 3 dengan maksud netral, angka 2 dengan maksud tidak setuju, dan angka 1 dengan maksud bahwa responden sangat tidak setuju dengan pernyataan tersebut.

#### 4.3. Hasil Pengujian

Bagian ini menunjukkan hasil dari tiga pengujian yang telah dilakukan sebelumnya. Hasil pengujian tersebut, yaitu hasil pengujian *fungsional aplikasi*, hasil pengujian *information retrieval*, hasil pengujian *akurasi ringkasan berita*

#### 4.3.1. Hasil pengujian fungsional aplikasi

**Tabel 2. Hasil pengujian fungsional aplikasi**

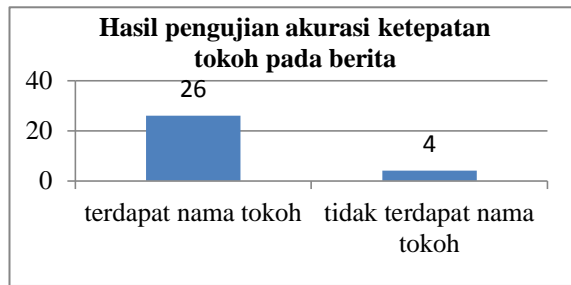
No	Skenario Pengujian	Presentase Keberhasilan	Presentase Kegagalan	Rata-rata waktu selesai
1	Melihat berita tokoh Sahat Tua P. Simanjuntak yang berasal dari region DPRD Jawa Timur dan partai ‘Golongan Karya’.	100 %	0 %	18s
2	Melihat link berita asli dari artikel “Ditunjuk Jadi Menpora, Imam Nahrawi: Ini Beban Buat Saya” dari tokoh ‘H.Imam Nahrawi’, region DPR RI, dan Partai Kebangkitan Bangsa.	100 %	0	22s
3	Melaporkan salah satu berita salah dari tokoh ‘H. Budi Leksono S.H.’ dari region ‘DPRD Surabaya’ dan partai ‘PDI Perjuangan’.	100 %	0 %	21s

Berdasarkan Tabel 2, rata-rata waktu yang dibutuhkan user untuk melihat berita adalah 18 detik, sedangkan waktu untuk melihat link berita asli adalah 22 detik, dan rata-rata waktu yang dibutuhkan untuk melaporkan berita yang salah adalah 21 detik. Hasil pengujian aplikasi menunjukkan bahwa secara *fungsional*, aplikasi sudah cukup baik dan dapat digunakan oleh masyarakat.

#### 4.3.2. Hasil pengujian information retrieval

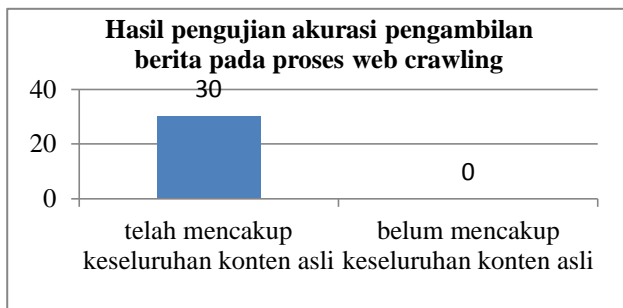
Dari hasil pengujian pada Bagan 1 dari 30 data *sampling*, jumlah berita yang terdapat nama tokoh adalah 26 berita dan jumlah berita yang tidak terdapat tokoh adalah 4 berita. Sehingga, dapat disimpulkan bahwa akurasi ketepatan tokoh berita mencapai 86,777%.





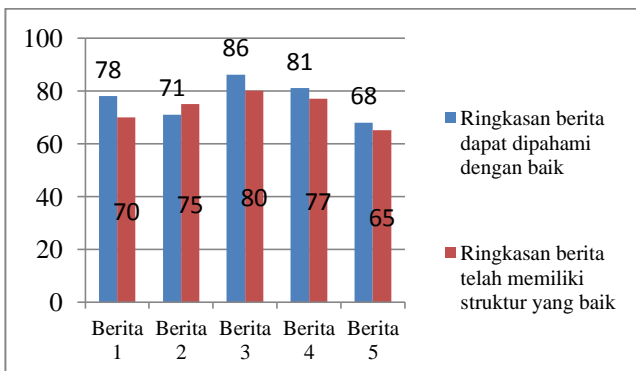
Bagan 1. Hasil pengujian akurasi ketepatan tokoh pada berita

Dari hasil pengujian pada Bagan 2 dari 30 data *sampling*, jumlah berita yang telah mencakup keseluruhan berita dari *website* asli adalah 30 berita dan tidak ada berita yang belum mencakup keseluruhan berita dari *website* asli. Sehingga, dapat disimpulkan bahwa akurasi ketepatan pengambilan berita dari proses *web crawling* mencapai 100%.



