TESIS - IF185401

KOMPRESI MULTILEVEL PADA META HEURISTIK *FOCUSED WEB CRAWLER*

Dian Septiani Santoso
NRP. 5116201029

DOSEN PEMBIMBING

Dr.Ir. Raden Venantius Hari Ginardi, M.Sc.
NIP: 19650518 199203 1 003

PROGRAM MAGISTER

BIDANG KEAHLIAN ALGORITMA PEMROGRAMAN

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2019

[Halaman ini sengaja dikosongkan]

TESIS - IF185401

MULTILEVEL COMPRESSION ON META-HEURISTIC FOCUSED WEB CRAWLER

Dian Septiani Santoso
NRP. 5116201029

THESIS ADVISOR

Dr.Ir. Raden Venantius Hari Ginardi, M.Sc.
NIP: 19650518 199203 1 003

MASTER PROGRAM

DEPARTMENT OF INFORMATICS

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:
Dian Septiani Santoso
Nrp. 5116201029

Dengan judul :
Kompresi Multilevel Pada Meta Heuristik Focused Web Crawler

Tanggal Ujian : 16-1-2019
Periode Wisuda : 2019 Gasal

Disetujui oleh:

Dr. Ir. Raden Venantius Hari Ginardi, M.Sc.
NIP. 196505181992031003 (Pembimbing 1)

Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.
NIP. 196707271992031002 (Penguji 1)

Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.
NIP. 197411232006041001 (Penguji 2)

Royyana Muslim I, S.Kom., M.Kom., Ph.D.
NIP. 197708242006041001 (Penguji 3)

Dekan Fakultas Teknologi Informasi dan Komunikasi,

Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 197208091995121001

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tesis saya dengan judul:

KOMPRESI MULTILEVEL PADA META HEURISTIK *FOCUSED WEB CRAWLER*

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pusaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 21 Januari 2019

Dian Septiani Santoso

NRP: 5116201029

[Halaman ini sengaja dikosongkan]

Judul

Kompresi Multilevel Pada Meta Heuristik *Focused Web Crawler*

Nama Mahasiswa	: Dian Septiani S
NRP	: 5116201029
Pembimbing	: Dr.Ir. Raden Venantius Hari Ginardi, M.Sc

ABSTRAK

Focused Web Crawler merupakan metode pencarian website yang sesuai dengan pencarian yang diingikan oleh user. Karena berbasis pencarian atau pencocokan maka diperlukan metode untuk menghasilkan pencarian yang memiliki tingkat kecocokan yang baik. Untuk mendapatkan kecocokan yang baik maka diperlukan waktu yang lebih lama dibandingkan pencarian web crawler pada umumnya dengan memakai alogaritma DFS (Depth First Search) maupun BFS (Breadth First Search). Untuk mengatasi hal tersebut maka muncul sebuah ide pencarian *Focused Web Crawler* dengan menggunakan metode metaheuristik pencarian cuckoo yang digabung dengan pencarian pada data *history* pencarian yang disimpan. Namun dengan adanya penyimpanan data pada setiap kali pencarian link maka data akan semakin bertambah,oleh karena itu diperlukan sebuah cara untuk mengurangi kebutuhan ruang penyimpanan.

Cara yang dilakukan untuk mengurangi ruang penyimpanan dan tidak mengurangi nilai informasi dari data penyimpanan sebelumnya adalah dengan melakukan kompresi data. Dalam penelitian ini diusulkan metode kompresi data dengan melakukan kompresi multilevel menggunakan dua metode kompresi yaitu pengurangan kata dan kompresi string berbasis kamus. Dari hasil percobaan didapatkan rasio penghematan rata-rata sebesar 36.4%.

Untuk menguji hasil dari kompresi data yaitu dengan melakukan perbandingan hasil pencarian link menggunakan metode Knutt Morris Pratt (KMP) dari data yang belum terkompresi dengan data yang telah terkompresi. Hasilnya didapatkan bahwa maksimum presisi dengan nilai 1 dan recall sebesar 0.73 masih bisa didapatkan dengan metode yang diusulkan.

.

Kata kunci: *Focused Web Crawler, Pencarian Metaheuristik,Pencarian Cuckoo, Kompresi multilevel, Kompresi Berbasis Kamus, Pencarian link, Knutt Morris Pratt*

[Halaman ini sengaja dikosongkan]

Multilevel Compression on Meta-Heuristic *Focused Web Crawler*

Student Name	: Dian Septiani Santoso
NRP	: 5116201029
Supervisor	: Dr.Ir. Raden Venantius Hari Ginardi, M.Sc

ABSTRACT

Focused Web Crawler is a search method of a website that matches the search desired by the user. Because it's search-based or matching it's necessary to generate a search that has a good match rate. To get a good match it takes longer time than web crawler search in general by using DFS (Depth First Search) algorithm and BFS (Breadth First Search). To overcome this then comes to a search idea Focused Web Crawler by using metaheuristic method of cuckoo search combined with a search on search *history* data stored. But with the storage of data on each time the link search will increase the data, therefore needed a way to reduce storage space needs.

The way that is done to reduce storage space and does not reduce the value of information from previous data storage is to perform data compression. In this study proposed data compression method by doing multilevel compression using two methods of compression ie reduction of words and dictionary based string compression. It is expected that by doing multilevel compression, the data will be maximally compressed and does not reduce the value of information from existing data. From the experimental results obtained an average savings ratio of 36.4%.

To test the results of data compression that is by comparing the link search results using the Knutt Morris Pratt method (KMP) from data that has not been compressed with compressed data. The results obtained that the maximum precision with a value of 1 and recall of 0.73 can still be obtained by the proposed method.

***Keywords**: Focused Web Crawler, Metaheuristic Search, Cuckoo Search, Multilevel Compression, Dictionary Based String Compression , Link Search, Knutt Morris Pratt*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Bismillahirrohmaanirrohiim. Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa. Atas rahmat dan kasih sayangNya, penulis dapat menyelesaikan tesis dalam bentuk buku ini yang berjudul Kompresi Multilevel Pada Meta Heuristik *Focused Web Crawler*.

Pengerjaan buku ini penulis tujukan untuk mengeksplorasi lebih mendalam topik topik yang tidak diwadahi oleh kampus, namun banyak menarik perhatian penulis. Selain itu besar harapan penulis bahwa pengerjaan tugas akhir sekaligus pengerjaan buku ini dapat menjadi batu loncatan penulis dalam menimba ilmu yang bermanfaat.

Penulis ingin menyampaikan rasa terima kasih kepada banyak pihak yang telah membimbing, menemani dan membantu penulis selama masa pengerjaan tesis maupun masa studi.

1. Allah SWT yang selalu memberi kebahagiaan dan makna pada hidupku.
2. Ibu dan Ayah yang selalu mendukung dalam segala hal.
3. Bapak Dr.Ir. Raden Venantius Hari Ginardi, M.Sc, selaku pembimbing penulis yang telah memberikan didikan, pengajaran, dan nasihat yang telah diberikan oleh beliau semasa pengerjaan tugas akhir.
4. Rekan-rekan satu angkatan 2016 mahasiswa magister Teknik Informatika yang tidak lelah membantu penulis semasa masa studi.
5. Penulis menyadari bahwa buku ini jauh dari kata sempurna. Maka dari itu, penulis memohon maaf apabila terdapat salah kata maupun makna pada buku ini. Akhir kata, penulis mempersembahkan buku ini sebagai wujud nyata kontribusi penulis dalam ilmu pengetahuan.

Surabaya, Januari 2019

Dian Septiani Santoso

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

Pada Bab ini akan dijelaskan mengenai beberapa hal dasar dalam pembuatan proposal penelitian yang meliputi latar belakang, perumusan masalah, tujuan, manfaat, kontribusi penelitian, dan batasan masalah.

1.1. Latar Belakang

Saat ini pencarian link website dengan memperhatikan isi dari dokumen website sesuai dengan pencarian kata atau dikenal dengan istilah *Focused Web Crawler* sudah mulai banyak di lakukan oleh para peneliti. Beberapa penelitian yang dilakukan antara lain adalah dengan mencari kesesuaian dari website dengan kata yang dicari oleh pengguna. Dalam penelitian yang telah dilakukan, untuk mencari kesesuian isi website dapat dilakukan melalui pencocokan pencarian pengguna dengan link web yang ada(Kan, 2005)(Dwivedi and Arya, 2017). Pencarian kesesuaian dengan menggunakan link dalam *focused web crawling* masih memiliki kekurangan yaitu dalam keterkaitan isi dari website karena tidak semua link merepresentasikan isi dari website(Pant and Srinivasan, 2006).

Untuk mengatasi tidak tervisitnya link yang tidak sesuai dengan inputan pengguna tetapi memiliki konten yang sesuai dengan pencarian penggunaan alogaritma genetika untuk pencarian link yang memiliki kemungkinan kesesuaian tinggi dilakukan oleh Banu Wirawan Yohanes dkk (Yohanes, Handoko and Wardana, 2011) penelitian ini membuktikan pencarian link dengan metode yang diusulkan lebih meminimalkan link yang tidak tervisit daripada metode BFS. Selain melakukan pencarian dari link yang ada cara lain untuk melakukan optimasi pencarian link yang memiliki keterkaitan adalah dengan melakukan pencarian isi dari konten website. Dalam pencarian konten web yang dilakukan adalah perhitungan relevansi percobaan Bireshwar Ganguly dkk (Ganguly and Raich, 2014) dari percobaan relevansi yang didapat cukup baik sehingga mengurangi kesalahan pencarian link terkait , untuk menambah keakuratan dari konten website penelitian dengan menggunakan relevansi topic diajukan oleh ZHAO Wei dkk (Wei *et al.*, 2016) dari penelitian ini didapatkan hasil pencarian link yang memiliki kualitas yang sesuai secara baik namun kekurangan yang ada adalah banyaknya waktu yang dibutuhkan untuk melakukan penilaian kerelavanan dari website yang dikunjungi.

Untuk mengatasi kekurangan dari lamanya proses pencarian seperti penelitian sebelumnya, Joy Dewanjee (Dewanjee, 2016) mencoba pendekatan baru menggunakan metode pencarian secara heuristic dengan mengetahui apa yang dicari, dari mana pencarian dilakukan dan data mana yang tidak dibutuhkan dengan menggunakan metode optimasi pencarian "Cuckoo Search". Dan setelah pencarian berhasil dilakukan ditambahkan teknik pengenalan pattern pada web yang sudah disimpan menjadi ekstrak web dengan pencocokan string dengan menggunakan alogaritma Knutt Morris Pratt untuk normalisasi keluaran link yang sebelumnya terekstrak pada penyimpanan alogaritma pencarian.

Namun yang menjadi kelemahan adalah metode ini membutuhkan space yang lebih banyak karena harus menyimpan link pencarian yang sudah dilakukan sebelumnya untuk menjadi acuan pencarian selanjutnya(*history* sistem), untuk mengatasi hal ini penulis melakukan setting overhead limit namun belum begitu jelas apakah hasil informasi yang diperoleh sudah maksimal atau belum.Untuk mengatasi permasalahan space penyimpanan salah satu yang bisa dilakukan adalah menggunakan teknik kompresi data. Dalam teknik kompresi data yang dipilih adalah teknik kompresi yang dapat menjamin bahwa hasil pencarian tetap memiliki nilai kebenaran yang mendekati pencarian sebelum terkompresi. Selain itu teknik kompresi juga harus memiliki rasio yang cukup baik untuk penghematan space.

.

1.2. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah meliputi hal sebagai berikut:

1. Bagaimana cara melakukan kompresi *history* pencarian website pada *focused web crawler* metaheuristik *cuckoo search*.
2. Berapa rasio rata-rata penyimpanan dengan menggunakan metode kompresi yang diusulkan.
3. Bagaimana membuktikan bahwa metode yang diusulkan dikatakan tidak mengurangi nilai kebenaran hasil pencarian daripada metode sebelumnya.

1.3. Tujuan

Tujuan yang akan dicapai dalam pembuatan tesis ini adalah mengurai penyimpanan data didalam memori sistem , tanpa mengurangi kebenaran hasil link yang dicari oleh pengguna pada pencarian *focused web crawler* dengan *cuckoo search* .

1.4. Manfaat

Manfaat dari penelitian ini adalah mengurangi ukuran penyimpanan file *history* pencarian pada disk pada *focused web crawler* dengan *cuckoo search*.

1.5. Kontribusi Penelitian

Kontribusi penelitian ini adalah mengurangi ukuran file penyimpanan *history* pencarian website dengan cara melakukan teknik kompresi multilevel dengan melakukan pengurangan kata yang tidak penting dan kompresi string menggunakan kamus kata sehingga mendapatkan file dengan ukuran lebih kecil tanpa mengurangi kebenaran link website yang dicari.

1.6. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data yang digunakan untuk dilakukan metode adalah kumpulan data link website yang didapat dari crawling pada beberapa situs berita https://www.detik.com/,https://www.tribunnews.com,https://www.cnnindonesia.com/,https://www.viva.co.id/,http://www.liputan6.com/,http://republika.co.id.
2. Fokus metode yang diajukan adalah teknik untuk mengompres data *history* penyimpanan link website yang memiliki keterkaitan dengan pencarian pengguna.
3. Data untuk percobaan kompresi dan pencarian pada sistem adalah link dan name tag hasil crawl pada 6000 link dengan banyak link dan name tag yang berbeda-beda sebanyak 124.881 link dengan ukuran file 16.8 Mb dari sumber website berita yang telah di pilih.
4. Pencocokan dan pencarian kata inputan dilakukan pada data telah ditentukan dan tersimpan di dalam sistem.

5. Pencocokan terhadap pencarian kata oleh pengguna hanya merupakan string matching dengan algoritma Knutt Morris Pratt (KMP), belum sampai konteks dari kata inputan.
6. Tidak memperhitungkan waktu.

BAB 2
KAJIAN PUSTAKA

Pada bab ini akan dijelaskan tentang pustaka yang terkait dengan landasan penelitian. Pustaka yang terkait dianalisa dan disajikan pada tiap-tiap sub bab.

2.1. *Focused Web Crawler*

Focused Web Crawler merupakan metode pencarian website yang sesuai dengan pencarian yang diingikan oleh user. *Focused Web Crawler* di kembangkan agar pencarian di dalam data web yang sangat besar dapat dilakukan dengan efisien baik waktu dan memori pada aplikasi ataupun database untuk hasil pencarian.

Beberapa strategi yang dipakai dalam *Focused Web Crawler* (Avraam, 2011) antara lain adalah strategi prioritas dimana dalam strategi ini website yang memiliki banyak relevansi akan diprioritaskan untuk dilakukan pengambilan informasinya. Strategi kedua adalah strategi pembelajaran, dalam strategi ini *Focused Web Crawler* dilakukan dengan dua cara yaitu pembelajaran saat crawler offline atau reinforcement/ penambahan pembelajaran data crawler saat online, pembelajaran ini dengan mengenalkan mesin pencari dengan sebuah pengetahuan agar pencarian dapat dilakukan dengan efisien. Strategi selanjutnya adalah strategi evaluasi dengan alat pengklasifikasi, strategi ini melakukan pencarian website berdasarkan nilai hasil klasifikasi dari pencarian yang berada dalam klasifikasi yang sama atau termasuk dalam rentang nilai yang telah ditentukan. Strategi yang lain adalah strategi training data fitur pada web page dimana ada dua macam tipe data fitur yakni data textual dan data non textual, dengan adanya training pada data fitur ini akan mempermudah proses pembelajaran mesin pencari.

Ada dua tipe pencarian *Focused Web Crawler* terhadap input yang diberikan oleh pengguna yakni pencarian berdasarkan link URL yang ada di dalam web page dan dari konten di dalam web page. Walaupun sudah ada beberapa strategi dalam *Focused Web Crawler*, dalam perjalanannya masih saja ada kelemahan dalam mendapatkan hasil pencarian yang memuaskan. Misalnya pada pencarian dengan pencocokan kata kunci terhadap link URL masih memberikan peluang bahwa link URL yang tidak merepresentasikan kata kunci akan tidak di kunjungi walaupun kontennya mengandung kata kunci yang dicari oleh user, oleh karena itu gagasan pencarian berdasarkan konten web page dilakukan dan mendapatkan hasil bahwa web page yang memiliki konten kata

kunci bisa terkunjungi (Wei *et al.*, 2016). Karena harus melakukan pencarian kata yang sesuai pada setiap website dan web page waktu yang diperlukan lebih lama, sedangkan untuk pencarian dengan menggunakan link URL memiliki waktu pencarian yang lebih cepat.

Beberapa penelitian juga dilakukan untuk memaksimalkan pencarian dengan menggunakan link yakni dengan penggunaan metode alogaritma genetik (Yohanes, Handoko and Wardana, 2011) untuk mengurangi hilangnya informasi dari link website yang memiliki relevansi namun berada pada link yang tidak berkaitan dengan *frontier* atau tidak berhubungan dengan link yang ditemukan. Cara yang dilakukan dengan melakukan pemilihan gen unggul dari hasil kawin silang link yang ada. Pada penggunaan alogaritma genetika keberhasilan dalam penanganan masalah kemungkinan link yang tidak tervisit menghasilkan dampak yang cukup baik. Namun proses yang dikerjakan cukup banyak perhitungan dan proses dari pemilihan individu dalam populasi , *croos over* antar gen tiap individu , memilih individu hasil proses persilangan untuk dijadikan individu unggul sesuai kebutuhan.

Dari masalah ini kemudian terpikirkan untuk memilih alternatif metode optimasi lainnya. Salah satu metode optimasi yang banyak digunakan karena memiliki hasil optimasi pencarian yang tidak kalah dan algoritma yang tidak begitu memerlukan banyak perhitungan adalah pencarian menggunakan metode pencarian cuckoo. Pencarian ini lebih menekankan pada pencarian dari data *history* yang mirip sebagai kamus pencarian. *Focused Web Crawler* yang memanfaatkan metode pendekatan metaheuristik dengan metode pencarian cuckoo(Dewanjee, 2016) untuk mengurangi waktu pencarian dan menambah keakuratan pencarian berdasarkan link website.

2.2. Pendekatan Metaheuristik

Metaheuristik merupakan metode tingkat lanjut dari metode heuristik, dimana metode heuristik sendiri adalah sebuah teknik yang digunakan untuk menyelesaikan masalah hingga ditemukan sebuah solusi (Wikipedia, 2018). Menurut Doddy dharma dkk (S, 2015) metode heuristik diartikan juga sebagai suatu kaidah yang merupakan metoda/prosedur yang berdasarkan kepada pengalaman dan praktek, syarat, trik atau bantuan lainnya yang membantu mempersempit dan memfokuskan proses pelacakan kepada suatu tujuan tertentu. Menurut Wikipedia metaheuristik merupakan metode

heuristik yang dibuat untuk menemukan, menghasilkan, atau memilih metode heuristik yang dapat memberikan solusi yang cukup baik untuk masalah optimasi, terutama dengan informasi permasalahan yang tidak lengkap atau tidak sempurna, atau kapasitas komputasi yang terbatas (Local *et al.*, 2018).

Cuckoo search merupakan sebuah alogaritma pencarian yang diciptakan oleh Xin-She Yang dan Suash Deb untuk mengatasi masalah optimasi. Alogaritma ini terinspirasi oleh parasit yang dilakukan spesies burung cuckoo dimana cuckoo menitipkan telur yang dimiliki pada sarang burung lain(Yang, Deb and Behaviour, 2009). Dalam cuckoo search ada tiga aturan ideal (Yang *et al.*, 2018) yaitu :

1. Setiap cuckoo meletakkan satu telur pada satu waktu dan meletakkannya pada sarang acak.
2. Sarang terbaik dengan kualitas telur terbaik akan terbawa pada generasi berikutnya (akan hidup).
3. Jumlah sarang host tetap, dan telur yang dititipkan akan diketahui induk pada sarang yang diinangi memiliki probabilitas sebesar 0 dan 1. Dari sini akan ditemukan sarang mana yang merupakan sarang terburuk dan tidak akan digunakan acuan untuk dilakukan penitipan sarang untuk telur selanjutnya.

Untuk lebih jelasnya berikut adalah alogaritma pencarian cuckoo search:

Algoritma Cuckoo Search (Yang dan Deb, 2009).

1: Fungsi Objektif f(x) , x = (x1 , … , xd)T ;
2: Inisialisasi populasi dari n sarang burung target Xi (i = 1 , 2 , …, n);
3: **While** (t < generasiTotal) atau (kriteria lain untuk berhenti);
4: Evaluasi nilai kualitas dari masing–masing burung cuckoo
5: Pilih dari burung cuckoo secara acak dan lakukan random walk
6: **If** (Fi> Fj)
7: Gantikan burung cuckoo j dengan burung cuckoo i
8: **End If**
9: Reset ulang sarang-sarang dengan kondisi terburuk (Pa)
10: Simpan sarang-sarang yang berhasil lolos

11: Urutkan solusi dan cari yang terbaik

12: **End While**

2.3. Kompresi Data

Kompresi data adalah sebuah teknik untuk memadatkan data sehingga hanya membutuhkan ruang yang kecil untuk penyimpanan sebuah data. Menurut Wol, J Gerard mengusulkan sebuah konsep kompresi yaitu SP (*Simplicity* and *Power*) dimana tujuan kompresi adalah melakukan penghilangan informasi yang berulang(*redundan*) sebagai tahap *simplicity* sehingga dapat mengurangi ruang penyimpanan data dan kompleksitas progam sebagai tahap *power* (Wol, 1999). Kompresi dilakukan untuk mengurangi ukuran data, data yang dilakukan proses kompresi antara lain adalah data gambar, data audio,data video dan data text. Pada setiap jenis data memiliki teknik yang berbeda pada masing-masing jenis data(Salomon, 2004).

Untuk data gambar beberapa teknik kompresi juga berbeda-beda sesuai jenis gambar, beberapa jenis kompresi antara lain metode pendekatan intuitif dengan melakukan subsampling dan kuantisasi pada pixel gambar, transformasi gambar dengan mengurangi pixel yang redundan,transformasi orthogonal, *Discrete Cosine Transform*, dan masih banyak lagi.

Untuk data berupa audio beberapa tenik kompresi yang ada antara lain *μ-Law and A-Law Companding,* ADPCM (*Audio Differential Pulse Code Modulation*), dan *Speech Compression*. Teknik kompresi pada data video antara lain adalah MPEG (*Moving Pictures Experts Group*). Dalam tenik ini dibedakan beberapa jenis MPEG sesuai dengan masing-masing bit rate video MPEG-1 digunakan bit rate minimum 15 Mbit/s, MPEG-2 pada bit rate 10 Mbit/s, MEPG-3 untuk HDTV dan MPEG-4 untuk bit rate dibawah 16Kbit/s.