Bagan 2. Hasil pengujian akurasi pengambilan berita pada proses *web crawling*

#### 4.3.3. Hasil pengujian akurasi ringkasan berita



Bagan 3. Total nilai ketepatan pada pengujian akurasi pengambilan berita

Dari perhitungan skala *likert* dan interpretasi skor yang telah dilakukan, pada Bagan 3 maka akhir hasil kuisioner dapat diproses dengan mencari nilai persentase dari setiap pernyataan tersebut pada tiap berita. Setelah itu, dilakukan perhitungan rata-rata atas persentase dari tiap pernyataan pada lima berita tersebut. Berikut adalah kriteria

interpretasi skor yang diambil adalah dari pembagian hasil nilai sempurna, yaitu 100 dengan total skala *likert* dengan nilai 20 sebagai intervalnya: Angka 0% - 19,99% = Sangat Lemah, Angka 20% - 39,99% = Lemah, Angka 40% - 59,99% = Cukup, Angka 60% - 79,99% = Kuat, Angka 80% - 100% = Sangat Kuat.

No	Pernyataan	Rata-rata Persentase
1	Ringkasan berita dapat dipahami dengan baik	76,8%
2	Ringkasan berita telah memiliki struktur yang baik	73,4%

Bagan 4. Hasil rata-rata pengujian akurasi ringkasan berita

Dari hasil rata-rata pada Bagan 4, dapat disimpulkan bahwa nilai pemahaman responden atas ringkasan berita sebesar 76,8% dari total 20 responden yang masuk dalam kategori 'kuat' dari interpretasi skor. Sedangkan, nilai struktur ringkasan berita yang baik sebesar 73,4% dari total 20 responden yang masuk dalam kategori 'kuat' dari interpretasi skor. Sehingga dapat disimpulkan bahwa pernyataan hasil ringkasan berita sudah dapat dipahami oleh pengguna dan telah memiliki struktur yang baik.

## 5. Kesimpulan dan Saran

Bagian ini akan dijabarkan mengenai kesimpulan atas hasil penelitian dan juga saran apabila ada penerapan aplikasi yang memiliki kemiripan dengan aplikasi yang dibuat dalam penelitian ini ataupun jika ada peneliti lain yang ingin melanjutkan hasil penelitian ini.

### 5.1. Kesimpulan

Berdasarkan hasil uraian baik dari proses perancangan hingga pada proses implementasi yang dilakukan, didapatkan kesimpulan sebagai berikut:

- 1) Penelitian ini telah berhasil membuat *web crawling* untuk situs detik.com dan kompas.com dengan url tertentu dengan akurasi 100% untuk akurasi pengambilan berita dan 83,333% untuk akurasi ketepatan tokoh berita.
- 2) Pada penelitian ini telah berhasil dibuat aplikasi *mobile* untuk menampilkan ringkasan berita dari tokoh legislatif dengan menggunakan algoritma *Term Frequency Inverse Document Frequency* dengan nilai pemahaman responden atas ringkasan berita yang dihasilkan sebesar 76,8% dan nilai hasil struktur ringkasan berita yang baik sebesar 73,4%.
- 3) Aplikasi ini dapat dijalankan pada *device* sesungguhnya dan berjalan dengan baik berdasarkan hasil dari pengujian *funksional* aplikasi.

- 4) Pada proses *implementasi* ditemukan adanya keterbatasan pada proses *crawling* yaitu hanya +- 15 *query* per koneksi.

### 5.2. Saran

Beberapa saran untuk penelitian selanjutnya sehubungan dengan aplikasi berita legislatif ini adalah:

- 1) Sehubungan dengan perkembangan teknologi aplikasi *mobile* yang pesat, maka dapat dipertimbangkan untuk membuat aplikasi ini untuk *platform* lainnya, seperti *ios*, *windows phone*.
- 2) Penambahan beberapa fitur, seperti:
  - a) Fitur untuk mencari berita berdasarkan nama tokoh.
  - b) Fitur untuk memberikan rating baik buruknya tokoh berdasarkan berita yang ada.
- 3) Menggunakan *open source* untuk proses *crawling* karena keterbatasan *crawling* pada aplikasi ini.

### DAFTAR PUSTAKA

- Belkin, Nicholas, J., & Vickery, A. (1985). *Interaction in information systems : a review of research from document retrieval to knowledge-based systems*. Library and Informa-tion Research Report.
- Hasiguan, j. (2006). Penggunaan Bahasa Alamiah dan Kosa Kata Terkendali dalam Sistem. *Jurnal Studi Perpustakaan dan Informasi, Vol.2 No.2*.
- Google Developers. (2013, 26 September 2013). XML API Reference. Diperoleh 22 April,2015, dari [https://developers.google.com/custom-search/docs/xml\\_results](https://developers.google.com/custom-search/docs/xml_results).
- Lalani, A. S. (2003). Data Mining of Web Access Logs. *Minor Thesis, School of Computer Science and Information Technology, Faculty of Applied Science, Royal Melbourne Institute of Technology, Melbourne, Victoria, Australia*.
- Republika. (2013, 02 November). Perpusnas: Minat Baca Masyarakat Indonesia Masih Rendah. Diperoleh 11 Mei, 2015 dari <http://www.republika.co.id/berita/nasional/daerah/13/11/02/mvmvq4-perpusn-asminat-baca-masyarakat-indonesia-masih-rendah>.
- Salam, A., Zeniarja, J., & Luthfiarta, A. (2013). Algoritma Latent Semantic Analysis (LSA) Pada Peringkat Dokumen Otomatis Untuk Proses Clustering Dokumen.
- Wibowo, A. (2011). *Pengujian Kerelevanan Sistem Temu Kembali Informasi*. Batam: Politeknik Negeri Batam.