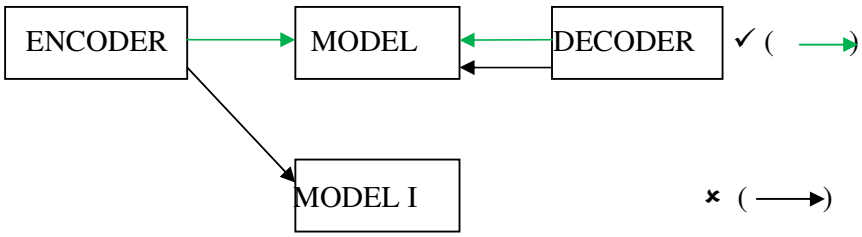
Pada data kompresi memiliki dua jenis hasil keluaran yakni kompresi *irreversible* dan kompresi *reversible* (Bell, Witten and Cleary, 1989).

a. Kompresi *irreversible* adalah kompresi yang tidak dapat mengembalikan data hasil kompresi sama seperti data sebelum kompresi. Kompresi *irreversible* bisa disebut juga kompresi *lossy* karena data kompresi yang dihasilkan dengan melakukan penghilangan data yang tidak berguna , data yang *redundan*, dan data yang berupa *noise*. Kompresi ini cocok digunakan untuk digitasi sinyal analog

seperti data audio dan gambar. Pada kasus tertentu ada data text yang menggunakan kompresi *irreversible,* kompresi dilakukan misalkan dengan menghilangkan spasi yang kosong dengan satu spasi saja. Karena ini akan merubah data text maka metode kompresi text *irreversible* tidak disarankan untuk data text normal tanpa kondisi yang diketahui oleh user.

b. Kompresi *reversible* adalah kompresi yang dapat mengembalikan data setelah kompresi sama persis dengan data sebelum kompresi. Kompresi jenis *reversible* atau bisa disebut juga kompresi lossless ini terjadi karena dalam kompresi reversible ini tidak menghilangkan data apapun yang ada di dalam data asli. Kompresi jenis ini sangat cocok digunakan untuk data berupa text karena di dalam data text penghilangan data akan mempengaruhi arti atau nilai data text.

Menurut T.Bell dkk (Bell, Witten and Cleary, 1989),secara umum metode dalam melakukan kompresi text dibedakan menjadi dua yaitu dengan cara kode statistik dan kamus. Metode kompresi statistik adalah setiap simbol diberi kode berdasarkan pada probabilitas bahwa symbol ada. Ketika sebuah simbol memiliki frekuensi kemunculan yang paling banyak, simbol tersebut akan mendapatkan kode pendek dan sebaliknya. Sedangkan pada metode kompresi kamus adalah mengelompokkan kata atau frasa yang ada dengan kode tertentu. Pada kompresi data dibagi menjadi dua bagian yaitu pembuat encode(*encoder)* yang benar-benar menghasilkan bitstream terkompresi dan pemodel(*modeler*) yang memberi informasi ke dalamnya. Dua bagian ini disebut *coding* dan *modeling*, modeling memberikan probabilitas ke simbol, dan coding menerjemahkan probabilitas ini ke suatu urutan bit. Prinsip kerja kompresi adalah decoder dan encoder, untuk membuat hasil kompresi berfungsi secara akurat maka model yang diakses antara decoder maupun encoder adalah model yang sama.



Gambar 2. 1 Pengaksesan Model untuk Kompresi Data

Untuk menilai apakah performa dari kompresi efektif, beberapa cara yang bisa dilakukan adalah (Salomon, 2004):

1. Dengan melakukan perhitungan rasio. Dimana rasio didefinisikan pada persamaan berikut:

$$rasio\ kompresi = \frac{Ukuran\ data\ setelah\ komprei}{Ukuran\ data\ sebelum\ kompresi} \quad (1)$$

   Dengan perhitungan diatas jika nilai yang dikeluarkan 0.5 memiliki arti bahwa data keluaran menempati 50% dari ukuran aslinya sebelum kompresi. Nilai kompresi bernilai negatif jika rasio kompresi memiliki nilai lebih dari 1 atau memiliki lebih dari 100% dari ukuran aslinya.

2. Invers dari rasio atau dikenal dengan sebutan faktor kompresi. Cara perhitungan faktor kompresi adalah kebalikan dari perhitungan rasio. Persamaan faktor kompresi adalah sebagai berikut:

$$faktor\ kompresi = \frac{Ukuran\ data\ sebelum\ komprei}{Ukuran\ data\ setelah\ kompresi} \quad (2)$$

   Kompresi data dikatakan baik atau efektif jika nilai faktor kompresi memiliki nilai positif, atau semakin banyak nilai faktor kompresinya maka kompresi data semakin efektif.

3. Penggunaan 100 × (1- nilai rasio kompresi) juga merupakan ukuran yang wajar untuk kinerja kompresi. Nilai 50 berarti aliran keluaran menempati 50% dari outputnya ukuran asli (atau kompresi telah menghasilkan penghematan 50%).

Pada penjelasan sebelumnya, telah dijelaskan bahwa pada data text kompresi memiliki dua jenis(Lelewer and Hirschberg, 2004), kompresi *irreversible* atau *lossy* (tidak dapat dikembalikan lagi) bisa digunakan dengan kondisi tertentu, namun tidak disarankan karena kemungkinan data hilang/tidak sama dengan data inputan. Teknik yang disarankan untuk kompresi data text adalah dengan kompresi *reversible* atau *lossless*. Dari studi literature, beberapa teknik kompresi data text (Kodabagi, 2015)(Kalajdzic, Ali and Patel, 2015)(Mahmood and Hasan, 2017) mencoba untuk mengurangi kapasitas penyimpanan, pada saat data berada di dalam aplikasi tanpa menghilangkan informasi apapun dari data yang ada (kompresi ***lossless***). Sehingga

kompresi data diharapkan mampu untuk meminimalkan pemakaian memori baik memori penyimpanan maupun memori pengaksesan.

Pada penelitian pertama oleh M. M. Kodabagi dkk (Kodabagi, 2015) kompresi yang dilakukan adalah dengan melakukan pengamanan menggunakan metode bit stuffing dan kompresi data menggunakan metode Huffman. Kompresi yang digunakan merupakan jenis kompresi statistik dimana dalam kompresi Huffman menggunakan jumlah frekuensi kemunculan simbol. Dimana dalam pembuatan model, simbol dengan kemungkinan kemunculan sedikit akan memiliki kode yang banyak dan sebaliknya. Dari hasil kompresi dapat mengurangi space sebanyak 45.41% dari data yang masuk awal. Percobaan ini dilakukan pada data text yang berada didalam jaringan di internet agar lebih aman dan membutuhkan memori yang tidak banyak.

Teknik kompresi kedua adalah yang dilakukan oleh Kenan Kalajdzic dkk (Kalajdzic, Ali and Patel, 2015) dengan memperkenalkan sebuah teknik untuk mengompresi data text volume kecil agar menghemat pemakaian biaya dan memori pengguna yaitu alogaritma b64pack. Teknik kompresi kedua merupakan jenis kompresi kamus. Cara yang dilakukan adalah melakukan transcoding (mengubah karakter yang ada dalam pesan dengan tabel/kamus yang sudah ditentukan oleh penulis) yang kemudian melakukan kompresi dengan merubah 8 bit simbol dengan 7 bit simbol pada kamus yang sudah dibuat sebelumnya. Hasil kompresi mampu mengurangi space sebanyak 21% dari data sebelumnya. Data yang digunakan teknik ini adalah data yang memiliki volume kecil yang berhubungan dengan internet seperti data berbasis pesan (twitter chat dll).

Ashiq Mahmood dkk (Mahmood and Hasan, 2017) mencoba kompresi dengan mengenalkan teknik 6bit encoding. Kompresi ini merupakan jenis kompresi kamus dimana untuk melakukan kompresi dilakukan dengan merubah text dengan nilai desimal sesuai tabel encoding 6BE yang sudah dibuat sebelumnya . Tabel ini menggunakan 28 macam nilai untuk semua karakter yang muncul di dalam keyboard,untuk selanjutnya merubah text yang berisi nilai desimal sesuai dengan nilai binary pada tabel 6BE yang ada. Dalam proses ini nilai binary yang dihasilkan adalah sebanyak 6 bit pada masing-masing karakter. Teknik kompresi ini mampu menghemat penyimpanan data 25% dari data asli.

Dari ketiga penelitian memiliki kelebihan dan kekurangan masing-masing, namun yang dapat disimpulkan penelitian yang telah dilakukan merupakan pendekatan baru untuk kompresi data text dengan cara melakukan encoding dan decoding bit pada data text. Untuk mendapatkan rasio kompresi yang cukup tinggi penelitian M. M. Kodabagi dkk (Kodabagi, 2015) dapat diimplementasikan karena mereka menggunakan kompresi multilevel dengan menggabungkan 2 buah metode yakni bit stuffing dan Huffman. Untuk memperoleh kecepatan kompresi data pada penelitian Kenan Kalajdzic dkk (Kalajdzic, Ali and Patel, 2015) mengenalkan waktu untuk melakukan proses dimulainya proses encoding sehingga waktu untuk melakukan kompresi bisa dilakukan saat input data sedang dimasukkan. Dan untuk penelitian Ashiq Mahmood dkk (Mahmood and Hasan, 2017) menemukan bahwa pada jenis data tertentu dapat mempengaruhi rasio kompresi dan kecepatan dalam kompresi sebuah data text walaupun cara kompresi menggunakan metode yang sama.

Dari ketiga penelitian teknik kompresi diatas kompresi selalu mengubah simbol/karakter menjadi bit yang lebih kecil daripada sebelumnya,dan untuk penulisan didalam penyimpanan setiap bit tersebut akan ditulis kedalam 8 bit / 1 byte sehingga akan dalam posisi terencode di dalam sistem. Karena pada penelitian ini tahap selanjutnya pencarian yang digunakan adalah pencarian string per byte dengan metode pencocokan string KMP maka diperlukan teknik kompresi yang menyimpan hasil kompresi per byte yang memiliki nilai statis. Nilai statis diperlukan karena setelah index pencarian ditemukan maka hasil bisa sesuai dengan model yang dibuat dan mengembalikan informasi secara tepat. Untuk itu dicari pendekatan metode lain yang memungkinkan untuk membuat sebuah teknik kompresi yang dapat mengurangi ukuran data dan mampu dilakukan pencarian dengan algoritma pencarian string KMP.

Shunsuke Kanda dkk (Kanda, Morita and Fuketa, 2017) melakukan pembuatan kompresi dengan pendekatan kamus kata yang dibuat berdasarkan jenis kata (tambahan, akhiran, kata dasar) . Kamus kata pertama yang dibuat berupa imbuhan kata setelah itu barulah dilakukan pembuatan kamus kata yang bukan kata awalan, setelah itu barulah dilakukan penggabungan 2 kata yang telah ada pada masing-masing kamus kata dan akan ditulis sesuai dengan urutan permodelan yang telah dibuat sebelumnya. Dari hasil kompresi ini teknik kompresi mampu menghemat rata-rata sampai dengan 28% dari data awal. Penelitian ini lebih membahas tentang efisiensi urutan index kata pada

pembuatan kamus kata. Penelitian ini bisa dijadikan acuan pembuatan kamus kata yang menggunakan jenis kata (awalan,akhiran,kata dasar) bukan per karakter atau per huruf. Selain itu untuk penyimpanan data kompresi di dalam sistem dilakukan dengan menulis index kedalam 2 byte data secara terus menerus.

2.4. Knutt Morris Pratt (KMP)

Knutt Morris Pratt merupakan salah satu alogaritma pencarian string yang dikembangkan oleh Donald Knuth , Vaughan Pratt, dan James H. Morris yang di perkenalkan pada tahun 1977. Persoalan dalam pencarian string adalah sebuah teks dengan panjang n, dan pattern yaitu sebuah string dengan panjang m karakter dimana (m<n) yang akan dicari di dalam teks. Alur yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string adalah sebagai berikut(Pratt, 2012):

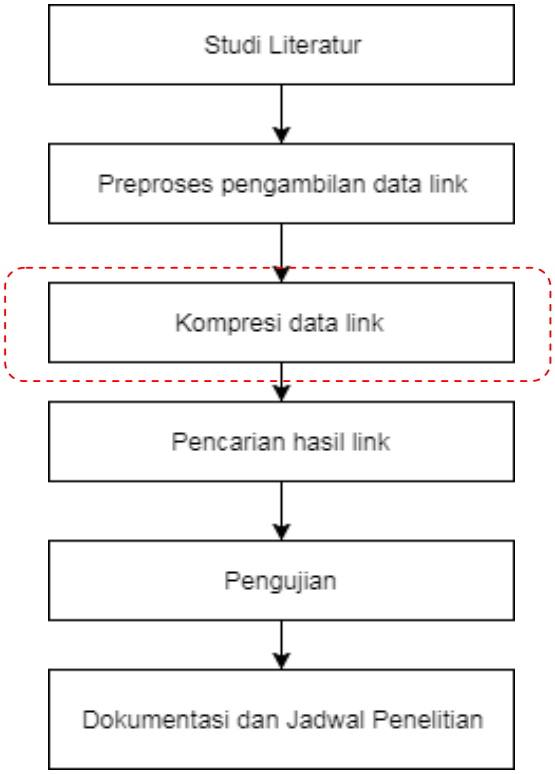
1. Pencocokan pattern dilakukan pada awal teks.
2. Algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di dalam teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi, pencocokan di lakukan dari kiri ke kanan:
    a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
    b. Semua karakter di pattern yang cocok. Kemudian algoritma akan memberitahukan penemuan di posisi mana pattern ditemukan.
3. Algoritma kemudian menggeser pattern berdasarkan tabel next, lalu mengulangi langkah 2 sampai pattern berada di akhir teks.

Dasar Ideologi dibalik algoritma KMP adalah: kapan pun saat terjadi ketidakcocokan (setelah beberapa pencocokan), alogaritma sudah mengetahui berapa karakter dalam teks. Misalkan pada pencarin kata “formasi” dari teks “info inform informasikan”. Langkah awal yang dilakukan adalah pencocokan pattern “formasi” dengan teks dari kiri ke kanan. Yang dicocokan pertama adalah huruf awal pattern dengan huruf awal pada teks, karena huruf awal pattern dimulai f dan huruf awal teks dimulai dengan i maka huruf pertama pada teks dilewati dan dilanjutkan pada huruf kedua dst sampai ditemukan pattern yang cocok dan mencatat indeks keberapa pattern ditemukan(Dewanjee, 2016).

[Halaman ini sengaja dikosongkan]

BAB 3
METODOLOGI PENELITIAN

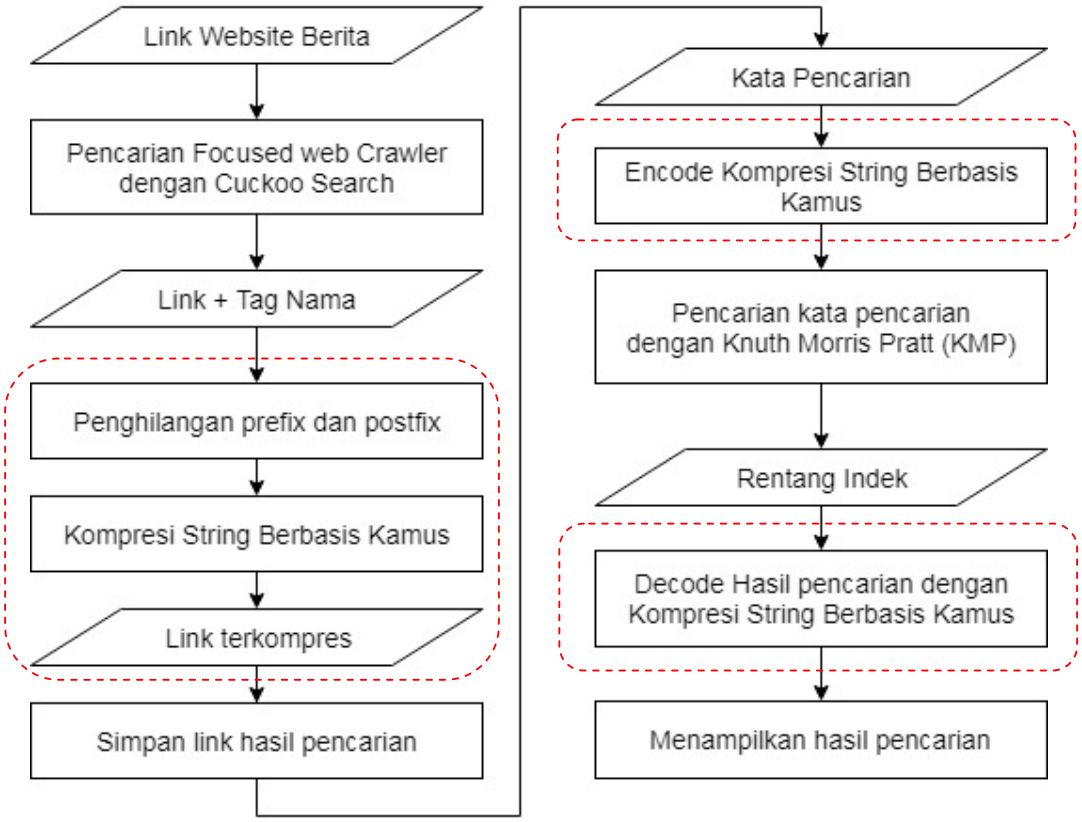
Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini, yang terdiri dari (1) studi literatur, (2) preproses pengambilan data link,(3) kompresi data link, (4) pencarian hasil link, (5) pengujian, dan (6) dokumentasi dan pembuatan laporan. Ilustrasi alur metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Metodologi Penelitian

Keterangan : ------- Kontribusi Penelitian

Untuk desain sistem akan di jelaskan pada gambar 3.2



Gambar 3. 2 Desain sistem

Pada gambar desain sistem menunjukkan *input*, pemrosesan dan *output* dari penelitian yang dilakukan. Karena penelitian ini merupakan pengembangan dari penelitian sebelumnya untuk kontribusi yang dilakukan di tandai dengan tanda ---- .

3.1. Studi Literatur

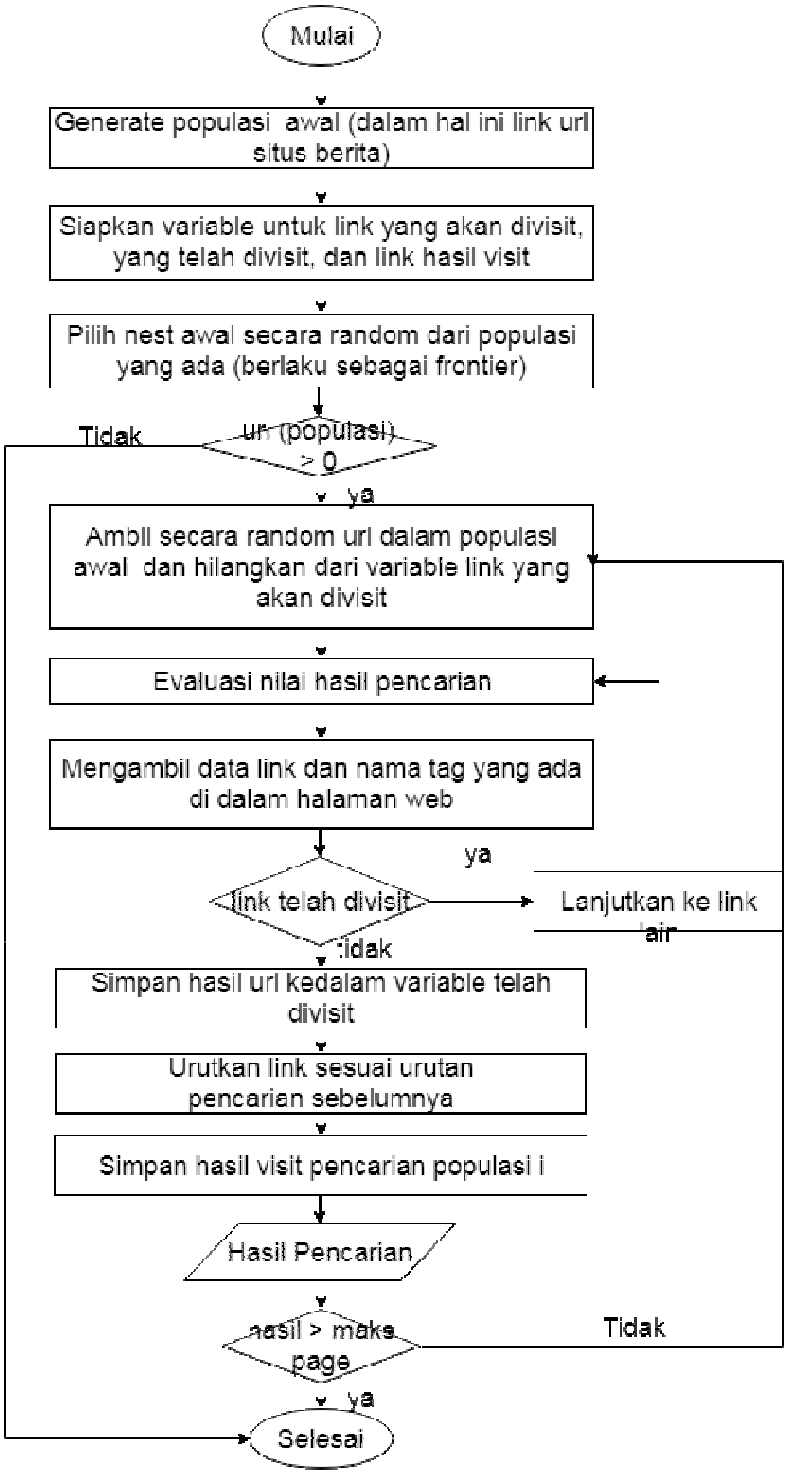
Tahap studi literatur bertujuan untuk mengumpulkan referensi - referensi yang dapat menunjang penelitian. Sumber referensi dapat berupa jurnal ilmiah, buku teks atau konferensi. Referensi yang dikumpulkan berhubungan dengan metode pencarian pada *Focused Web Crawler* serta kompresi pada data text. Referensi tersebut digunakan untuk merumuskan permasalahan yang menjadi landasan dilakukannya penelitian ini dan solusi yang akan diusulkan. Berdasarkan studi literatur yang telah dilakukan, informasi yang berkaitan dengan penilitian yang dilakukan ini, seperti berikut :

1. Pada metode *Focused Web Crawler*, pencarian informasi sebuah web sesuai dengan input yang dimasukkan oleh user .
2. Untuk mendapatkan halaman web yang sesuai dengan pencarian oleh user dalam metode *Focused Web Crawler* memiliki waktu yang cukup lama karena harus melakukan pengecekan minimal kesamaan kata dalam sebuah halaman web, pada tingkat yang lebih dalam adalah pencarian topik dalam sebuah halaman yang sesuai dengan inputan user sehingga memerlukan tambahan waktu untuk melakukan proses ini .
3. Untuk mempersingkat waktu proses pencarian web yang sesuai dengan inputan dilakukan metode pendekatan metaheuristik untuk menyimpan informasi yang telah dilakukan sebelumya sehingga waktu pencarian bisa dipersingkat tetapi masih memiliki relevansi terhadap inputan yang diberikan oleh user.
4. Penyimpanan data masih terjadi kelemahan yaitu jumlah data yang disimpan cukup banyak.
5. Solusi yang bisa diambil agar data yang disimpan tidak banyak dan informasi data tidak hilang adalah dengan melakukan kompresi data pada penyimpanan.

3.2. Preproses Pengambilan Data Link

Pada tahap ini proses yang dilakukan adalah preproses untuk mendapatkan data link hasil pencarian dengan web crawler. Tahapan ini merupakan hasil studi literatur tentang cara melakukan proses pencarian link pada website dengan metode *Focused Web Crawler* dengan menggunakan *cuckoo search*. Data hasil pencarian berupa link html dengan tag nama link yang akan digunakan sebagai data *history* pencarian web. Pengambilan data link menggunakan bahasa pemrograman Java versi 8.

Pustaka - pustaka pendukung yang akan digunakan dalam implementasi metode adalah jsoup. Langkah yang dilakukan adalah pencarian link website dengan menggunakan pencarian metaheuristik cuckoo search. Alurnya seperti dibawah ini :



Gambar 3. 3 Pencarian *Focused Web Crawler* dengan *Cuckoo Search*

Generate populasi awal dalam hal ini menyediakan situs dari link berita yang telah ditentukan sebelumnya seperti https://www.detik.com/ , https://www.tribunnews.com, https://www.cnnindonesia.com/, https://www.viva.co.id/, dan http://www.liputan6.com/.

Data yang dihasilkan dari pencarian ini merupakan data link website dan tag nama link. Data itu akan digunakan sebagai *history* pencarian pada pencarian metaheuristik. Contoh data yang akan disimpan di dalam *history* pencarian :

Tabel 3. 1 Tabel *history* pencarian

Tag Nama	Alamat url
Liputan Khusus	https://www.liputan6.com/news/liputankhusus
Zona MPR RI	https://www.liputan6.com/news/zona-mpr-ri
Cek Fakta	https://www.liputan6.com/news/cek-fakta
Divonis 15 Tahun Penjara, Setya Novanto Tak Ajukan Banding	https://www.liputan6.com/news/read/3496583/divonis-15-tahun-penjara-setya-novanto-tak-ajukan-banding
Internasional	https://news.detik.com/internasional
Mensos Idrus Marham buka suara soal stiker cagub-cawagub Jatim yang diselipkan dalam penyaluran program PKH di Lamongan. Apa penjelasan Mensos?	https://news.detik.com/berita-jawa-timur/d-4000690/penjelasan-mensos-soal-stiker-cagub-jatim-di-program-pkh
100 Lebih WN China Ditangkap di Bali Terkait Kejahatan Siber	https://www.liputan6.com/news/read/3496570/100-lebih-wn-china-ditangkap-di-bali-terkait-kejahatan-siber

Setelah diambil sesuai format yang diinginkan maka selanjutnya yang dilakukan adalah menulis data tersebut pada file text dan disimpan didalam aplikasi. Jadi di dalam file text merupakan kumpulan link url hasil pencarian menggunakan pencarian cuckoo search.

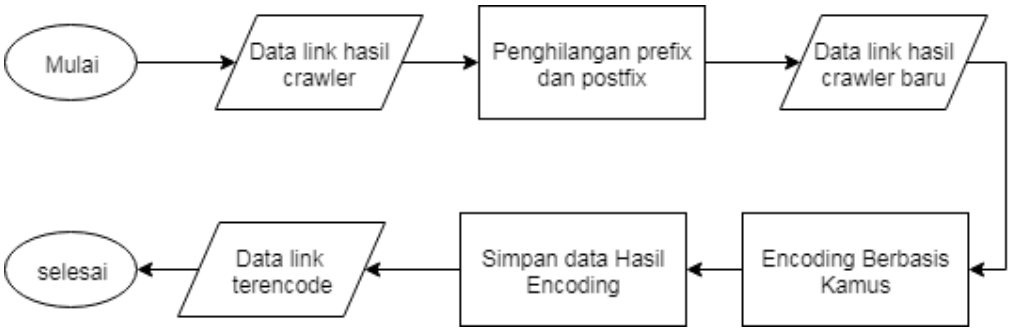
3.3. Kompresi Data Link

Proses selanjutnya adalah kompresi data link, yang dimaksud disini adalah data hasil pencarian dengan *focused web crawler* pada tahap sebelumnya. Untuk metode

kompresi yang di pakai dengan menggunakan proses penghilangan kata (prefix dan postfix) dan selanjutnya di lakukan kompresi string berbasis kamus , metode ini dipilih karena hasil *encode* dari setiap kata yang dikamuskan disimpan dalam byte yang tepat satu kata atau huruf sehingga tidak mempersulit dalam pencarian / pencocokan byte menggunakan metode KMP.

Rasio merupakan perbandingan antara besar data hasil kompresi dengan data asli sebelum kompresi. Hasil ini berlaku untuk data bertipe distinct yaitu data yang jarang muncul pada text , tipe data distinct lebih banyak terdiri dari angka dan karakter daripada huruf. Karena data pada link web merupakan data distinct maka dengan metode kompresi kamus diharapkan dapat hasil yang baik.

. Alur kompresi data dapat dilihat dari gambar 3.3 dibawah ini :



Gambar 3. 4 Alur Kompresi Data Link

Selanjutnya adalah langkah yang dilakukan dalam kompresi dari awal sampai akhir adalah sebagai berikut:

3.3.1 Persiapan data link

Langkah pertama yang dilakukan untuk kompresi data adalah mempersiapkan data. Data yang akan di lakukan kompresi adalah data *history* link web yang sebelumnya telah disimpan di dalam aplikasi. Kumpulan data link tersebut berisi string yang nantinya akan dilakukan proses kompresi.

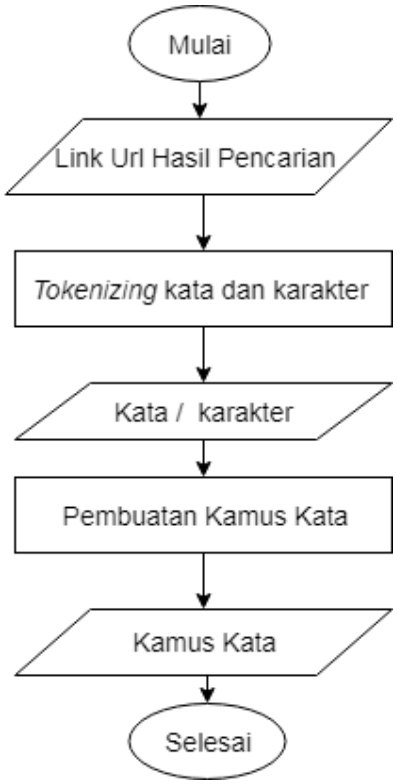
3.3.2 Penghilangan *prefix* dan *postfix*

Selanjutnya adalah proses penghilangan kata pada awalan (prefix), pada proses penghilangan prefix kata yang di hilangkan adalah http:// dan https:// . Untuk

penghilangan kata akhiran dilakukan dengan menghilangkan kalimat atau kata setelah karakter “#” dan menghilangkan karakter “/” atau “\” yang terdapat pada akhir kalimat. Contoh pengurangan kata awalan dan akhiran dapat dilihat misalnya pada link “https://en.wikipedia.org/wiki/Data_compression#Lossless/” untuk pengurangan pada awalan kata yaitu “https://” untuk pengurangan akhiran karena setelah karakter # ada kata maka karakter # hingga akhir dapat dihilangkan. Penghilangan awalan dan akhiran ini bisa dilakukan karena tanpa menggunakan kata yang dihilangkan link masih tetap dapat di akses dan memiliki nilai yang sama. Selain itu dalam proses ini juga dilakukan pengubahan link url menjadi *lower case* ,hal ini dilakukan agar jenis kata yang nanti disimpan sebagai kamus kata memiliki macam yang lebih sedikit.

3.3.3 Encoding Berbasis Kamus

Langkah selanjutnya adalah melakukan encoding pada data yang telah dilakukan pengurangan awalan dan akhiran. Metode yang dilakukan adalah sebagai berikut :



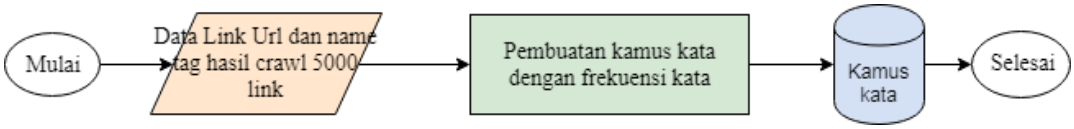
Gambar 3. 5 Encoding Berbasis Kamus

Pada gambar 3.5 dijelaskan untuk melakukan encoding dimulai dengan memasukkan url yang kemudian dilanjutkan dengan melakukan proses *tokenizing* untuk mengambil kata yang dipisahkan oleh karakter-karakter seperti /, . : = \ - _ ! ? & % $ #

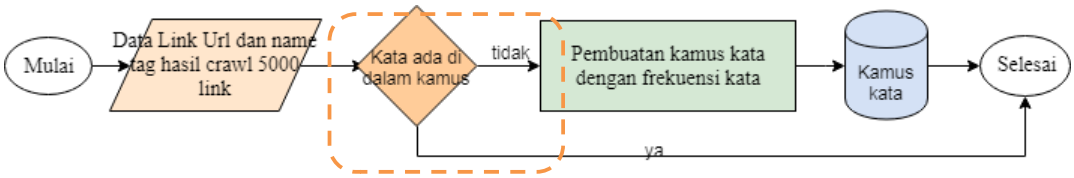
dan seterusnya. Setelah dilakukan pemisahan kata menurut karakter pemisah maka proses selanjutnya adalah pembuatan kamus kata.

Dikarenakan proses crawling pada sistem memiliki kemungkinan untuk dilakukannya penambahan pada file history pencarian web ke dalam bank kamus kata maka di dalam proses pembuatan kamus kata terdapat beberapa perbedaan proses. Pada awal mula aplikasi dibuat, kompresi data link dilakukan ketika hasil dari pencarian link sebanyak 5000 link yang di crawl telah diambil dan disimpan di dalam text. Ini dilakukan untuk mempermudah sistem dalam melakukan pembuatan kamus kata dengan menghitung frekuensi kemunculan kata kompresi dalam satu waktu. Untuk lebih jelasnya alur proses kompresi awal ditunjukkan dalam gambar 3.6.

Untuk selanjutnya proses yang terjadi adalah kompresi ketika dilakukan proses crawling ketika kamus kata sudah dibuat sebelumnya. Cara yang dilakukan adalah dengan mengecek kamus kata apakah data link dan tag nama baru yang telah dilakukan tokenizing ada pada kamus. Jika ada, maka indeks kata dalam kamus kata diambil dan tidak dilakukan proses pembuatan index baru pada kamus kata dan proses selesai. Jika tidak ada di dalam kamus kata, maka kata baru yang diperoleh dari hasil tokenizing akan dimasukkan kedalam kamus kata dengan nilai indek kata adalah nilai indek kata terakhir + 1, proses ini ditunjukkan pada gambar 3.7 , tanda ----- merupakan step yang membedakan kedua proses.



Gambar 3. 6 Alur Pembuatan Kamus Kata Awal Proses



Gambar 3. 7 Alur Pembuatan Kamus Kata Proses Selanjutnya

Dalam pembuatan kamus kata yang dilakukan adalah membuat data tabel yang berisi kata dan kode pada tiap kata yang ditemukan. Langkah pertama yang dilakukan

dalam pembuatan tabel adalah menghitung banyaknya kata yang ditemukan dalam data url yang akan dijadikan sebagai kamus kata. Misal ada 10 url dimana semua memiliki kata .com maka dalam tabel pertama yang dibuat menyimpan kata com dengan jumlah 10.

Tabel 3. 2 Tabel Jumlah Frekuensi Kata

Kata	Jumlah
liputan6	10
detik	3
news	10
viva	6
.	20
tribun	1
com	10

Setelah tabel frekuensi kata selesai dibuat maka selanjutnya adalah tahapan membuat kamus kata yang digunakan sebagai kamus untuk *encode* dan *decode* url yang ada.

Tabel 3. 3 Tabel Kamus Kata

Kata	Kode
.	1
com	2
liputan6	3
news	4
viva	5
detik	6
tribun	7

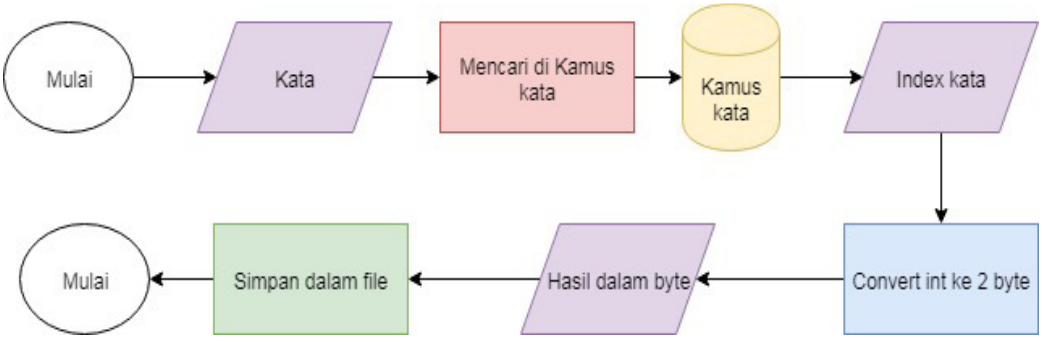
Dalam tabel kamus kata bisa dilihat dilakukan pengurutan penyimpanan kata berdasarkan jumlah kata yang ditemukan. Semakin banyak kata ditemukan didalam masukan maka kata tersebut akan mendapat nilai kode /indeks terkecil dan akan dicatat kedalam kamus kata terlebih dahulu. Hal ini dilakukan untuk kepentingan pencarian

nantinya, dimana meletakkan kata yang sering muncul pada indeks tabel lebih kecil akan mengurangi kinerja aplikasi/sistem untuk mencari data yang biasanya muncul.Dalam pembuatan kamus kata ini kode yang muncul adalah byte. Setiap kata akan menjadi 2 byte dalam penyimpanan, ini dilakukan karena kode yang disimpan merupakan representasi dari byte. Penyimpanan menggunakan 2 byte karena jumlah nilai bit pada maksimum angka bit 2^8 adalah hanya 0-255. Untuk menyiasati banyaknya jumlah jenis kata maka yang digunakan adalah 2^16. Dimana bit 2^16 memiliki variasi byte sebanyak 0-65.535 jenis.

Pembuatan kamus kata ini dilakukan pertama kali dengan cara melakukan pencatatan setiap kata yang muncul dalam link url yang telah dikumpulkan. Untuk pembuatan awal yang dilakukan adalah mencatat semua link yang dikumpulkan dari ribuan link yang tervisit. Semakin banyak link yang dikumpulkan di awal proses pembuatan kamus kata, kamus kata diharapkan memiliki banyak pengetahuan kata. Jika kamus memiliki banyak pengetahuan maka ketika dilakukan proses pencarian kata maka sistem akan menemukan kata teresebut didalam kamus tanpa melakukan pencarian kembali dan proses pendaftaran kata baru kedalam kamus. Hal ini akan lebih mempercepat kinerja sistem pencarian.

3.3.4 Simpan data hasil encoding

Hal yang dilakukan adalah merubah url dan nametag yang telah dikumpulkan dengan menulisnya sesuai index kata pada kamus kata yang sudah dibuat. Setelah index kata ditemukan barulah index dirubah menjadi 2 byte dan ditulis kedalam file text (test-encode.txt) dan merupakan data hasil encoding. Selanjutnya adalah menyimpan hasilnya kedalam aplikasi yang disimpan dalam file.

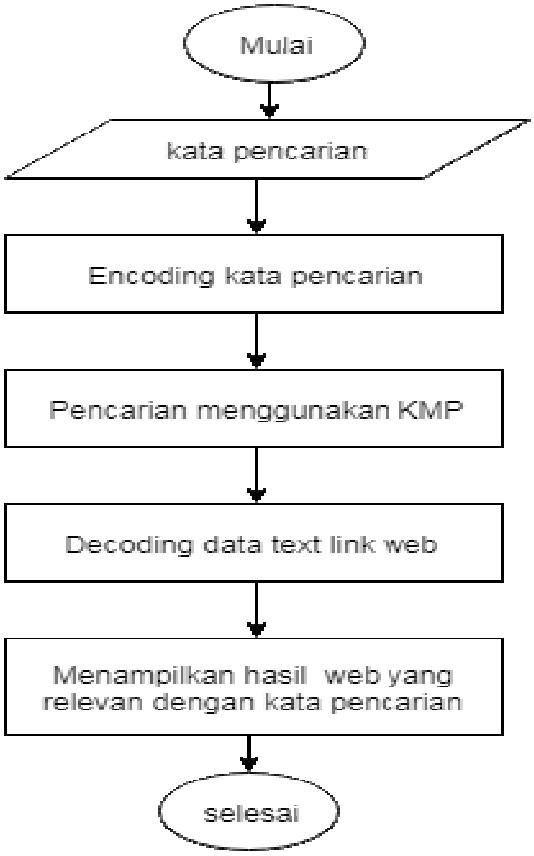


Gambar 3. 8 Alur Konversi Index ke dalam 2 byte

.Gambar 3.8 menunjukkan alur konversi index ke dalam 2 byte di dalam sistem.

3.4. Pencarian Hasil Link

Tahap selanjutnya adalah tahap pencarian hasil link pada data yang disimpan. Alur proses yang ada akan dijabarkan pada gambar 3.9 dibawah :



Gambar 3. 9 Alur Pencarian Hasil Link

Pada proses ini adalah bagaimana mencari link dari data *history* yang telah terencoding agar dapat memunculkan hasil pencarian. Langkah-langkah nya adalah sebagai berikut:

3.4.1 Input Kata Pencarian

Untuk proses ini kata pencarian yang digunakan adalah kata pencarian yang telah diinputkan oleh pengguna. Kata pencarian berupa string, string dapat berupa sebuah kata atau lebih.

3.4.2 Encoding Kata Pencarian

Langkah selanjutnya adalah memasukkan kata pencarian misal “detik”, dari kata yang dicari ini akan dilakukan proses encoding dengan menggunakan tabel kamus kata

yang sebelumnya telah dibuat. Untuk kata “detik” maka akan dirubah menjadi angka 6. Untuk kata misalkan “detik.com” yang akan ditulis kedalam variabel pencarian adalah 612. Penulisan dilakukan dengan menulis konversi 2 byte(16 bit) dari masing-masing angka. Untuk encoding link “detik.com” maka akan dihasilkan nilai 6 byte, dengan hasil akhir angka biner sebagai berikut 0000000000000110 0000000000000001 0000000000000010. Jika data kata tidak ditemukan didalam kamus kata maka yang dilakukan adalah memasukkan nilai index terakhir untuk kata kosong dengan nilai 65.355. Nilai index ini sengaja dibuat agar proses encode kata inputan tidak menghasilkan nilai null sehingga tidak terjadi kesalahan dalam program akibat nilai null.

3.4.3 Pencarian menggunakan Knutt Morris Pratt (KMP)

Tahap selanjutnya adalah pencarian kata pada data *history* yang sesuai, yaitu dengan melakukan pencocokan kata yang bisa disebut dengan pattern pada seluruh kata/pattern yang ada di dalam data *history*. Pencocokan dilakukan dengan memasukkan kata pencarian yang telah terencode lalu mencocokan satu persatu dari awal kata hingga akhir kata dalam data histori link yang disimpan dalam bentuk encode. Karena pada kompresi multilevel ini penyimpanan setiap index kata disimpan kedalam 2 byte binary maka pencarian dengan menggunakan algoritma KMP juga dilakukan dengan melakukan pencocokan per 2 byte binary.

Kelebihan pencarian dengan mengunakan KMP daripada pencocokan string biasa adalah jika pattern sudah ditemukan pada indeks kata ke sekian maka pencocokan dilakukan pada indeks kata selanjutnya, namun jika pencocokan string biasa hal ini akan diulangi pada awal kata lagi sehingga akan memiliki lebih banyak waktu pencarian.

3.4.4 Decoding Kata Inputan

Proses selanjutnya adalah proses decoding kata inputan yang telah ditemukan didalam file terencode. Karena proses pencocokan string menggunakan algoritma KMP hanya menemukan letak index dari kata inputan yang dicocokan maka perlu adanya skenario untuk ditambahkan agar url yang mengandung kata itu bisa di ambil dimana nantinya bisa ditampilkan kepada pengguna dalam bentuk url link dan name tag. Cara yang dilakukan adalah dengan menambahkan string batasan antar url, dalam aplikasi batasan url adalah karakter “ | ”. Dimana nantinya letak indeks string batasan akan dicari. Jika letak string-string batasan url telah ditemukan (dengan menggunakan

pencocokan KMP) selanjutnya akan dilakukan perhitungan dua jarak terdekat antara index letak kata inputan dengan letak index dua string batasan url. Gambar 3.10 merupakan alur dari cara decoding kata inputan.

Gambar 3. 10 Alur Decoding Pencarian Kata Inputan

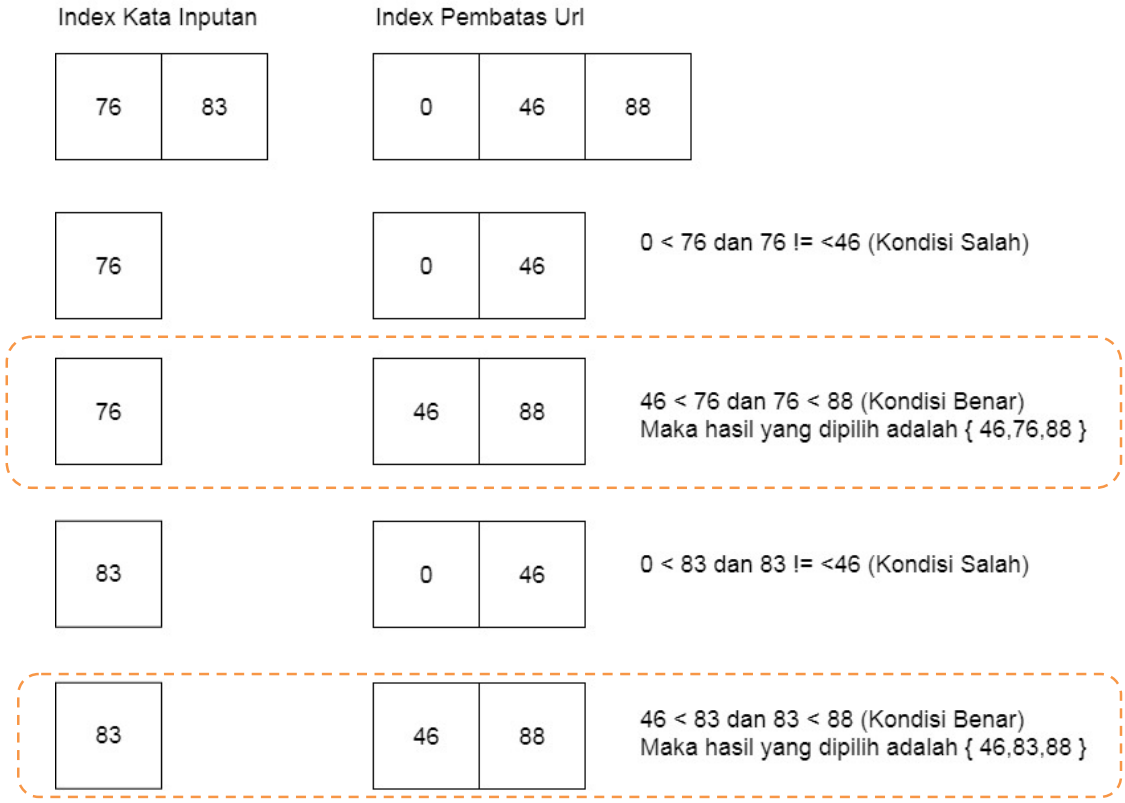
Untuk proses memperoleh rentang index pada kumpulan history link agar didapatkan url mana yang memiliki kata inputan dari pengguna. Contoh perhitungan rentang index : Berikut adalah contoh inputan halaman url web beserta name tag, dimana untuk pembatas adalah karakter “|” dan ditandai tanda [ ] merupakan pembatas url dan name tag. Dalam contoh dibawah merupakan 2 url dan nametag. Index ke 0 merupakan karakter “|” dan index ke 88 adalan “|” pada akhir kata. Gambar 3.11 merupkan visualisasi index pada pembatas url dan kata inputan di dalam data.

Gambar 3. 11 Index Pembatas Url dan Inputan Kata

Inputan misalkan adalah kata “hijab” maka akan dilakukan pencarian menggunakan KMP dan diperoleh dua index yakni index ke 76 dan index ke 83. Selain pencarian letak index inputan kata, index pembatas url juga dicari. Untuk contoh diatas index Perhitungan rentang index untuk pembatas url adalah ada pada index berikut 0, 46, dan 88. Setelah itu perhitungan nilai terdekat dilakukan dengan posisi pencarian adalah kata inputan dan untuk pencarian 2 titik terdekat di lakukan menggunakan data index pembatas url.

Pada Gambar 3.12 dibawah merupakan contoh perhitungan pencarian rentang index yang menunjukkan url di dalam data text. Setelah di lakukan pencarian dua nilai

index url terdekat dari index kata inputan maka hasil rentang akan diambil dan akan di lakukan proses decoding binary pada rentang index dengan kamus kata yang telah dibuat. Dari gambar 4.12 rentang yang didapat untuk dua index kata inputan yang ditemukan menghasilkan rentang {46,76,88} dan {46,83,88} karena kedua index pertama dan terakhir sama untuk menghilangkan url ganda dilakukan penghilangan salah satu rentang. Didalam percobaan ini yang dipakai adalah rentang yang pertama ditemukan yaitu {46,76,88} dan untuk rentang setelahnya {46,83,88} tidak dilakukan proses decode lagi.



Gambar 3. 12 Pengambilan Rentang Index

Pada proses decoding rentang index data *history* kumpulan link karena pada tahap sebelumnya encoding dilakukan dengan menggunakan kompresi string berbasis kamus, maka untuk proses decoding menggunakan data kamus kata juga seperti yang ada di dalam tabel kamus kata Tabel 3.3.

Karena data text yang terakhir disimpan adalah kumpulan byte dari url yang telah terencode, jika hasil penyimpanan url adalah nilai index 3,2,6,1,4 maka akan dilakukan pembacaan data dengan melihat tabel kamus kata yang telah dibentuk pada

proses encoding. Proses decoding di mulai dengan pembacaan tiap angka dan mengambil byte untuk dikonversikan kedalam kata atau karakter sebenarnya.

3.4.5 Menampilkan Hasil Pencarian

Selanjutnya adalah proses menampilkan link yang relevan dengan pencarian pengguna. Proses ini merupakan proses untuk menampilkan hasil decode link url yang memiliki kata pencarian dari pengguna. Setelah url terencode berhasil di decode sesuai dengan kamus kata langkah terakhir adalah menyajikan hasil decode link tadi menjadi string dan ditampilkan kepada pengguna dalam bentuk url beserta name tag. Hasil yang ditampilkan misalnya adalah http://detik.com: .

3.5. Pengujian

Pengujian bertujuan untuk mengevaluasi performa dari metode yang diusulkan dalam mengatasi permasalah yang telah dirumuskan. Pengujian dilakukan dengan menghitung nilai presisi dan recall (Rawat, 2012) pada hasil pencarian link website dari kata menggunakan focused web crawler cuckoo dan cuckoo + Kompresi multilevel . Perhitungan presisi dihitung dengan rumus:

$$p = a/(a + b) \tag{3}$$

dimana p = presisi , a = page yang memiliki konten sama/benar dengan pencarian , dan b = web page yang memiliki konten beda/salah dengan pencarian. Untuk perhitungan recall dilakukan dengan rumus:

$$r = a/c \tag{4}$$

dimana r = recall , a = page yang memiliki konten sama/benar dengan pencarian , dan c = seluruh page yang berhasil di download yang ada di dalam data set.Hasil perhitungan akan dimasukkan kedalam tabel perbandingan pada dua jenis pencarian *focused web crawler*.

Tabel 3. 4 Tabel *Fokused Web Crawler Cuckoo Search*

Keyword	Halaman relevan (a)	Halaman tidak relevan (b)	Presisi	Recall
Kerja	2	1	0.66666667	0.2

Sekolah alam	2	1	0.66666667	0.2
Rumah anak yatim	3	1	0.75	0.3
Total halaman	7	3		

Tabel 3. 5 Tabel *Fokused Web Crawler Cuckoo Search* + Kompresi multilevel

Keyword	Halaman relevan (a)	Halaman tidak relevan (b)	Presisi	Recall
Kerja	2	1	0.66666667	0.2
Sekolah alam	3	0	1	0.3
Rumah anak yatim	3	1	0.75	0.3
Total halaman	8	2		

Untuk evaluasi metode, penghematan ukuran juga dihitung. Yang dilakukan dengan perhitungan rasio , perhitungan ini menghitung perbandingan ukuran file asli dengan file hasil kompresi dengan rumus :

$$rasio = 100\ x\left(1 - \frac{Ukuran\ data\ setelah\ kompresi}{Ukuran\ data\ sebelum\ kompresi}\right) \quad (5)$$

Hasil dari perhitungan ini merupakan bentuk presentase, semakin besar nilai rasio maka semakin efisien kompresi yang dilakukan. Setelah mendapatkan hasil dari tiap-tiap percobaan akan dilakukan penarikan kesimpulan dari data yang telah dikumpulkan dan didapatkan bahwa metode yang diusulkan apakah lebih baik atau lebih buruk dari metode sebelumnya.

Tabel 3. 6 Tabel Rasio *Fokused Web Crawler Cuckoo Search* + Kompresi multilevel

Banyak Halaman	**Ukuran sebelum (byte)**	**Ukuran setelah(byte)**	**Rasio**
1000	3,000,000	1,900,000	36.7
2000	5,910,000	4,000,000	32.3
3000	8,900,000	4,660,672	47.6
4000	11,000,000	9,483,300	13.8
5000	19,039,808	9,445,376	50.4
6000	17,719,200	15,284,480	13.7

Untuk perhitungan lama waktu proses yang dibutuhkan untuk melakukan pencarian juga dilakukan perhitungan dengan cara :

$$waktu\ proses = waktu\ proses\ selesai - waktu\ mulai \quad (6)$$

Perhitungan lama waktu proses ini dilakukan dengan membandingkan dua pencarian *focused web crawler cuckoo search* saja dan *focused web crawler cuckoo search* + metode kompresi terhadap inputan pencarian kata. Dan akan ditampilkan dalam perbandingan tabel seperti dibawah :

Tabel 3. 7 Tabel Perbandingan Lama Proses.

Inputan Kata	Kecepatan FWC CK (detik)	Kecepatan FWC CK + Kompresi (detik)	Selisih (detik)
dipatuk king cobra	110	120	10
meminimalisir nyeri rahang	460	475	15
jus buah kemasan	910	930	20
perizinan online terpadu	1400	1420	20
pecahkan rekor berenang	3000	3050	50

[Halaman Sengaja Dikosongkan]

BAB 4
HASIL DAN PEMBAHASAN

Pada sub bab ini akan membahas tentang implementasi, pengujian dan pembahasan terkait penelitian yang diusulkan. Tahapan implementasi yang dilakukan sesuai dengan alur pada metodologi penelitian meliputi pencarian link url menggunakan *focused web crawler* dengan *cuckoo search*, kompresi url link menggunakan kamus kata , dan step terakhir adalah pencarian kata pada file kompresi. Untuk pengujian dilakukan menggunakan skenario pengujian pada bab 3 meliputi perhitungan rasio keberhasilan metode kompresi, ketepatan hasil pencarian kata meliputi perhitungan nilai presisi dan juga recall, serta perhitungan lama waktu proses pencarian kata.

4.1. Spesifikasi Perangkat Pengujian

Metode penyederhanaan ruang pada *focused web crawler* dengan Kompresi multilevel diimplementasikan menggunakan perangkat keras dan lunak seperti pada spesifikasi yang ada dalam tabel 4.1 dibawah ini:

Tabel 4. 1 Tabel spesifikasi perangkat pengujian

Nama	Spesifikasi
Processor	Intel(R) Core(TM) i5-4200U CPU @1.60GHz 2.30 GHz
RAM	4.00 GB
Sistem Operasi	Microsoft Windows 7 32-bit
Bahasa Pemrograman	Java ver 1.8
Tools	Eclipse , Jsoup library

4.2. Data Uji Coba

Data uji coba yang dipakai dalam penelitian ini adalah data hasil *focused web crawler* dari beberapa sumber link url. Sumber link url adalah merupakan website berita Indonesia seperti http://republika.co.id, https://www.detik.com/ , https://www.tribunnews.com , https://www.cnnindonesia.com/, https://www.viva.co.id/, https://www.merdeka.com dan http://www.liputan6.com.

Dari data url sumber dilakukan kunjungan sebanyak 6.000 link dan mengambil setiap link dan *name tag* yang ada didalam link yang dikunjungi untuk pembuatan bank

data pada proses awal. Untuk data pada proses pengujian kompresi data link dan name tag yang diambil Contoh pengambilan url dan name tag :

Tabel 4. 2 Tabel link dan *name tag*

Tag Nama	Alamat url
Liputan Khusus	https://www.liputan6.com/news/liputankhusus
Zona MPR RI	https://www.liputan6.com/news/zona-mpr-ri
Cek Fakta	https://www.liputan6.com/news/cek-fakta
Divonis 15 Tahun Penjara, Setya Novanto Tak Ajukan Banding	https://www.liputan6.com/news/read/3496583/divonis-15-tahun-penjara-setya-novanto-tak-ajukan-banding
Internasional	https://news.detik.com/internasional
Mensos Idrus Marham buka suara soal stiker cagub-cawagub Jatim yang diselipkan dalam penyaluran program PKH di Lamongan. Apa penjelasan Mensos?	https://news.detik.com/berita-jawa-timur/d-4000690/penjelasan-mensos-soal-stiker-cagub-jatim-di-program-pkh
100 Lebih WN China Ditangkap di Bali Terkait Kejahatan Siber	https://www.liputan6.com/news/read/3496570/100-lebih-wn-china-ditangkap-di-bali-terkait-kejahatan-siber

Untuk setiap link yang dikunjungi jumlah link dan name tag yang bisa disimpan atau ditemukan tidak akan sama. Hal ini tergantung dari jumlah link beserta name tag yang ada didalam sebuah website yang dikunjungi. Namun maksimum jumlah website untuk dikunjungi dapat dilakukan pembatasan sesuai dengan kebutuhan sistem.

4.3. Hasil dan Uji Coba

Pada sub bab ini akan dibahas mengenai hasil uji coba terhadap metode yang telah diusulkan. Tujuan dari tahap ini adalah untuk mengetahui performa dari metode yang diusulkan terhadap pengurangan ukuran file yang tersimpan sebagai *history* pencarian. Selain itu juga bertujuan untuk mengetahui apakah link hasil pencarian data

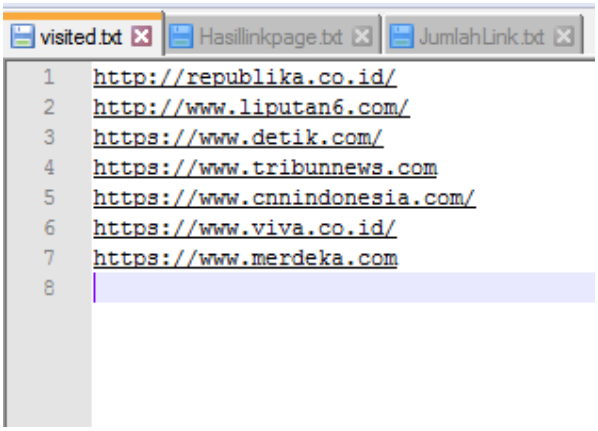
yang telah terkompres tidak mengalami perubahan data dan atau informasi dari data sebelum kompresi. Inputan yang dilakukan pengguna adalah kata atau kalimat.

4.3.1 Preproses Pengambilan Data Link

Pseudocode Preproses Pengambilan Data Link

```
Deklarasi
url sumber ,link search                          : list array string
visited                                          : hashset string
frontier , hasil ,url link ,name tag, next url   : string
max                                              : integer
==========================================================
Begin
set  url sumber  ←— tujuh website berita
frontier  ←— random (tujuh website berita)
While (frontier > 0) do
set  max        maksimum halaman crawl page.
        While (link search.size() < max) do
        hasil  ←— get url link + name tag.
        link search ←— add url link.
        visited  ←— hasil.
        next url ←—  url link.
        End While
url sumber  = remove frontier
End While
End
==========================================================
```

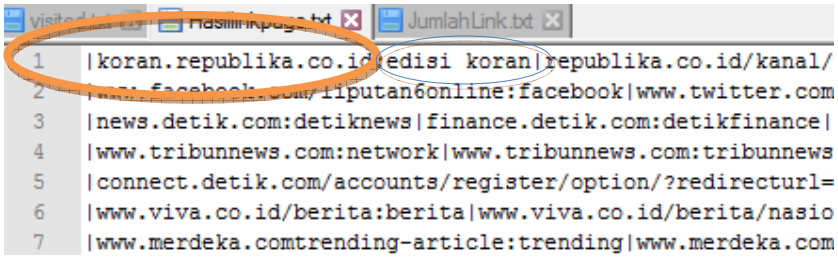
Dari ketujuh link website yang ada, ketujuh link tersebut dimasukkan kedalam sebuah array dimana akan diakses secara random untuk melakukan pencarian *frontier*. Setelah link frontier sudah dipilih misalkan http://www.liputan6.com/ , maka sistem akan melakukan pencarian (crawl) di internet dengan menggunakan pendekatan cuckoo search. Setelah itu untuk pengambilan data url dan name tag diambil dengan fungsi dari library jsoup. Dalam sekali sistem berjalan , jumlah pencarian link maksimum untuk di kunjungi dan diambil data link url dan *tag name* bisa ditentukan oleh admin program pada saat sistem dijalankan. Untuk uji coba jalannya sistem dilakukan percobaan pertama yang dilakukan adalah dengan mengeset maksimum crawl adalah 1. Dari pencarian pada tujuh halaman website dengan maksimum halaman crawl adalah 1 maka akan menghasilkan pencarian sebagai berikut:



Gambar 4. 1 Url Visited yang Tersimpan di sistem

Gambar 4.1 menunjukkan jumlah link website yang dicrawl/visit dalam proses pencarian. Setiap kali proses pencarian link yang telah diproses dicatat di dalam file "Visited.txt". Ini dilakukan agar url tidak lagi di crawl dan disimpan lagi oleh sistem. Karena maksimum yang ditentukan adalah satu halaman pada masing-masing frontier maka ada 7 halaman yang divisit. Gambar 4.2 merupakan hasil url dan name tag yang disimpan didalam history link.

Menunjukkan url dan merupakan name tag. Semua hasil disimpan kedalam file text.



Gambar 4. 2 Hasil Link Crawl dalam Bentuk Url dan Name Tag

Pada Gambar 4.3 menunjukkan semua link dan name tag yang dihasilkan dari proses pencarian. Link dan name tag pada satu halaman pencarian akan ditulis di setiap baris file penyimpanan. Dari tujuh halaman didapatkan tujuh baris kumpulan url dan name tag. Jumlah total url yang ada dalam sebuah halaman web juga dicatat(hanya untuk keperluan melihat hasil perolehan total link). Untuk hasil total jumlah url pada masing-masing halaman website dirangkum didalam gambar 4.4. Dari total hasil yang didapat, total perolehan url dan name tag sebanyak 3.212 .

|www.viva.co.id/sport/balap/1051204-jadwal-lengkap-f1-gp-inggris-5-8-juli-2018?medium=lihat-juga&campaign=lihat-juga-1:jadwal lengkap f1 gp inggris 5-8 juli 2018|www.viva.co.id/sport/balap/1047409-hamilton-start-terdepan-di-f1-gp-prancis?medium=lihat-juga&campaign=lihat-juga-2:hamilton start terdepan di f1 gp prancis |www.viva.co.id/tag/balapan-f1:balapan f1|www.viva.co.id/tag/f1-gp-inggris:f1 gp inggris|www.viva.co.id/tag/mercedes-f1:mercedes f1|www.viva.co.id/tag/lewis-hamilton:lewis hamilton

Gambar 4. 3 Hasil Link Crawl pada 7 Halaman yang Dikunjungi

Gambar 4. 4 Hasil Jumlah Link Pada Halaman Website yang di Crawl

Untuk percobaan selanjutnya, dilakukan pengambilan data url dengan melakuan pencarian url masing-masing sebanyak 1000, 2000, 3000, 4000, dan 5000 url. File hasil penyimpanan link dalam file text dan jumlah link yang diperoleh akan ditampilkan pada tabel 4.3.

Tabel 4. 3 Tabel jumlah link

Jumlah Maksimum Page Crawl	Ukuran file text (bytes)	Ukuran file text (MB)	Jumlah link yang ditemukan
1000	3,010,560	2.87	20.970
2000	5,914,624	5.64	41.687
3000	8,912,896	8.50	62.900
4000	11,751,424	11.2	83.444
5000	14,839,808	14.1	104.744
6000	17,719,296	16.8	124.898

4.3.2 Kompresi Data Link

Pseudocode Kompresi Data Link

```
Deklarasi
content,prefix, postfix ,in,enc                 : string
kata                                            : stringtokenizer
map, dictionary_encode                          : hashmap <string,integer>
dictionary_decode                               : hashmap< integer, string>
tabledata                                       : list
frek                                            : integer
b                                               :byte[]
tabledata_encode, tabledata_decode,encode       :txt file
=======================================================
Begin
//proses persiapan link dan kompresi 1
set  content  ⟵  link hasil crawl pada proses sebelumnya
content ⟵  delete content  prefix(http://, https://) dan postfix (karakter "/" atau "#")
//proses encoding berbasis kamus (kompresi 2)
kata  ⟵ tokenizing content
While kata memiliki token lagi
if kata tidak sama dengan null
        if map memiliki kata
                map ⟵  put kata, map get kata + 1
        else
                map  ⟵ put kata, 1
        end if
end if
tabledata  ⟵ add urutan data map value menurut frek kata
dictionary_encode  ⟵ add tabledata (string, integer)
dictionary_decode  ⟵ add tabledata (integer, string)
tabledata_encode ⟵  write tabledata(string)
tabledata_encode ⟵ write tabledata(integer)
End While
//proses penyimpanan hasil encoding
While membaca url belum selesai
in  ⟵ add url link pencarian
enc  ⟵ tokenizing in
if enc tidak ada di dictionary_encode
        remap kamus kata
else
        b  ⟵ convert integer ke bentuk 2 byte
        encode  ⟵  write b
end if
End While
End
=======================================================
```

Pada algoritma kompresi data link diatas akan dilakukan percobaan pada masing-masing bagian sub bab selanjutnya untuk tiap-tiap proses.

4.3.2.1 Persiapan data link

Proses selanjutnya adalah persiapan data link, data yang disiapkan adalah data hasil crawl pada 5000 link dari tujuh website berita. Proses persiapan link ini merupakan proses persiapan data link dan name tag untuk proses selanjutnya yaitu proses metode kompresi yang pertama penghilangan postfix dan prefix.

4.3.2.2 Penghilangan prefix dan postfix

Pada proses penghilangan prefix dan postfix di lakukan pada saat program melakukan crawl pada link website. Ini terjadi karena proses penghilangan bisa dibuat didalam fungsi crawl didalam class pencarian dan tidak membuat fungsi baru atau class baru untuk melakukan proses penghilangan prefix dan postfix. Selain penghilangan prefix dan postfix dalam tahap ini juga dilakukan pemberian karakter pembatas link dan name tag. Karakter pembatas yang digunakan adalah “|” pada setiap link dan name tag yang ditemukan. Pembatas berguna saat pengambilan link dan name tag pada sistem data setelah pencarian inputan kata dilakukan oleh algoritma string matching KMP.

Hasil percobaan akan ditampilkan url link sebelum dan sesudah di lakukan penghilangan prefix dan postfix pada gambar 4.5.

Gambar 4. 5 Perbandingan Hasil Penghilangan Prefix dan Postfix (Kompresi Tahap 1)

Gambar diatas merupakan sebagian link yang berhasil didapat dan dicoba ditampilkan. Dari hasil tersebut menunjukkan hasil link sebelum dan sesudah proses penghilangan prefix dan postfix. Untuk prefix kata http:// dan https:// dihilangkan, link yang memiliki jumlah lebih dari satu di sebuah halaman web hanya satu saja yang disimpan dalam history link penyimpanan. Contoh kasus dapat dilihat pada gambar 4.6 dan 4.7, Untuk kasus perubahan pertama seperti link http://www.tribunnews.com| ada

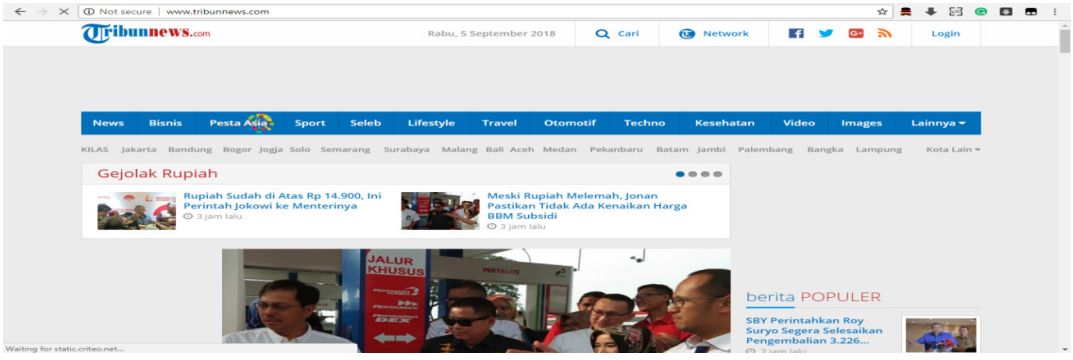
dua pada gambar sebelah kiri, setelah dilakukan penghilangan maka http://www.tribunnews.com| hanya dicatat satu saja pada gambar sebelah kanan yakni www.tribunnews.com.

Gambar 4. 6 Contoh Kasus perubahan 1

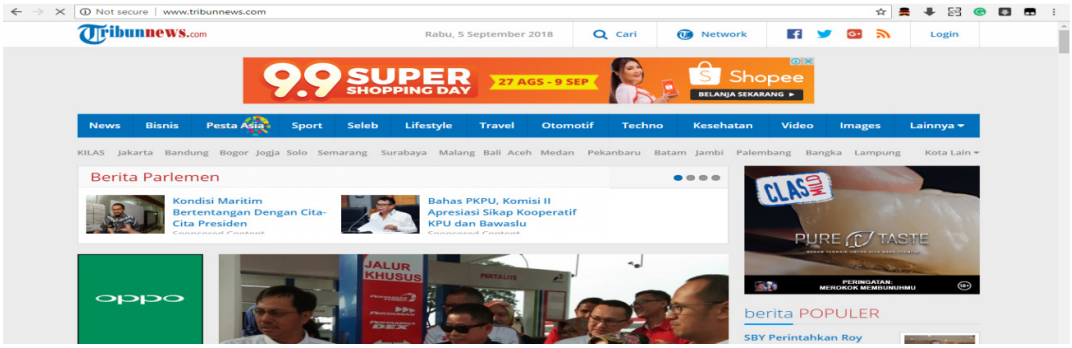
Kasus perubahan kedua, untu huruf "#" yang ditemukan (baris ke 35) juga dihilangkan sehingga urutan link yang tercatat menjadi lebih sedikit daripada sebelumnya namun menaikkan link setelahnya.

Gambar 4. 7 Contoh Kasus perubahan 2

Untuk mengecek apakah setelah penghilangan prefix dan postfix link masih bisa dijalankan berikut gambar hasil link yang di cari pada halaman browser. Link http://www.tribunnews.com dan www.tribunnews.com saat diketik pada browser menghasilkan halaman yang sama. Gambar 4.8 merupakan hasil dengan mengetikkan http://www.tribunnews.com kedalam pencarian browser. Sedangkan untuk gambar 4.9 merupakan hasil dengan mengetikkan www.tribunnews.com kedalam pencarian browser.



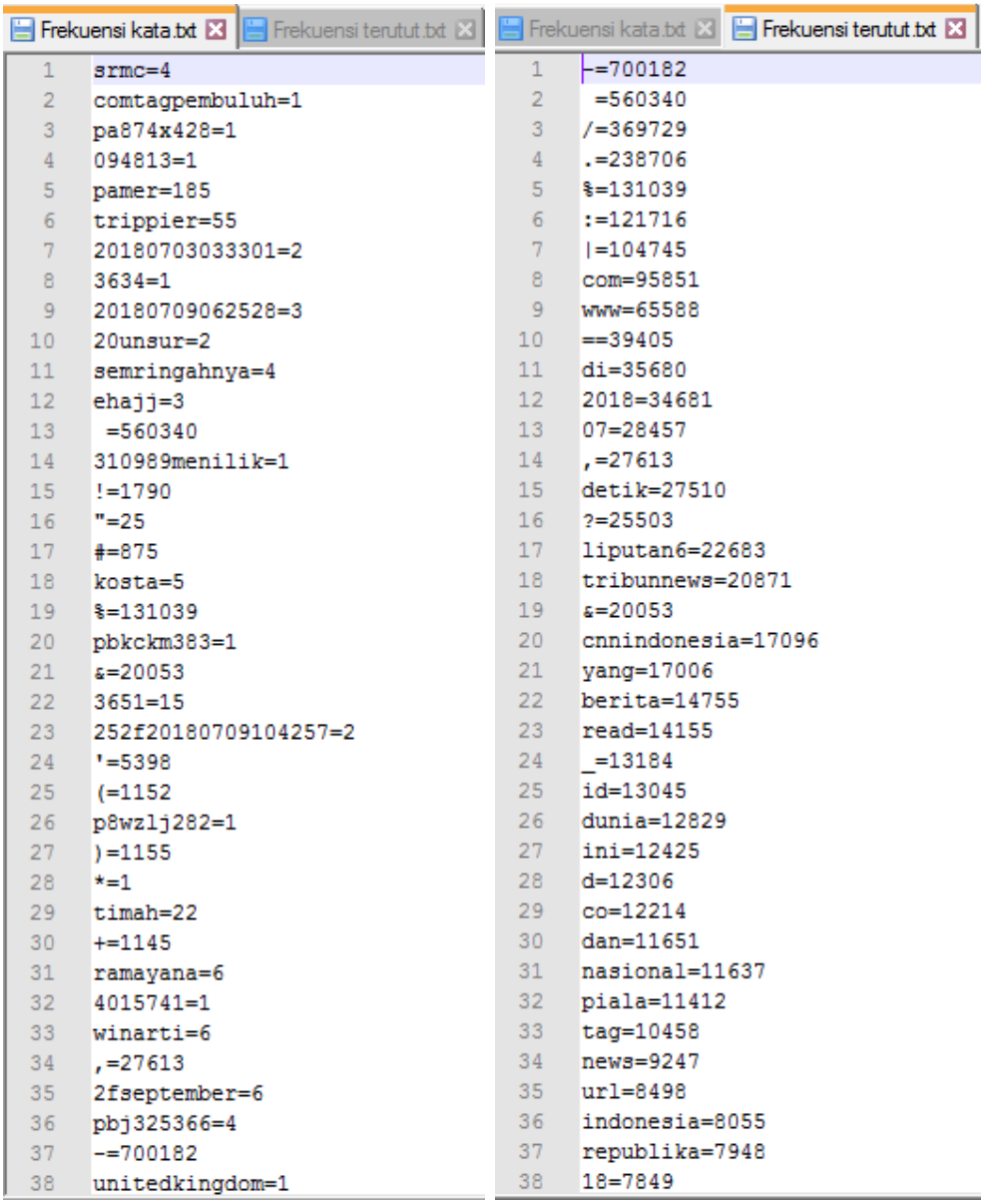
Gambar 4. 8 Hasil pencarian http://www.tribunnews.com



Gambar 4. 9 Hasil pencarian www.tribunnews.com

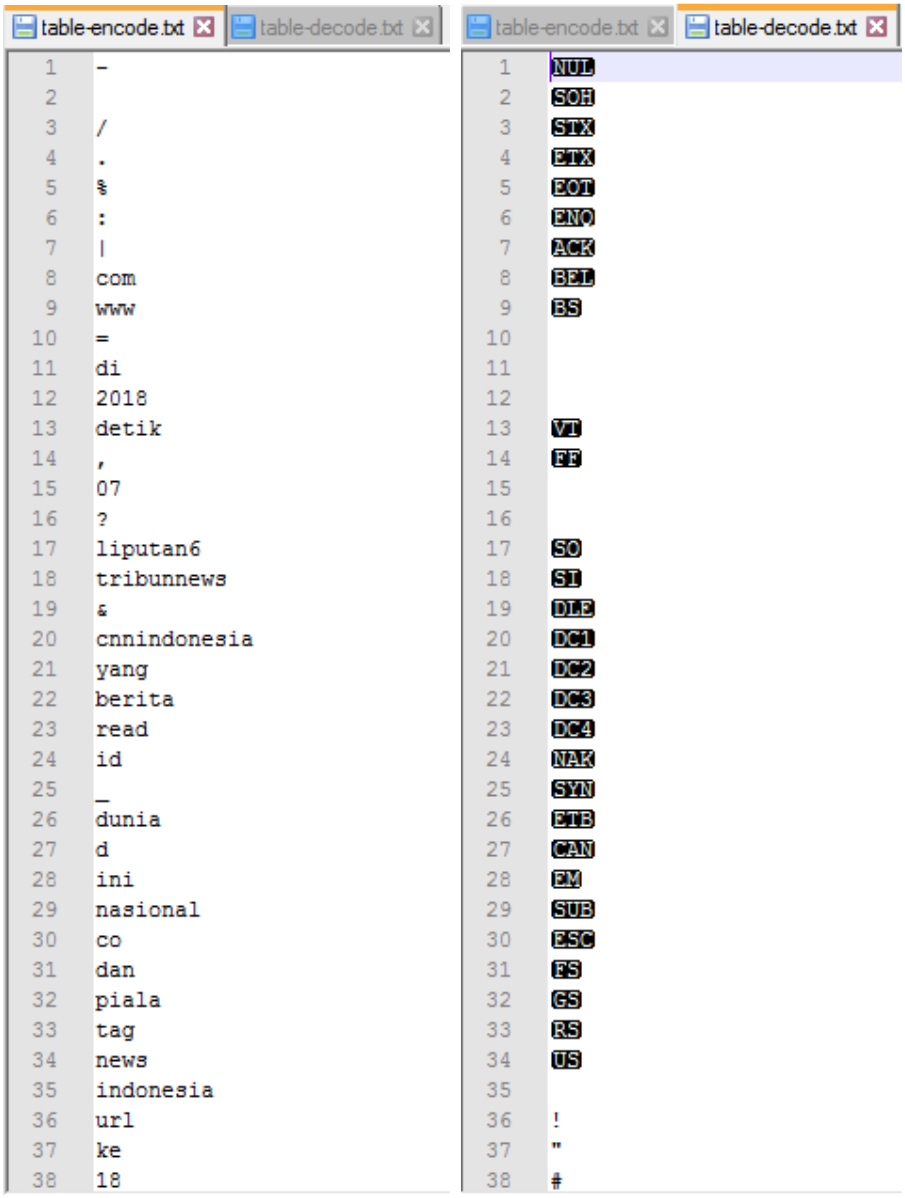
4.3.2.3 Encoding Berbasis Kamus

Proses selanjutnya adalah melakukan kompresi tahap kedua yakni encoding kata berbasis kamus. Dalam proses ini yang dilakukan adalah mengubah link dan name tag yang telah ada menjadi kamus kata kemudian menulis indeks dari hasil kamus kata untuk disimpan kedalam sistem. Proses pertama yang dilakukan adalah membuat kamus kata berdasarkan perhitungan frekuensi kata muncul pada data link dan name tag. Hasil perhitungan kata dari percobaan 5000 link yang telah dikumpulkan bisa terlihat pada gambar dibawah :



Gambar 4. 10 Hasil perhitungan frekuensi

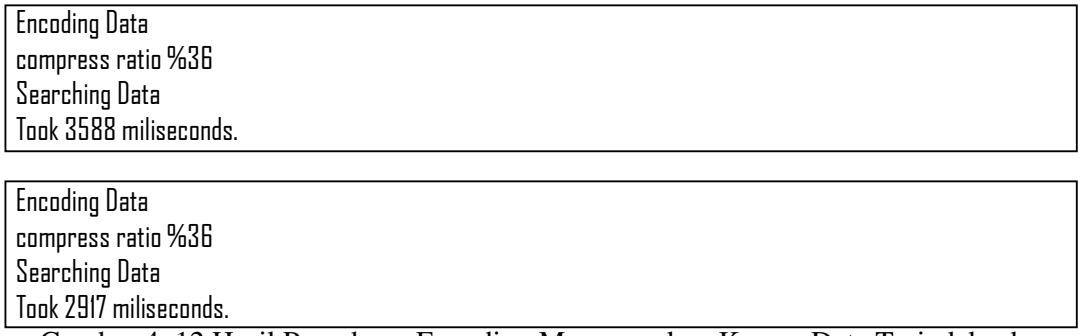
Pada gambar 4.10 ada dua output hasil perhitungan frekuensi dimana kata atau karakter dikiri huruf “=” menunjukkan kata dan angka dikanan huruf “=” adalah frekuensi kemunculan kata. Pada output sebelah kanan kata telah diurutkan dan disimpan pada tabel encode dan decode. Dimana tabel encode berisi kata, sedangkan tabel decode berisi angka index dimana kata itu terletak. Tabel encode maupun decode disimpan di dalam sistem dengan ekstensi file text.



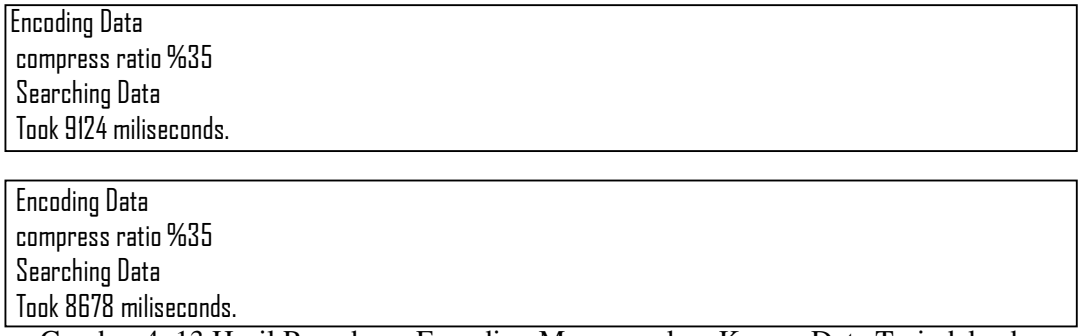
Gambar 4. 11 Tabel encode dan decode

Untuk membuktikan fungsi dari pengindeksan kata, dilakukan percobaan dengan skenario sistem dengan kamus data terindeks dan tidak. Data yang dilakukan untuk melakukan encoding adalah satu data text yang berisi url dan name tag dari 100,200, dan 300 link yang telah di crawl sebelumnya. Dengan data url dan name tag tersebut dilakukan proses encode yakni dengan pengecekan terhadap kamus kata yang ada(karena dianggap sudah ada) dan ini merupakan juga tahap pembuatan kamus kata bagian kedua.

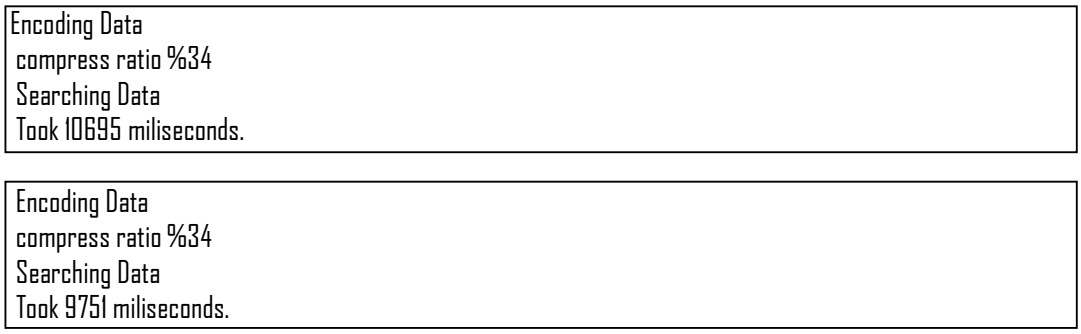
Dari hasil percobaan encoding dengan kamus kata yang telah ada di dapatkan hasil waktu pada kamus yang terindeks dan tidak. Gambar 4.12 , 4.13 dan 4.14 menunjukkan output dari ketiga percobaan yang telah dilakukan. Dan seluruh hasil dirangkum pada tabel 4.4.



Gambar 4. 12 Hasil Percobaan Encoding Menggunakan Kamus Data Terindeks dan Tidak Terindeks (100 link url)



Gambar 4. 13 Hasil Percobaan Encoding Menggunakan Kamus Data Terindeks dan Tidak Terindeks (200 link url)



Gambar 4. 14 Hasil Percobaan Encoding Menggunakan Kamus Data Terindeks dan Tidak Terindeks (300 link)

Tabel 4. 4 Tabel Hasil Percobaan Encoding Kamus Terindeks

Jumlah Halaman Crawl Url	Waktu proses (millisecond)		Selisih (millisecond)
	Kamus kata tanpa indeks	Kamus kata berindeks	
100	3588	2917	671
200	9124	8678	446
300	10695	9751	944

Pada Percobaan pertama menunjukkan proses encoding dengan kamus tanpa indeks memerlukan waktu 3588 miliseconds dan 2917 miliseconds untuk kamus berindeks. Selisih waktu proses adalah sebanyak 671 miliseconds. Untuk percobaan kedua memiliki selisih waktu proses sebanyak 446 miliseconds. Dan pada percobaan ketiga selisih waktu proses adalah sebanyak 944 miliseconds. Dari hasil percobaan encode menggunakan kamus kata terindeks lebih cepat daripada kamus kata tanpa indeks dan semakin banyak jumlah link yang di encode dalam satu waktu proses encode memiliki selisih rentang waktu proses tidak selalu semakin banyak namun membuktikan jika hasil encode berdasarkan kamus terindeks memiliki proses waktu yang lebih cepat.

4.3.2.4 Simpan data hasil encoding

Dari hasil data crawl pada 5000 link website yang telah diencode dan disimpan kedalam file text hasilnya nampak seperti pada gambar 4.15 dibawah. Terlihat hasilnya merupakan huruf encode yang tidak bisa dibaca sebelum ada proses decode terlebih dahulu.

Gambar 4. 15 Hasil Penyimpanan Data yang Terkonversi

Dari hasil data website berita yang telah disimpan, sebelumnya memiliki ukuran file sebesar 14.1 MB(14,839,808 bytes) setelah dilakukan proses kompresi multilevel menjadi 9.00 MB(9,445,376 bytes). Dari hasil yang di dapatkan kompresi multilevel dapat meringkas kebutuhan penyimpanan 5.1 MB(5,394,432 bytes). Jika di hitung presentasi dari ukuran file mula dapat meringkas sebesar 36.17%.

4.3.3 Pencarian Hasil Link

Proses selanjutnya adalah proses pencarian hasil link pada sistem pencarian. Pencarian hasil link ini merupakan proses yang dilakukan oleh pengguna untuk mencari link website sesuai dengan inputan yang diingikan.

Pseudocode Pencarian Hasil Link

```
Deklarasi
kata,pat, txt, pembatas, hasil          : string
part                                    : string[]
enc, i                                  : integer
rentang                                 : integer[]
dictionary_encode                       : hashmap <string,integer>
ListKata                                : ArrayList<String>
ListindexKata,ListindexPembatas         : ArrayList<Integer>
b                                       :byte[]
history link                            :txt file
============================================================
Begin
//proses input kata pencarian
set  kata
set pembatas
part ◄── split kata

//proses encoding kata pencarian
ListKata ◄── add part
for all ListKata[ i] do
     enc  ◄── get dictionary_encode kata
      b    ◄── convert nilai integer enc ke bentuk byte
     pat   ◄── add b ke bentuk string
end for
txt  ◄── add data pada history link

//proses pencarian KMP & decoding
ListindexPembatas ◄── add index kata hasil pencarian KMP dengan inputan (pembatas, txt)
ListindexKata  ◄──  add index kata hasil pencarian KMP dengan inputan (pat, txt)
for ListindexKata [i]
```

rentang ⟵ mencari dua nilai terdekat dari ListindexPembatas dengan inputan ListindexKata [i]
hasil ⟵ **add** hasil decode byte yang ada pada index rentang.

End for
End

==========

Untuk lebih jelasnya pada setiap step akan dilakukan percobaan seperti sub bab dibawah ini.

4.3.3.1 Input Kata Pencarian

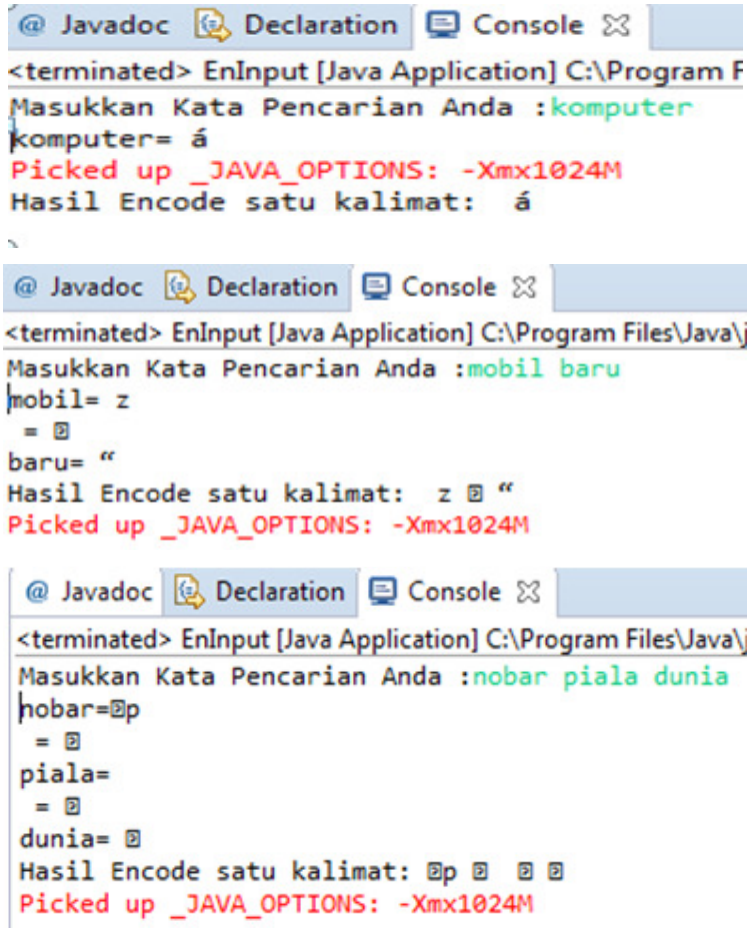
Inputan kata pencarian merupakan kata kunci untuk pencarian di sistem. Kata inputan ini dimasukkan oleh user sesuai dengan kebutuhan. Kata inputan tidak dibatasi panjang dan jenisnya(kata dasar, kata berawalan, kata berakhiran, maupun bersisipan). Namun untuk percobaan disini kata yang dimasukkan adalah kata dasar, kata berawalan, kata berakhiran, kata bersisipan dengan jumlah kata maksimum 3 inputan kata (kalimat dengan 3 kata penyusun). Contoh beberapa jenis inputan yang nanti akan diuji coba antara lain “komputer”, “mobil baru”, ”nobar piala dunia” dan beberapa kata lain yang nanti akan di uji coba.

4.3.3.2 Encoding Kata Pencarian

Proses yang dilakukan setelah pengguna menginputkan kata kunci adalah dengan encoding kata pencarian. Encoding kata pencarian adalah dengan melakukan pencarian indeks pada kamus kata yang telah di buat sebelumnya. Dalam encoding kata pencarian jika kata pencarian belum ada di dalam kamus kata yang dilakukan adalah melakukan pengisian index yang belum ada kata kunci sehingga akan memiliki nilai, dan bukan memberikan nilai null yang mengakibatkan kesalahan pada sistem. Kata inputan harus di encode karena untuk menemukan file yang telah terencode jenis kata inputan haruslah sama. Dibawah ini merupakan sintax untuk mendapatkan nilai index kata inputan pada kamus kata.

```
 public String EncodeKata(String ek)
{
 ConverterUtil converterUtil = new ConverterUtil();
 String pat;
 if (dictionary_encode.get(ek) == null) {
    // mengeset nilai index kata yang belum ada
      byte[] ben = converterUtil.intToBytes(65355);
           pat = new String(ben); }
 else {
           byte[] ben = converterUtil.intToBytes(dictionary_encode.get(ek));
           pat = new String(ben); }
        return pat;
 }
```

Karena inputan kata merupakan satu kata atau beberapa kata yang digabung menjadi sebuah kalimat, maka untuk masukan kata inputan harus di masukkan kedalam sebuah array string yang berisi kata dengan cara memisah kata dan karakter satu persatu.



Gambar 4. 16 Contoh Output Encoding Kata Inputan

Gambar 4.16 merupakan contoh hasil encoding kata pencarian. Pada gambar tersebut tiga macam inputan kata dicoba yaitu satu kata inputan, dua kata inputan dan tiga kata inputan.

4.3.3.3 Pencarian menggunakan Knutt Morris Pratt (KMP)

Proses selanjutnya adalah pencarian dengan alogaritma KMP. Dalam pencarian KMP data yang diuji coba adalah data hasil crawl 5000 link website yang sebelumnya telah terencode. Inputnya adalah berupa kata inputan yang sebelumnya telah terencode dan output merupakan letak index kata ditemukan. Berikut percobaan untuk pencarian menggunakan algoritma pencocokan string. Inputan yang dicoba berupa satu, dua dan

tiga inputan kata. Beberapa kata inputan yang dicoba akan ditunjukkan pada gambar 4.17, 4.18 dan 4.19 dibawah.

```
Masukkan Kata Pencarian Anda :komputer
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[141604, 141640, 194196, 194240, 851932, 1200688, 2360552, 3059840, 3570104, 3570132, 4976512, 5244736,
6412064, 6412096, 6436756, 6886596, 6886628, 7248164, 7248196, 7532040, 7588732, 7588764, 8095220]
Found: 23 match
Pencarian Terkait kata 'komputer' sebanyak 23 :
Hasil Pencarian sebanyak 16 kata
Total execution time: 1708 milisecond
====================================
Pencarian Sebelumnya + KMP
Pattern ditemukan pada index ke :
[222537, 222598, 302075, 302136, 1349233, 1349242, 1894474, 1894483, 3711489, 3711498, 4810189, 4810198,
5611149, 5611200, 7819460, 7819469, 8250083, 8250092, 10068989, 10069038, 10106005, 10106014, 10810165,
10810214, 11375922, 11375973, 11820606, 11820615, 11913306, 11913355, 12714106, 12714115]
Found: 32 match
Pencarian Terkait kata 'komputer' sebanyak 32 :
Hasil Pencarian sebanyak 16 kata
Total execution time: 558 milisecond
Picked up _JAVA_OPTIONS: -Xmx1024M
```

Gambar 4. 17 Hasil Pencocokan String dengan KMP inputan 1 kata

```
Masukkan Kata Pencarian Anda :mobil baru
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[776468, 1020196, 1421016, 1723248, 2490900, 3187840, 3708220, 4252876, 4853708, 5524632, 5608932,
5829488, 5869520, 6078124, 6291676, 6687972, 6688488, 7086156, 7333192, 7713728, 7940168, 8570188]
Found: 22 match
Pencarian Terkait kata 'mobil baru' sebanyak 22 :
Hasil Pencarian sebanyak 22 kata
Total execution time: 1996 milisecond
====================================
Pencarian Sebelumnya + KMP
Pattern ditemukan pada index ke :
[1226981, 1616539, 2247475, 2720696, 3917003, 5007955, 5821000, 6680864, 7623042, 8701605, 8830519,
9164471, 9223988, 9540101, 9880759, 10494875, 10495604, 11124899, 11501646, 12107595, 12467454, 13466426]
Found: 22 match
Pencarian Terkait kata 'mobil baru' sebanyak 22 :
Hasil Pencarian sebanyak 22 kata
Total execution time: 546 milisecond
Picked up _JAVA_OPTIONS: -Xmx1024M
```

Gambar 4. 18 Hasil Pencocokan String dengan KMP inputan 2 kata

```
Masukkan Kata Pencarian Anda :nobar piala dunia
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[13412, 22392, 22536, 23320, 56544, 93304, 94520, 97476, 104828, 164728, 166396, 169928, 402432, 424608,
468208, 478892, 531268, 535764, 543616, 554812, 559588, 561844, 565896, 566208, 590392, 598288,
616324, 617068, 617372, 620684, 633320, 639348, 644224, 730212, 762148, 767692, 1324248, 1367504,
1827680, 1892548, 2057512, 2057596, 2057720, 2266372, 2568508, 2698532, 2726476, 2726560, 2726684,
2747440, 2747588, 2842628, 2893780, 2893864, 2893988, 2922184, 2993368, 3212656, 3218324, 3218408,
3218532, 3265428, 3382856, 3862736, 3970564, 3973228, 3985384, 4011472, 4011556, 4011680, 4092460,
4589168, 4591764, 4593260, 4624444, 4624528, 4624652, 4643692, 4668420, 5993332, 6127456, 6477256,
6740948, 7341756, 7364204, 9170692, 9172284, 9299764, 9379312]
Found: 89 match
Pencarian Terkait kata 'nobar piala dunia' sebanyak 89 :
Hasil Pencarian sebanyak 89 kata
Total execution time: 6181 milisecond
================================================
Pencarian Sebelumnya + KMP
Pattern ditemukan pada index ke :
[22845, 37474, 37716, 39065, 92673, 151575, 153374, 157792, 168925, 258090, 260466, 265862, 631763,
667473, 740157, 757116, 844053, 851078, 863427, 881281, 888350, 891789, 898057, 898501, 935038, 946749,
973561, 974705, 975204, 980272, 999000, 1007979, 1014970, 1151780, 1203674, 1212360, 2092679, 2165704,
2885754, 2982166, 3232428, 3232553, 3232733, 3562701, 4042509, 4245433, 4287030, 4287155, 4287335,
4319189, 4319436, 4464563, 4540998, 4541123, 4541303, 4584347, 4702434, 5045061, 5053367, 5053492,
5053672, 5125385, 5313988, 6071585, 6235324, 6239046, 6257617, 6295424, 6295549, 6295729, 6428211,
7211875, 7215468, 7217729, 7264080, 7264205, 7264385, 7293424, 7332244, 9409982, 9618339, 10169928,
10573811, 11514649, 11548466, 14406229, 14408614, 14601752, 14721554]
Found: 89 match
Pencarian Terkait kata 'nobar piala dunia' sebanyak 89 :
Hasil Pencarian sebanyak 89 kata
Total execution time: 605 milisecond
Picked up _JAVA_OPTIONS: -Xmx1024M
```

Gambar 4. 19 Hasil Pencocokan String dengan KMP inputan 3 kata

4.3.3.4 Decoding Kata Inputan

Pada gambar 4.20 dan 4.21 merupakan hasil rentang yang diperoleh dari perhitungan nilai terdekat . Kata inputan yang digunakan dalam percobaan ini adalah kata “komputer”. Gambar 4.20 adalah hasil ketika telah dilakukan penghilangan rentang yang memiliki index pertama dan terakhir sama (dianggap dalam 1 url). Gambar 4.21 merupakan hasil sebelum dilakukan penghilangan multiple url yang memiliki beberapa indeks inputan kata dalam sebuah url dan nametag. Dari kedua percobaan dengan melihat kesamaan url , rentang url yang seharusnya ditulis 29 url bisa

dihilangkan 9 halaman. Kesembilan Halaman dari contoh percobaan sebagai url yang dihilangkan bisa dilihat pada list dibawah ini.

Hasil Double: [141550, 141640, 141642]

Hasil Double: [194146, 194240, 194258]

Hasil Double: [3570038, 3570132, 3570158]

Hasil Double: [6412022, 6412096, 6412114]

Hasil Double: [6886554, 6886628, 6886646]

Hasil Double: [7248090, 7248196, 7248198]

Hasil Double: [7588690, 7588764, 7588782]

Hasil Double: [10259054, 10259080, 10259082]

Hasil Double: [10259170, 10259188, 10259190]

```
Masukkan Kata Pencarian Anda :komputer
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[141604, 141640, 194196, 194240, 851932, 1200688, 2360552, 3059840, 3570104, 3570132, 4976512, 5244736,
6412064, 6412096, 6436756, 6886596, 6886628, 7248164, 7248196, 7532040, 7588732, 7588764, 8095220]
Found: 16 match
Hasil: [141550, 141604, 141642]
Hasil: [194146, 194196, 194258]
Hasil: [851918, 851932, 851934]
Hasil: [1200674, 1200688, 1200690]
Hasil: [2360538, 2360552, 2360554]
Hasil: [3059826, 3059840, 3059842]
Hasil: [3570038, 3570104, 3570158]
Hasil: [4976498, 4976512, 4976514]
Hasil: [5244722, 5244736, 5244738]
Hasil: [6412022, 6412064, 6412114]
Hasil: [6436742, 6436756, 6436758]
Hasil: [6886554, 6886596, 6886646]
Hasil: [7248090, 7248164, 7248198]
Hasil: [7532026, 7532040, 7532042]
Hasil: [7588690, 7588732, 7588782]
Hasil: [8095206, 8095220, 8095222]
Hasil Pencarian sebanyak 16 link
Total execution time: 2970 milisecond
```

Gambar 4. 20 Pengambilan Rentang dengan Memperhatikan Kesamaan Url

```
Masukkan Kata Pencarian Anda :komputer
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[141604, 141640, 194196, 194240, 851932, 1200688, 2360552, 3059840, 3570104, 3570132, 4976512, 5244736,
6412064, 6412096, 6436756, 6886596, 6886628, 7248164, 7248196, 7532040, 7588732, 7588764, 8095220]
Found: 23 match
Hasil: [141550, 141604, 141642]
Hasil: [141550, 141640, 141642]
Hasil: [194146, 194196, 194258]
Hasil: [194146, 194240, 194258]
Hasil: [851918, 851932, 851934]
Hasil: [1200674, 1200688, 1200690]
Hasil: [2360538, 2360552, 2360554]
Hasil: [3059826, 3059840, 3059842]
Hasil: [3570038, 3570104, 3570158]
Hasil: [3570038, 3570132, 3570158]
Hasil: [4976498, 4976512, 4976514]
Hasil: [5244722, 5244736, 5244738]
Hasil: [6412022, 6412064, 6412114]
Hasil: [6412022, 6412096, 6412114]
Hasil: [6436742, 6436756, 6436758]
Hasil: [6886554, 6886596, 6886646]
Hasil: [6886554, 6886628, 6886646]
Hasil: [7248090, 7248164, 7248198]
Hasil: [7248090, 7248196, 7248198]
Hasil: [7532026, 7532040, 7532042]
Hasil: [7588690, 7588732, 7588782]
Hasil: [7588690, 7588764, 7588782]
Hasil: [8095206, 8095220, 8095222]
Hasil Pencarian sebanyak 16 link
Total execution time: 1810 milisecond
```

Gambar 4. 21 Pengambilan Rentang tanpa Memperhatikan Kesamaan Url

Untuk proses decode menggunakan kamus kata hasilnya bisa terlihat dari gambar 4.22 dibawah. Dalam gambar 4.22 akan ditampilkan binary terencode dalam rentang url dan hasil setelah dilakukan decode.

```
Masukkan Kata Pencarian Anda :komputer
Pencarian Menggunakan Multilevel Kompresi + KMP
Pattern ditemukan pada index ke :
[141604, 141640, 194196, 194240, 851932, 1200688, 2360552, 3059840, 3570104, 3570132, 4976512,
Found: 23 match
Pencarian Terkait kata 'komputer' sebanyak 23 :
Sebelum terdecode:  ▯ Í ▯ ▯ ▯ ▯ ▯ ▯ ▯  ▯° ▯ß^ ▯▯X  ▯Ÿ  ▯▯  ▯ë  ▯?  WÝ     Ü  ³ ▯▯X ▯▯Ÿ ▯▯▯
æ‟ ▯▯ë ▯▯? ▯WÝ ▯        Ü ▯³
Setelah terdecode: null|health.detik.com/berita-detikhealth/3934444/cegah-kerja-lembur-korsel-
Sebelum terdecode:  ▯ ! ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯  ÌH ▯ =  ▯Ø
  ▯▯  =ù  ▯,  ³  ▯?  C6   T  ▯
 ▯ = ▯▯Ø ▯
 ▯▯▯ ▯=ù ▯▯, ▯³ ▯▯? ▯C6 ▯ T ▯▯
```

Gambar 4. 22 Hasil Decode Rentang Url Terencode

4.3.3.5 Menampilkan Hasil Pencarian

Selanjutnya adalah menampilkan hasil pencarian dari proses sebelumnya. Tampilan yang diberikan kepada pengguna adalah url dan name tag. Pada percobaan kata inputan yang dicari adalah "komputer". Hasil url yang ditampilkan adalah seperti gambar 4.25 dibawah ini :

```
Masukkan Kata Pencarian Anda :komputer
Pencarian Menggunakan Kompresi multilevel + KMP
Pattern ditemukan pada index ke :
[141604, 141640, 194196, 194240, 851932, 1200688, 2360552, 3059840, 3570104, 3570132, 4976512, 5244736,
6412064, 6412096, 6436756, 6886596, 6886628, 7248164, 7248196, 7532040, 7588732, 7588764, 8095220]
Found: 23 match
Pencarian Terkait kata 'komputer' sebanyak 23 :
|health.detik.com/berita-detikhealth/3934444/cegah-kerja-lembur-korsel-vivo-pekerjanya-matikan-
komputer:cegah kerja lembur, korsel vivo pekerjanya matikan komputer
|news.detik.com/berita/d-4102018/2-maling-di-aceh-gondol-11-komputer-secara-yayasan-anak-yatim:2 maling
di aceh gondol 11 komputer secara yayasan anak yatim
|www.tribunjualbeli.comkomputer:komputer
|www.tribunjualbeli.comkomputer:komputer
|www.tribunjualbeli.comkomputer:komputer
|www.tribunjualbeli.comkomputer:komputer
|www.kevin.co.id/berita/jurnal-haji/kabar-dari-tanah-suci/18/04/13/p74ae4385-data-komputer-saqdhar-
kaligafer-resmi-kiswah-kabah:data komputer saqdhar kaligafer resmi kiswah ka'bah
|www.tribunjualbeli.comkomputer:komputer
|www.tribunjualbeli.comkomputer:komputer
|www.tribunnews.com/techno/2018/07/09/5-ipad-dan-komputer-baru-resmi-20owi-apple:5 ipad dan
komputer baru resmi 20owi apple
|www.tribunjualbeli.comkomputer:komputer
|www.tribunnews.com/techno/2018/07/09/5-ipad-dan-komputer-baru-resmi-20owi-apple:5 ipad dan
komputer baru resmi 20owi apple
|trendtek.republika.co.id/berita/trendtek/internet/18/06/26/paxcuw349-ternyata-ada-beragam-cara-kirim-
sms-lewat-komputer:ternyata ada beragam cara kirim sms lewat komputer
|www.tribunjualbeli.comkomputer:komputer
|www.tribunnews.com/techno/2018/07/09/5-ipad-dan-komputer-baru-resmi-20owi-apple:5 ipad dan
komputer baru resmi 20owi apple
|www.tribunjualbeli.comkomputer:komputer
Hasil Pencarian sebanyak 16 link
Total execution time: 2591 milisecond
```

Gambar 4. 23 Hasil Pencarian Url

4.3.4 Pengujian dan Analisa

Pengujian yang dilakukan didalam penelitian meliputi tiga hal yakni pengujian analisa presisi dan recall hasil output link website untuk mengetahui kebenaran informasi setelah dilakukan proses kompresi dengan metode kompresi multilevel yang diusulkan. Pengujian lainnya adalah pengujian rasio kompresi file history pencarian website untuk mengetahui berapa persen metode kompresi dapat menghemat ruang pada disk. Untuk pengujian yang ketiga adalah uji coba waktu pencarian *keyword* oleh pengguna antara metode pencarian focused web crawler sebelumnya dengan metode pencarian focused web crawler dengan kompresi multilevel untuk mengetahui perbedaan waktu yang dibutuhkan kedua metode pencarian.

4.3.4.1 Uji Coba dan Analisa Presisi dan Recall

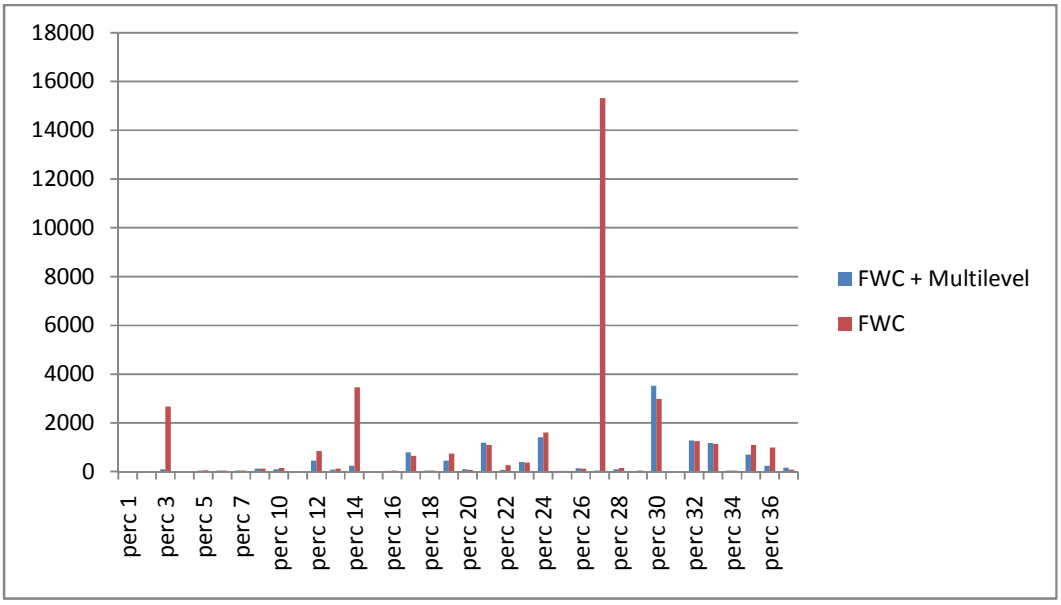
Uji Coba dilakukan dalam pencarian beberapa kata dan frasa. Untuk hasil yang didapat akan dilakukan pengecekan secara manual dengan melihat kebenaran penemuan kata yang dicari dari hasil link yang didapatkan oleh pencarian. Berikut akan disajikan beberapa tabel hasil dari pencarian pada dua metode pencarian *Focused Web Crawler* + *Cuckoo Search* + KMP (penelitian sebelumnya) ditulis **FWC** dan pencarian *Focused Web Crawler* + *Cuckoo Search* + *Kompresi multilevel* + KMP ditulis **FWCM** . Pencarian dilakukan terhadap beberapa masukan kata yaitu percobaan yang dilakukan dari satu kata, dua kata dan tiga kata inputan dari user. Dalam percobaan presisi dan recall ini dilakukan terhadap link yang ada pada 5.000 halaman web yang telah tercrawler dengan total url link yang tersimpan adalah 104.744 link url. Untuk mendapatkan nilai presisi dan recall, pada masing-masing hasil output link akan di lakukan pengklasifikasian yakni link benar dan link salah. Link benar adalah link yang memiliki hasil output sesuai dengan kata pencarian dan link adalah link yang valid (bisa dijalankan ketika dimasukkan pada browser), sedangkan link yang salah adalah link sesuai inputan namun tidak valid. Dalam sub bab percobaan ini link benar ditulis **LB** dan link salah ditulis **LS**.

4.3.4.1.1 Uji Coba dan Analisa Satu Kata Pencarian

Pada uji coba satu kata pencarian dilakukan dengan melakukan percobaan pencarian kata untuk 1 kata yang berupa kata dasar dan kata bukan kata dasar (memakai awalan/akhiran/keduanya).

1. Kata Dasar

Dari pencarian link website yang telah dilakukan pada metode penelitian sebelumnya focused web crawler cuckoo search dan pada metode focused web crawler cuckoo search dengan kompresi multilevel didapatkan beberapa hasil yang berpengaruh pada perhitungan presisi dan recall link website hasil. Untuk jumlah link yang didapatkan dari 36 percobaan menunjukkan bahwa metode focused web crawler memiliki hasil yang lebih banyak daripada metode focused web crawler dengan kompresi multilevel. Dari 36 percobaan, 10 percobaan memiliki hasil link perolehan lebih sedikit sedangkan sisanya sejumlah 26 memiliki link yang lebih banyak. Hasil percobaan pencarian yang telah dilakukan untuk satu kata inputan yakni dengan menggunakan kata dasar dengan 36 percobaan kata didapatkan data seperti grafik gambar 4.24 dibawah ini:



Gambar 4. 24 Grafik Pencarian Link Inputan Satu Kata Dasar

Pada hasil pencarian link metode focused web crawler tanpa menggunakan kompresi multilevel dari 26 percobaan kata dasar yang memiliki jumlah lebih banyak dapat di lihat dari tabel 4.5. Untuk percobaan yang ditampilkan pada tabel 4.5 merupakan percobaan yang memiliki selisih jumlah hasil link yang cukup banyak. Untuk sampel rentang, yang diambil adalah 10 terbanyak. Berikut data dari 10 percobaan yaitu perc 27,perc 14,perc 3,perc 36, perc 35, perc 12,perc 19,perc 22, perc

24, dan perc 10 yang dirangkum dalam tabel 4.5. Urutan percobaan yang ditampilkan sesuai dengan jumlah selisih link.

Tabel 4. 5 Tabel Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih banyak dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Dasar

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
27	nasi	15320	43	15277
14	ikan	3461	235	3226
3	alam	2672	105	2567
36	seni	989	241	748
35	sehat	1096	706	390
12	hilang	846	463	383
19	kerja	739	442	297
22	lari	278	76	202
24	masuk	1606	1408	198
10	hantu	149	95	54

Untuk ke 10 nilai dimana pencarian link pada metode dengan penambahan kompresi multilevel memiliki hasil link yang lebih banyak daripada metode sebelumnya terjadi pada perc 30, perc 17,perc 21,perc 37,perc 33,perc 20,perc 32, perc 23 dan perc 26. Berikut data dari kesepuluh hasil percobaan yang dirangkum dalam tabel 4.6 . Urutan tabel yang ditampilkan sesuai dengan banyaknya selisih metode tanpa kompresi multilevel dan dengan kompresi multilevel.

Tabel 4. 6 Tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih sedikit dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Dasar

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
30	olahraga	2986	3517	531
17	juara	647	796	149

21	korban	1106	1191	85
37	serius	88	166	78
33	rumah	1132	1167	35
20	konten	72	103	31
32	perang	1251	1278	27
23	listrik	374	399	25
26	murah	128	135	7

Untuk perbedaan yang signifikan pada kasus metode sebelumnya memiliki jumlah url temuan yang lebih banyak daripada metode yang diusulkan, dilakukan pengamatan terhadap hasil link yang dihasilkan. Dari pengamatan yang dilakukan ada dua jenis hasil pencarian yang dihasilkan yaitu kata yang sama persis dengan inputan saja dan kata yang mengandung kata inputan.

www.cnnindonesia.com/teknologi/20180706164544-387-312094/infografis-10-mobil-terlaris-mei-2018:infografis: 10 mobil terlaris mei 2018 teknologi • 4 jam yang lalu

news.detik.com/foto-news/d-3975369/lari-terbirit-birit-ini-detik-detik-ott-pegawai-pajak:detiknews lari terbirit-birit, ini detik-detik ott pegawai pajak

internasional.republika.co.id/berita/internasional/eropa/18/07/02/pb8krs330-gengster-melarikan-diri-dari-penjara-pakai-helikopter:gengster melarikan diri dari penjara pakai helicopter

Gambar 4. 25 Link yang Ditemukan Pada Metode Focused Web Crawler

www.cnnindonesia.com/hiburan/20171009154750-234-247181/personel-vixx-29-lari-bawa-obor-olimpiade-di-korea:personel vixx 29 lari bawa obor olimpiade di korea hiburan 8 bulan yang lalu

www.cnnindonesia.com/tv/20180701132616-405-310554/lomba-lari-jelang-asian-games-2018:01:21 lomba lari jelang asian games 2018 cnn indonesia news report • 01 july 2018 13:26

twitter.com/intent/tweet?original_referer=www.tribunnews.com/metropolitan/2018/07/07/sandi-ikuti-kegiatan-lari-antam-golden-run-50&text=sandi%20ikuti%20kegiatan%20lari%20%27antam%20golden%20run%205.0%27&url=www.tribunnews.com/metropolitan/2018/07/07/sandi-ikuti-kegiatan-lari

Gambar 4. 26 Link yang Ditemukan Pada Metode Focused Web Crawler dengan Kompresi multilevel

Jika dilihat dari link yang diperoleh pada hasil tanpa kompresi multilevel gambar 4.25 menunjukkan kata lari pada link pertama didapatkan dari kata “terlaris”,

link kedua pada kata “lari” , link ketiga pada kata “melarikan”. Untuk metode kompresi multilevel yaitu gambar 4.26 kata lari pada link pertama dan kedua ditemukan pada kata “lari” sedangkan untuk hasil ketiga ditemukan pada kata “lari” dan “20lari”. Perbedaan temuan link dari kata yang sesuai inputan dan kata yang mengandung kata inputan akan ditunjukkan pada tabel 4.7 dibawah. Percobaan yang digunakan adalah percobaan seperti yang diambil pada tabel 4.5 yaitu Tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih banyak dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) urutan percobaan juga masih disesuaikan dengan tabel sebelumnya.

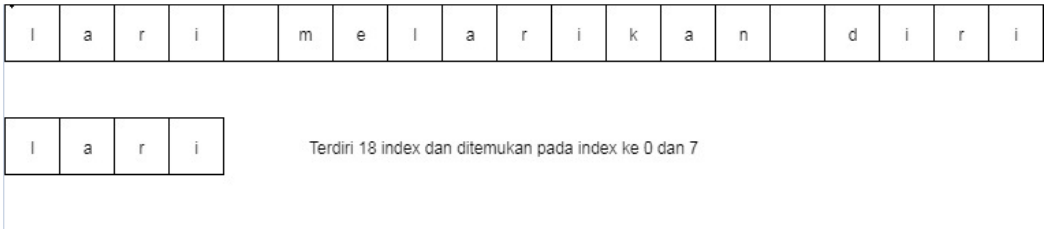
Tabel 4. 7 Tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) dan Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Sesuai Inputan dan Inputan Ada di Dalam Kata Lain (Link Hasil Temuan FWC Lebih Banyak) Inputan Satu Kata Dasar

No Percobaan	**Kata**	**Sesuai dengan Inputan**		**Inputan ada di kata lain**	
		FWC	**FWCM**	**FWC**	**FWCM**
27	nasi	38	43	15020	2
14	ikan	215	211	3091	8
3	alam	118	105	2265	16
36	seni	136	83	815	1
35	sehat	636	629	523	75
12	hilang	422	454	500	53
19	kerja	418	432	341	27
22	lari	81	76	178	1
24	masuk	1377	1318	205	0
10	hantu	92	95	57	1

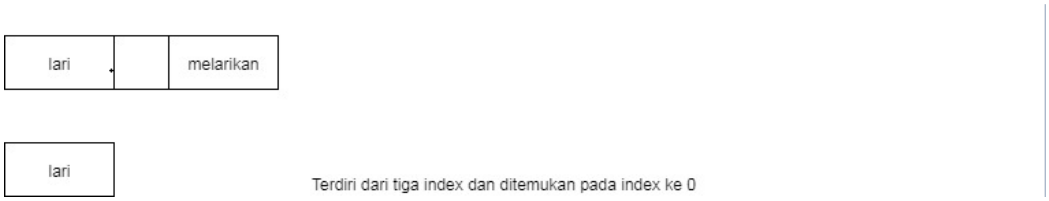
Dari rekap hasil perbedaan dan pengamatan data hasil link temuan, pada metode tanpa kompresi multilevel akan memiliki hasil yang lebih banyak pada kata dasar inputan yang sering muncul dan kata dasar yang ada pada kata lain. Selain itu kata dasar yang memiliki bentuk berimbuhan yang banyak dan sering digunakan dalam bahasa juga mempengaruhi jumlah hasil link pencarian. Ini terjadi karena pencarian dengan menggunakan KMP di dalam metode sebelum kompresi multilevel dilakukan dengan

melakukan pencocokan index untuk setiap huruf inputan kata dengan huruf didalam file kumpulan link, visualisasi proses ditunjukkan di dalam gambar 4.27 .

Sedangkan didalam pencarian KMP pada metode yang menggunakan kompresi multilevel ialah dengan mencocokkan nilai index kata inputan setelah dilakukan proses encode dari kamus kata yang telah dibuat sebelumnya dengan file kumpulan link yang terencode sesuai dengan nilai index pada kamus kata. Karena proses pembuatan kamus kata dilakukan dengan cara melakukan tokenizing kata (spasi dan beberapa karakter yang ditentukan sebelumnya seperti tanda -/?$#@! dll) maka KMP akan melakukan pencocokan sesuai dengan kamus kata yang ada yang berupa kata atau kumpulan angka dan bukan lagi huruf. Gambar 4.28 menunjukkan cara pencarian index KMP di dalam metode kompresi multilevel.



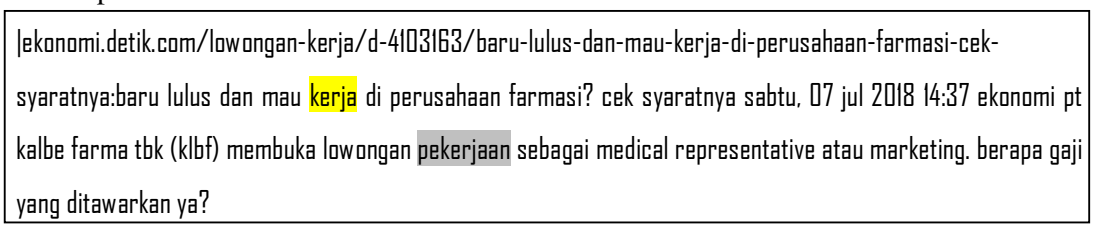
Gambar 4. 27 Pencarian KMP pada Metode Focused Web Crawler tanpa Kompresi multilevel



Gambar 4. 28 Pencarian KMP pada Metode Focused Web Crawler dengan Kompresi multilevel

Pada hasil pencarian link pada metode kompresi multilevel masih menunjukkan inputan kata ada didalam kata lain jika dilihat dari tabel 4.7 sebelumnya. Padahal jika dilihat dari konsep pencarian seharusnya hanya pada index yang sesuai saja maka link akan ditemukan. Setelah di lakukan pengamatan,ada inputan kata yang mengandung kata lain. Hal ini terjadi karena adanya index kata lain yang ada didalam 1 link dan name tag dengan index kata yang mengandung kata inputan(dua index kata berbeda namun jika index kata inputan ditemukan dalam satu link maka index kata lain akan ikut diambil). Pada gambar 4.29 dibawah merupakan contoh kata inputan yang ditemukan

namun dalam link temuan mengandung index kata lain yang kebetulan mengandung kata inputan.



Gambar 4. 29 Penemuan Kata Inputan di Dalam Kata Lain pada Metode Focused Web Crawler dengan Kompresi multilevel Inputan Satu Kata

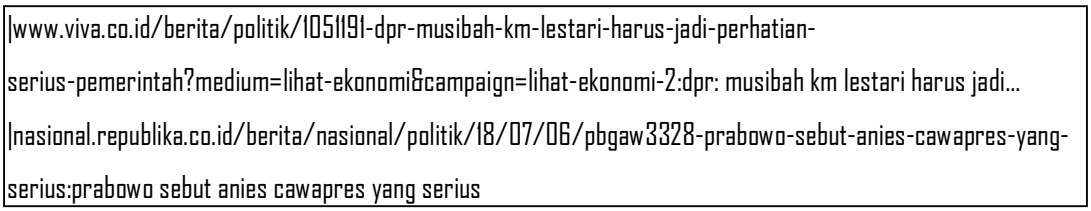
Jadi KMP berhasil hanya untuk pencarian index kata inputan namun karena dalam satu link dan name tag kedua index disimpan.

Pada hasil percobaan ada 10 percobaan yang menunjukkan metode Kompresi multilevel memiliki hasil temuan link lebih banyak daripada metode sebelumnya. Untuk menunjukkan rekap akan ditunjukkan pada tabel 4.8 dibawah. Untuk percobaan yang diambil adalah sama seperti tabel 4.6 yaitu tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih sedikit dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) dimana urutan masih disamakan dengan tabel sebelumnya.

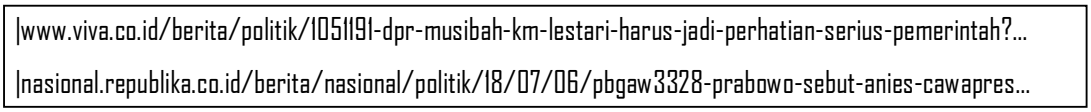
Tabel 4. 8 Tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) dan Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Sesuai Inputan dan Inputan Ada di Dalam Kata Lain (Link Hasil Temuan FWC Lebih Sedikit) Inputan Satu Kata Dasar

No Percobaan	Kata	Sama dengan Inputan		Inputan ada di kata lain	
		FWC	FWCM	FWC	FWCM
30	olahraga	2463	2659	521	37
17	juara	582	578	93	30
21	korban	1026	1076	141	49
37	serius	78	84	9	0
33	rumah	995	1068	208	69
20	konten	71	103	0	0
32	perang	1102	1142	193	38
23	listrik	365	385	32	19
26	murah	116	135	21	11

Pengamatan data dilakukan dan beberapa hasil link dari metode dengan kompresi multilevel menunjukkan ada kata hasil proses decode dalam satu link yang seharusnya di tulis didalam satu baris menjadi dua baris. Hasil pengamatan di tunjukkan pada gambar 4.30 dan gambar 4.31 dibawah.

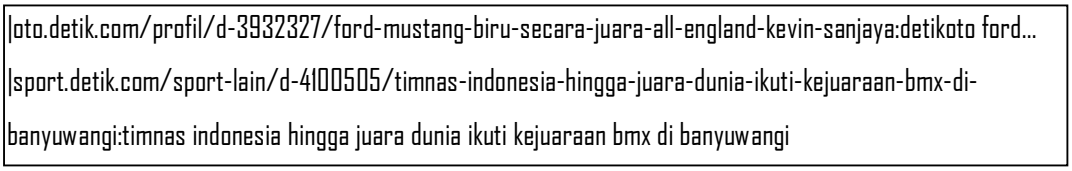


Gambar 4. 30 Penemuan Link di metode kompresi multilevel pada pencarian kata "serius"



Gambar 4. 31 Penemuan Link tanpa metode kompresi multilevel pada pencarian kata "serius"

Dari hasil pengamatan sebetulnya link yang dihasilkan sama namun ada penulisan di dalam file text menjadi berbeda baris sehingga menyebabkan jumlah baris link yang dihasilkan berbeda. Tidak semua hasil link pada metode kompresi multilevel mengalami penulisan berbeda baris, pada kasus diatas yang menyebabkan terjadi penulisan dalam satu link menjadi dua baris adalah kata "serius". Seharusnya kata serius yang menjadi baris baru merupakan kesatuan dari baris sebelumnya. Sehingga seharusnya pada metode kompresi multilevel menghasilkan 1 link dalam satu baris menjadi 1 link dan dua baris. Untuk hasil lainnya misal inputan kata "juara" yang membuat penulisan hasil link terpisah adalah kata "banyuwangi" dan "belanda" dimana ini juga sama seperti kasus penemuan kata juara. Gambar 4.32 dan 4.33 merupakan hasil pencarian.



Gambar 4. 32 Penemuan Link di metode kompresi multilevel pada pencarian kata "juara"

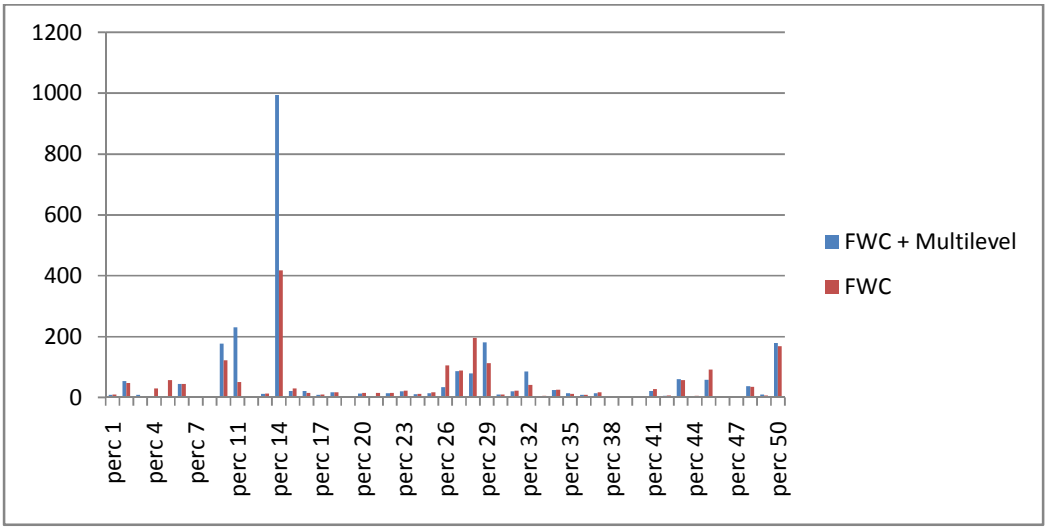
|oto.detik.com/profil/d-3932327/ford-mustang-biru-milik-juara-all-england-kevin-sanjaya:detikoto ford...

|sport.detik.com/sport-lain/d-4100505/timnas-indonesia-hingga-juara-dunia-ikuti-kejuaraan-bmx-di...

Gambar 4. 33 Penemuan Link tanpa metode kompresi multilevel pada pencarian kata "juara"

2. Kata Berimbuhan

Dari hasil percobaan pencarian yang telah dilakukan untuk satu kata inputan yakni dengan menggunakan kata dasar yang berimbuhan dengan 50 percobaan kata berimbuhan didapatkan data seperti grafik pada gambar 4.34 dibawah ini:



Gambar 4. 34 Grafik Pencarian Link Inputan Satu Kata Berimbuhan.

Dari percobaan yang dilakukan pada 50 kata berimbuhan masih ada perbandingan hasil link yang didapat, namun perbandingan jarak yang dihasilkan tidak terlalu banyak seperti pada inputan kata dasar. Ini terjadi karena dengan inputan yang lebih spesifik dengan pencarian KMP pada metode sebelumnya maka kata yang ditemukan akan lebih sedikit. Sedangkan untuk pencarian KMP dengan metode kompresi multilevel karena index berupa kata dalam kamus dan merupakan kata hasil proses tokenizing maka hasil link yang diperoleh adalah sesuai dengan index yang dicari yang merupakan kata berimbuhan bukan huruf.

Tabel 4. 9 Tabel Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih banyak dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Berimbuhan

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
27	pekerja	196	79	117
25	menghilang	106	34	72
5	ingatan	57	0	57
44	perubahan	92	58	34
4	hargai	30	2	28
20	menceritakan	15	2	13
14	lengkapi	30	22	8
40	perbaikan	28	21	7
22	menepi	23	20	3
24	mengatur	17	14	3
26	menunggu	89	86	3

Pada tabel 4.9 diatas menunjukkan percobaan dimana metode tanpa kompresi multilevel menghasilkan jumlah lebih banyak. Dari keseluruhan percobaan yang dilakukan ada pula percobaan yang menunjukkan banyaknya halaman metode kompresi multilevel mendapatkan lebih banyak link. Pada tabel 4.10 dibawah ini menunjukkan beberapa percobaan yang hasil dengan metode kompresi multilevel lebih banyak.

Tabel 4. 10 Tabel Perbandingan Jumlah Link Temuan Metode Focused Web Crawler (FWC) lebih sedikit dari Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Berimbuhan

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
13	kesehatan	418	994	576
10	kekuasaan	50	230	180
28	pekerjaan	113	181	68
9	kehilangan	123	177	54

31	pemandangan	42	85	43
50	travelling	168	179	11
3	dilaut	2	9	7
2	berenang	47	53	6
15	melanggar	15	21	6
42	peringatan	57	60	3

Untuk hasil pencarian KMP pada kompresi multilevel memiliki hasil yang lebih banyak ternyata masih sama seperti kasus pada pencarian kata dasar, ada link temuan yang seharusnya menjadi 1 baris link dia berubah menjadi dua baris keluaran sehingga dianggap menjadi dua link yang berbeda oleh program pengecekan. Namun sama seperti kasus sebelumnya tidak semua link yang dihasilkan merupakan link yang menjadi dua link. Hanya beberapa hasil saja yang terjadi, setelah dilakukan pengamatan terhadap cara kerja pencarian KMP ternyata pencarian dengan menggunakan KMP akan terjadi kesalahan saat index dari karakter pembatas antar link yaitu "|" merupakan bilangan ganjil. Ini terjadi karena dalam pembacaan byte untuk kompresi multilevel membutuhkan 2 byte. Jika index ganjil maka byte yang diperoleh juga hanya satu byte. Dalam sistem penanganan perolehan index yang hanya 1 byte hanya dengan menambahkan 1 index secara manual sehingga yang terjadi karakter untuk pembatas link menjadi salah indeks dan membaca karakter lain sehingga hasil decode dari rentang nilai batasan menjadi bermasalah. Itulah mengapa untuk index dari karakter pembatas link jika semua merupakan bilangan genap maka link akan secara utuh didapatkan dan bisa ditampilkan dalam satu baris pada hasil output.

3. Analisa Presisi dan Recall 1 Kata Inputan

Untuk perhitungan presisi dan recall pada satu inputan kata karena pada pengamatan hasil yang sama persis dengan inputan dan hasil kata yang ada didalam kata lain menjadi unsur yang juga mempengaruhi hasil link yang didapat, maka presisi dan recall akan ditampilkan menurut kata yang sama persis dan kata yang ada di dalam kata lain. Untuk perhitungan presisi dan recall pada tiap percobaan kata dasar maupun berimbuhan akan ditunjukkan pada tabel 4.11 dibawah. Percobaan yang diambil adalah percobaan dimana metode focused web crawler tanpa kompresi multilevel memiliki

lebih banyak hasil pencarian (tabel 4.5) dan percobaan dimana focused web crawler dengan kompresi multilevel memiliki lebih banyak hasil pencarian (tabel 4.6).

Tabel 4. 11 Tabel Perbandingan Jumlah Nilai Presisi dan Recall Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Dasar

No Percobaan	Kata	Presisi (sesuai inputan)		Presisi (ada pada kata lain)		Recall (sesuai inputan)		Recall (ada pada kata lain)	
		FWC	FWCM	FWC	FWCM	FWC	FWCM	FWC	FWCM
27	nasi	0.002	1	0.98	0.047	0.00186	0.73685	0.00328	0.00015
14	ikan	0.062	0.898	0.893	0.034	0.01055	0.15164	0.01608	0.00061
3	alam	0.044	1	0.848	0.152	0.00579	0.11112	0.008	0.00122
36	seni	0.138	0.344	0.824	0.004	0.00667	0.03998	0.00632	0.00008
35	sehat	0.58	0.891	0.477	0.106	0.0312	0.02566	0.04793	0.00572
12	hilang	0.499	0.981	0.591	0.114	0.0207	0.02453	0.0346	0.00404
19	kerja	0.566	0.977	0.461	0.061	0.02051	0.01673	0.03292	0.00206
22	lari	0.291	1	0.64	0.013	0.00397	0.00873	0.00579	0.00008
24	masuk	0.857	0.936	0.128	0	0.06755	0.01006	0.10043	0
10	hantu	0.617	1	0.383	0.011	0.00451	0.0028	0.00724	0.00008
30	olahraga	0.825	0.756	0.174	0.011	0.12083	0.20262	0.02556	0.00282
17	juara	0.9	0.726	0.144	0.038	0.02855	0.04404	0.00456	0.00229
21	korban	0.928	0.903	0.127	0.041	0.05033	0.08199	0.00692	0.00373
37	serius	0.886	0.506	0.102	0	0.00383	0.0064	0.00044	0
33	rumah	0.879	0.915	0.184	0.059	0.04881	0.08138	0.0102	0.00526
20	konten	0.986	1	0	0	0.00348	0.00785	0	0
32	perang	0.881	0.894	0.154	0.03	0.05406	0.08702	0.00947	0.0029
23	listrik	0.976	0.965	0.086	0.048	0.01791	0.02934	0.00157	0.00145
26	murah	0.906	1	0.164	0.081	0.00569	0.01029	0.00103	0.00084

Dari 19 sampel percobaan yang diambil dari 36 percobaan dengan melihat hasil perhitungan untuk presisi (yang sesuai inputan) metode kompresi multilevel memiliki 14 percobaan dengan nilai yang lebih tinggi daripada metode tanpa kompresi multilevel. Hal ini terjadi karena pada metode kompresi multilevel halaman tidak sesuai yang ditemukan lebih sedikit daripada metode tanpa kompresi multilevel.

Halaman tidak sesuai pada metode tanpa kompresi multilevel adalah link yang didapat ketikan inputan ada didalam kata lain, misalkan pada pencarian kata “kerja” maka link yang mengandung kata kerja seperti pekerja, bekerja juga akan dikumpulkan.

Untuk perolehan presisi (ada pada kata lain) metode tanpa kompresi multilevel memiliki nilai presisi yang lebih banyak daripada metode dengan kompresi multilevel, dari 19 percobaan sampel yang diambil 18 percobaan metode tanpa kompresi multilevel lebih unggul.

Untuk perhitungan recall dibedakan menjadi dua , yaitu untuk recall (sesuai inputan) dari 19 sampel percobaan 15 percobaan metode dengan kompresi multilevel memiliki nilai yang lebih tinggi daripada metode tanpa kompresi multilevel. Ini terjadi karena jumlah halaman sesuai yang ditemukan lebih banyak, selain itu jumlah keseluruhan link yang dikumpulkan pada metode dengan kompresi multilevel lebih sedikit dripada metode tanpa kompresi multilevel. Hal ini menyebabkan pembagi nilai link yang sesuai menjadi semakin kecil dan menghasilkan nilai recall yang lebih besar. Sedangkan untuk recall (ada pada kata lain) dari sampel percobaan , 18 percobaan pada metode tanpa kompresi multilevel memiliki nilai recall yang lebih banyak daripada metode tanpa kompresi multilevel.

Pada hasil nilai recall untuk kata inputan ada pada kata lain menunjukkan bahwa hasil link sesuai yang didapatkan lebih banyak daripada metode tanpa kompresi multilevel.Walaupun total link hasil keseluruhan pada percobaan metode tanpa multilevel memiliki nilai yang banyak namun link sesuai yang dihasilkan masih memiliki nilai yang lebih tinggi daripada nilai recall pada metode dengan kompresi multilevel.Untuk hasil perhitungan presisi dan recall kata berimbuhan akan di tampilkan pada tabel 4.12 dibawah. Percobaan yang diambil adalah percobaan dimana metode focused web crawler tanpa kompresi multilevel memiliki lebih banyak hasil pencarian (tabel 4.9) dan percobaan dimana focused web crawler dengan kompresi multilevel memiliki lebih banyak hasil(tabel 4.10).

Tabel 4. 12 Tabel Perbandingan Jumlah Nilai Presisi dan Recall Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Satu Kata Berimbuhan

No Percb	Kata	Presisi (sesuai inputan)		Presisi (ada pada kata lain)		Recall (sesuai inputan)		Recall (ada pada kata lain)	
		FWC	FWCM	FWC	FWCM	FWC	FWCM	FWC	FWCM
27	pekerja	0.398	1	0.597	0	0.03833	0.0301	0.05749	0
25	menghilang	0.321	1	0.726	0.206	0.01671	0.01295	0.03784	0.00267

5	ingatan	0	0	0.86	0	0	0	0.02408	0
44	perubahan	0.663	1	0.413	0.017	0.02998	0.0221	0.01867	0.00038
4	hargai	0.033	1	0.933	0	0.00049	0.00076	0.01376	0
20	menceritakan	0.133	1	0.8	0	0.00098	0.00076	0.0059	0
14	lengkapi	0.667	1	0.533	0.364	0.00983	0.00838	0.00786	0.00305
40	perbaikan	0.964	1	0	0	0.01327	0.008	0	0
22	menepi	0.87	1	0.435	0.4	0.00983	0.00762	0.00491	0.00305
24	mengatur	0.824	1	0.118	0	0.00688	0.00533	0.00098	0
26	menunggu	0.921	1	0.247	0.186	0.04029	0.03276	0.01081	0.0061
30	pemadaman	0.87	1	0.435	0.4	0.00983	0.00762	0.00491	0.00305
13	kesehatan	0.983	0.431	0.033	0.006	0.20197	0.16305	0.00688	0.00229
10	kekuasaan	0.98	0	0.16	0.035	0.02408	0	0.00393	0.00305
28	pekerjaan	0.956	0.448	0.115	0.011	0.05307	0.03086	0.00639	0.00076
9	kehilangan	0.976	0.412	0.081	0	0.05897	0.02781	0.00491	0
31	pemandangan	0.952	0.471	0.024	0	0.01966	0.01524	0.00049	0
50	travelling	0.994	0.933	0	0	0.08206	0.06362	0	0
3	dilaut	0.5	0	0	0	0.00049	0	0	0
2	berenang	0.979	1	0	0.019	0.0226	0.02019	0	0.00038
15	melanggar	0.933	1	0	0	0.00688	0.008	0	0
42	peringatan	0.947	1	0.035	0	0.02654	0.02286	0.00098	0

Untuk perhitungan presisi dan recall pada 1 kata berimbuhan dilakukan dengan mengambil 22 sampel percobaan. Sampel yang diambil merupakan percobaan yang memiliki rentang yang cukup banya dan sudah dijelaskan pada pembahasan sebelumnya. Untuk perhitungan presisi (sesuai inputan) metode dengan kompresi multilevel memiliki rata-rata nilai presisi lebih tinggi daripada metode tanpa kompresi multilevel. Dari 22 percobaan 15 percobaan memiliki nilai presisi yang lebih tinggi nilai presisi 1 ada didalam 14 percobaan. Untuk nilai presisi (ada pada kata lain) metode tanpa multilevel memiliki nilai rata-rata yang lebih tinggi ,17 dari 22 percobaan menunjukkan nilai presisi lebih tinggi daripada metode dengan kompresi multilevel. Namun untuk nilai presisi , presisi (sesuai inputan) memiliki nilai rata-rata hampir 1 berarti link yang dihasilkan lebih sesuai dari pada presisi (ada pada kata lain) dimana nilai presisi tidak sampai 1.

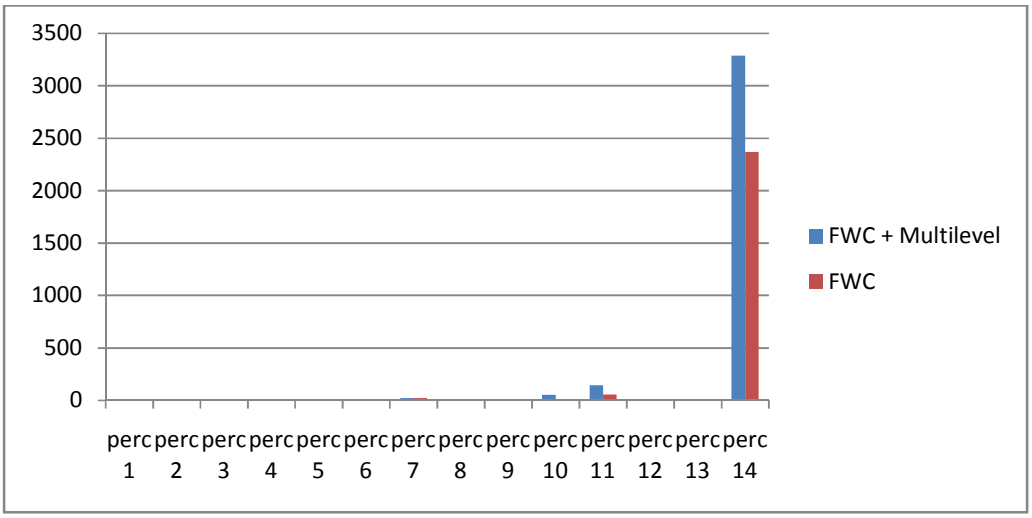
Jika nilai presisi mencapai 1 maka hasil link semua adalah sesuai dan tidak ada hasil yang tidak sesuai dengan inputan. Untuk nilai recall yang dihasilkan baik recall(sesuai inputan) dan recall (ada pada kata lain) metode tanpa kompresi multilevel menunjukkan rata-rata nilai recall yang lebih tinggi. Ini terjadi karena pada keseluruhan hasil link pada metode dengan kompresi multilevel memiliki nilai yang lebih banyak daripada metode tanpa kompresi multilevel. Selain memiliki hasil link yang lebih banyak, yang membuat nilai recall menjadi lebih sedikit adalah karena link yang ditemukan memiliki hasil yang tidak sesuai lebih banyak (pada percobaan kata inputan "kesehatan"). Dalam percobaan metode dengan kompresi multilevel link yang dihasilkan lebih banyak karena kesalahan dalam penulisan link kedalam hasil, dimana 1 link yang seharusnya dianggap 1 baris menjadi 2 baris yang dianggap menghasilkan dua link web. Untuk menjadikan nilai recall lebih baik maka salah satu cara yang bisa dilakukan adalah dengan memperbaiki proses pengambilan hasil link pada metode dengan kompresi multilevel.

4.3.4.1.2 Uji Coba dan Analisa Dua Kata Pencarian

Untuk pengujian nilai presisi dan recall yang kedua adalah dengan menggunakan dua inputan kata pencarian. Dua inputan kata pencarian ini antara lain berupa dua kata dasar yang diinputkan, dua kata yang berupa kata dasar dan berimbuhan, yang ketiga adalah berupa dua kata yang berimbuhan. Pengamatan yang dilakukan sama seperti pada uji coba pertama dimana nilai presisi dan recall akan dibedakan menjadi dua yakni berdasarkan kata yang sama seperti inputan dan berdasarkan kata inputan yang ada didalam kata lain. Untuk skenario pertama dilakukan dengan pencarian dua kata inputan dimana keduanya merupakan kata dasar.

1. Dua Kata Dasar

Pada percobaan dua kata dasar dilakukan 14 percobaan untuk dua kata inputan berupa kata dasar. Inputan dua kata akan dilakukan kepada dua metode pencarian. Dari 14 percobaan tersebut menghasilkan jumlah link pencarian kedua metode yang akan ditunjukkan di dalam grafik pada gambar 4.35 dibawah.



Gambar 4. 35 Grafik Jumlah Halaman Link Inputan 2 Kata Dasar.

Dari 14 percobaan yang dilakukan, 12 percobaan menunjukkan metode focused web crawler tanpa menggunakan kompresi multilevel memiliki hasil link yang lebih banyak. Untuk dua percobaan lain menunjukkan kata yang diinputkan tidak ditemukan oleh algoritma pencarian di dalam history hasil crawl pada 5000 link yang tersimpan. Pada hasil link yang didapat untuk kata inputan yang banyak tersimpan didalam history link penyimpanan akan muncul banyak pula, terlihat seperti percobaan 14 yang sangat jauh dengan percobaan lain karena kata inputan pada percobaan 14 paling banyak disimpan didalam history link. Rincian jumlah link yang ditemukan kedua metode akan ditunjukkan kedalam tabel 4.13 dibawah ini.

Tabel 4. 13 Tabel Jumlah Hasil Link pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Dua Kata Dasar

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
1	rumah sehat	5	0	5
2	udara segar	0	0	0
3	masuk rumah	2	2	0
4	cabang olahraga	1	1	0
5	listrik mati	0	0	0
6	insentif pajak	4	3	1

7	sisa uang	24	23	1
8	aktor hollywood	5	5	0
9	teroris bom	1	1	0
10	rumah zakat	52	52	0
11	orang tewas	144	144	0
12	investasi migas	16	16	0
13	kolesterol naik	2	2	0
14	piala dunia	3326	3289	37

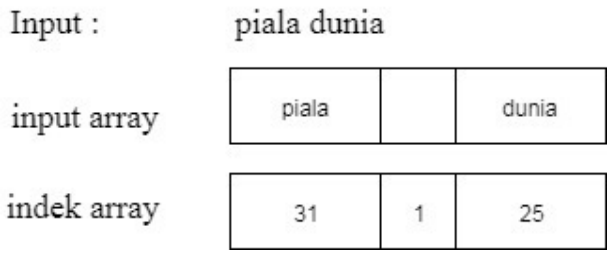
Sama seperti pengujian sebelumnya, pengamatan dilakukan terhadap inputan kata yang sesuai dan inputan kata yang ada didalam kata lain, pengamatan hasil kesamaan kata pada masing-masing percobaan akan ditampilkan pada tabel 4.14 dibawah :

Tabel 4. 14 Tabel Perbandingan Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Dua Kata Dasar

Kata	**FCW**				**FCWM**			
	LB (Sama)	**LB (ada kata inputan)**	**LS (sama)**	**LS (ada kata inputan)**	**LB (Sama)**	**LB (ada kata inputan)**	**LS (sama)**	**LS (ada kata inputan)**
udara segar	0	0	0	0	0	0	0	0
masuk rumah	2	0	0	2	1	0	1	2
rumah zakat	5	36	47	16	0	0	52	52
orang tewas	139	0	5	144	119	0	25	144
berikan sosialisasi	1	0	0	1	0	0	1	1
ikuti kejuaraan	14	0	0	14	0	0	14	14
menangkal radikalisme	5	0	3	8	0	0	7	7
menunggu pesanan	10	0	0	10	10	0	0	10
jenggotnya memutih	2	0	0	2	2	0	0	2
siaran langsung	131	0	14	145	130	0	15	145
untuk kesehatan	6	8	10	8	5	1	12	16
melangkah pasti	12	0	0	12	0	0	13	13

Dari hasil perhitungan hasil link yang sesuai dengan kata inputan maupun kata inputan ada didalam kata lain pada metode tanpa kompresi multilevel menunjukkan hasil link pencarian masih terdapat inputan kata yang berada didalam kata lain atau mengandung kata lain. Ini terjadi karena pada pencarian dengan algoritma KMP untuk pencocokan huruf tidak ada masalah apabila kata tidak berdiri sendiri atau ada kata maupun huruf lain yang berdekatan karena pengecekan adalah per huruf dari kata inputan. Namun pada kata inputan yang sudah mulai spesifik seperti inputan lebih dari satu kata pencarian KMP juga akan menghasilkan link pencarian yang lebih sedikit karena di dalam kata atau bahasa, pencarian dua kata sekaligus dalam proses akan lebih sedikit daripada pencarian satu kata yang memiliki kemungkinan lebih banyak dia berdiri sendiri sebagai kata ataupun ada pada kata.

Untuk metode focused web crawler dengan kompresi multilevel pencarian link dengan algoritma KMP masih bisa mengeluarkan hasil pencarian yang sesuai walaupun didalam tabel kamus kata untuk dua kata yang diinputkan tidak ada. Hal ini terjadi karena pada proses pembuatan kamus kata, inputan kata dilakukan proses tokenizing yaitu pemisahan kata berdasarkan karakter seperti titik (.), koma (,), spasi ( ) dan karakter lainnya. Cara yang dilakukan agar algoritma pencarian dapat menghasilkan kata inputan yang dicari adalah dengan cara menjadikan kata inputan yang memiliki lebih dari satu kata menjadi array dari index kamus kata. Prosesnya seperti gambar dibawah 4.36 ini :



Gambar 4. 36 Proses Encode Kata Inputan Untuk Pencarian KMP pada Metode Kompresi multilevel

Setelah inputan kata dijadikan array lalu nilai index pada tiap kata akan didapatkan dari proses encode dari kamus kata. Hasil index array dijadikan input kedalam pencarian dengan algoritma pencocokan KMP. Selain itu jika inputan tidak dijadikan sebagai potongan kata maka proses encode kata inputan yang merupakan

kumpulan kata tidak akan berhasil karena tidak ditemukan didalam kamus kata(dilakukan dengan melihat tabel kamus kata kata “piala dunia” tidak terdaftar).

Dari segi kata inputan yang berada didalam kata lain, baik metode focused web crawler tanpa maupun dengan kompresi multilevel untuk jumlah temuan pada tiap percobaan lebih sedikit dibandingkan dengan pengujian pada 1 kata inputan. Selain jumlah temuan dalam satu link yang mengalami pengurangan, semakin sedikit pula percobaan yang menghasilkan kata dengan inputan yang ada didalam kata lain. Dari hasil pengamatan link pencarian yang memiliki inputan yang berada di kata lain kebanyakan selain ada kata lain , kata inputan yang dihasilkan juga kata inputan yang berdekatan dengan karakter-karakter huruf. Gambar 4.37 dan 4.38 dibawah merupakan contoh hasil link yang mengandung inputan tidak sama persis dengan kata inputan.

|www.detik.compialadunia:piala dunianew
|sport.detik.com/sepakbola/indeksfokus/3819/spanyol-out-dari-piala-dunia/berita:02 spanyol out dari piala dunia |sport.detik.com/sepakbola/berita/d-4105877/piala-dunianya-premier-league:1 piala dunianya premier league

Gambar 4. 37 Hasil Penemuan Link Inputan yang Ada didalam Kata Lain pada Metode Focused Web Crawler tanpa Kompresi multilevel Inputan Dua Kata Dasar

|sport.detik.com/sepakbola/liga-spanyol/d-4105984/hazard-diprospek-madrid-usai-piala-dunia:hazard diprospek madrid usai piala dunia?
|www.detik.comtagpiala-dunia-2018:piala dunia 2018

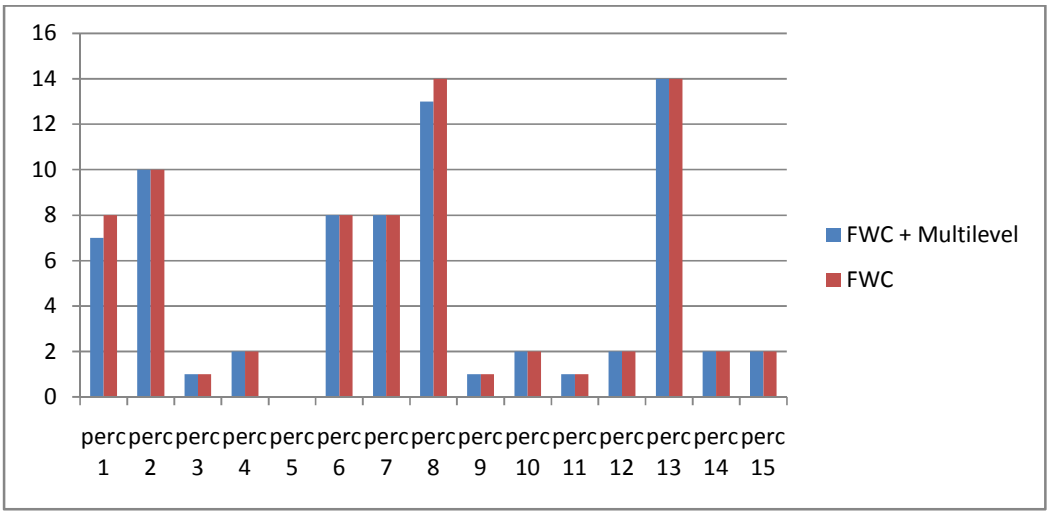
Gambar 4. 38 Hasil Penemuan Link Inputan yang Ada didalam Kata Lain pada Metode Focused Web Crawler dengan Kompresi multilevel Inputan Dua Kata Dasar

Untuk hasil pencarian link pada percobaan kedua rata-rata metode tanpa kompresi multilevel memiliki hasil pencarian link yang lebih banyak dari metode dengan kompresi multilevel. Selain karena pada metode tanpa kompresi multilevel memiliki hasil kata yang mirip dengan inputan bisa ditampilkan dalam beberapa percobaan, dengan metode kompresi multilevel masih terjadi kesalahan dalam hal pengambilan batasan link di dalam history link yang telah terencode didalam data. Contoh percobaan yang mengalami perbedaan dikarenakan kesalahan pengambilan link

adalah pada percobaan ke 4 yaitu kata inputan “cabang olahraga”. Untuk percobaan kata inputan lainnya tidak ada masalah.

2. Dua Kata Berimbuhan

Untuk percobaan selanjutnya dua kata inputan adalah pencarian dengan dua kata yang merupakan perpaduan antara kata dasar dan kata berimbuhan. Dalam percobaan ini akan dilihat apakah pencarian metode focused web crawler dengan kompresi multilevel dapat menghasilkan hasil ataukah tidak. Hasil link temuan percobaan akan dimasukkan kedalam grafik pada gambar 4.39 dibawah.



Gambar 4. 39 Grafik Jumlah Link yang diperoleh pada Metode Focused Web Crawler dengan dan tanpa Mulitilevel Kompresi Inputan Dua Kata Berimbuhan.

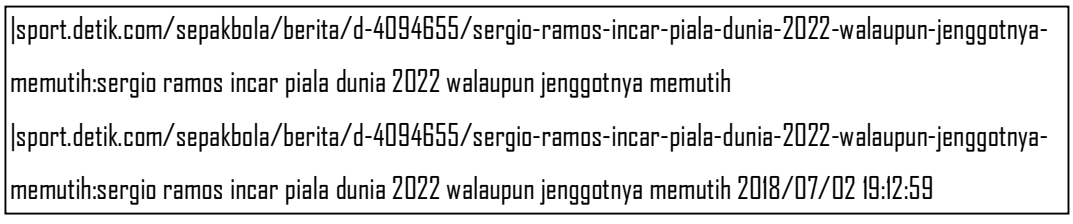
Dari grafik yang ada, perbedaan hasil link yang ditemukan pada percobaan dari kedua metode hanyalah 1 link dimana focused web crawler tanpa metode kompresi multilevel memiliki hasil yang lebih banyak. Berikut kata yang dimasukkan kedalam tiap-tiap percobaan akan ditampilkan kedalam tabel 4.15 dibawah:

Tabel 4. 15 Tabel Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Dua Kata Berimbuhan

Kata	FCW					FCWM				
	Hasil Link	LB (sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)	Hasil Link	LB (sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)
menangkal radikalisme	8	5	0	3	8	7	0	0	7	7

menunggu pesanan	10	10	0	0	10	10	10	0	0	10
berikan sosialisasi	1	1	0	0	1	1	0	0	1	1
terakhir pencarian	2	2	0	0	2	2	2	0	0	2
berikan sosialisasi	0	0	0	0	0	0	0	0	0	0
hadiri pertemuan	8	8	0	0	8	8	8	0	0	8
sebelum menyelam	8	8	0	0	8	8	8	0	0	8
mengulik konsolidasi	14	14	0	0	14	13	13	0	0	13
merespons pelemahan	1	1	0	0	1	1	1	0	0	1
ratusan tahanan	2	2	0	0	2	2	2	0	0	2
pastikan persiapan	1	1	0	0	1	1	1	0	0	1
melakukan penyisiran	2	2	0	0	2	2	2	0	0	2
ikuti kejuaraan	14	14	0	0	14	14	0	0	14	14
jenggotnya memutih	2	2	0	0	2	2	2	0	0	2
atasi kekurangan	2	2	0	0	2	2	2	0	0	2

Dari hasil percobaan dengan kata inputan yang tidak memiliki kata dasar menyebabkan hasil pencarian string diantara kedua metode semakin menunjukkan hasil link pencarian sama banyak. Untuk mengamati hasil link pencarian yang hanya berbeda satu link pencarian dilakukan pengamatan terhadap hasil link dari percobaan yang telah dilakukan. Dari hasilnya diketahui semua halaman yang dihasilkan pada kedua metode merupakan link yang sama dan sama-sama menghasilkan link yang mengandung kata inputan dan satu yang berbeda merupakan link yang tidak sesuai dengan kata inputan. Berikut hasil pencarian pada kedua metode akan ditampilkan pada gambar 4.40 dan 4.41 dibawah:



Gambar 4. 40 Hasil Pencarian Metode tanpa Mulitilevel Kompresi Inputan Dua Kata Berimbuhan

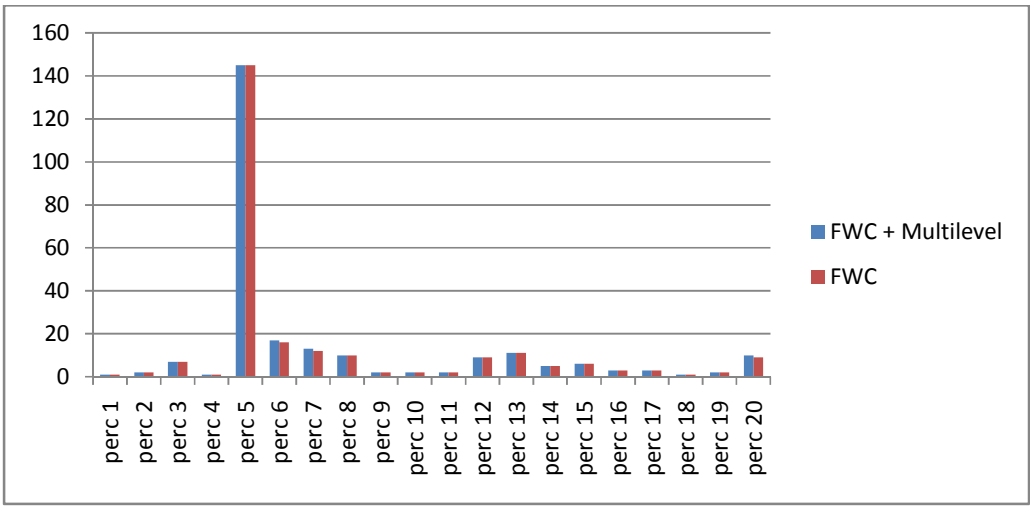
|sport.detik.com/sepakbola/berita/d-4094655/sergio-ramos-incar-piala-dunia-2022-walaupun-jenggotnya-memutih:sergio ramos incar piala dunia 2022 walaupun jenggotnya memutih
|sport.detik.com/sepakbola/berita/d-4094655/sergio-ramos-incar-piala-dunia-2022-walaupun-jenggotnya-memutih:sergio ramos incar piala dunia 2022 walaupun jenggotnya memutih 2018/07/02 19:12:59

Gambar 4. 41 Hasil Pencarian Metode dengan Mulitilevel Kompresi Inputan Dua Kata Berimbuhan

Pada percobaan menggunakan dua kata inputan ini untuk inputan pencarian dengan menggunakan algoritma pencarian KMP sama seperti percobaan sebelumnya. Yang berbeda adalah jenis kata inputan dan ini berpengaruh pada hasil link temuan. Semakin spesifik (adanya imbuhan) maka semakin jarang kata tersebut akan ditemui dekat dengan kata lain sehingga hasil link yang didapatkan semakin sesuai inputan. Untuk metodefocused web crawler dengan menggunakan kompresi multilevel tidak mengalami masalah karena setiap kata inputan yang dimasukkan akan dicari nilai index kata didalam kamus kata dan akan dicari dengan pencarian KMP untuk masing-masing nilai index inputan kata.

3. Kata Dasar dan Kata Berimbuhan

. Pada percobaan dua kata inputan yang ketiga dilakukan sebanyak 20 percobaan dengan kata dasar dan kata berimbuhan. Percobaan dilakukan dengan 20 macam inputan, dari 20 jenis inputan jumlah hasil link yang dihasilkan pada masing-masing metode akan ditampilkan pada grafik gambar 4.42 dibawah:



Gambar 4. 42 Grafik Jumlah Link yang diperoleh pada Metode Focused Web Crawler dengan dan tanpa Mulitilevel Kompresi Inputan Dua Kata (Dasar dan Berimbuhan).

Untuk detail percobaan akan di tampilkan pada tabel 4.16 dibawah ini :

Tabel 4. 16 Tabel Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Dua Kata (Dasar dan Kata Berimbuhan)

Kata	FCW					FCWM				
	Hasil link	LB (sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)	Hasil link	LB (sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)
amankan wilayah	1	0	1	1	0	1	0	1	1	0
untuk pertumbuhan	2	2	0	0	2	2	2	0	0	2
penerbangan ekstra	7	7	0	0	7	7	7	0	0	7
balas ucapan	1	1	0	0	1	1	1	0	0	1
siaran langsung	145	131	0	14	145	145	130	0	15	145
untuk kesehatan	16	6	8	10	8	17	5	1	12	16
melangkah pasti	12	12	0	0	12	13	0	0	13	13
yang diminati	10	10	0	0	10	10	10	0	0	10
salurkan beasiswa	2	2	0	0	2	2	2	0	0	2
pasca akuisisi	2	0	2	2	0	2	0	0	2	2
meminimalisir nyeri	2	2	0	0	2	2	0	0	2	2
manfaat tidur	9	8	1	1	8	9	7	0	2	9
pemilihan umum	11	11	0	0	11	11	11	0	0	11
tetapkan pengganti	5	4	0	1	5	5	0	0	5	5
negara terkuat	6	4	0	2	6	6	4	0	2	6
penguatan ekonomi	3	3	0	0	3	3	1	0	2	3
kembali ditemukan	3	3	0	0	3	3	0	0	3	3
alat pertanian	1	1	0	0	1	1	1	0	0	1
percepat proses	2	2	0	0	2	2	2	0	0	2
alat pertanian	1	1	0	0	1	1	1	0	0	1

Hasil pada kata dasar dan berimbuhan untuk jumlah link yang ditemukan pada dua percobaan yang menunjukkan metode tanpa kompresi multilevel memiliki hasil pencarian link yang lebih banyak ternyata masih ada kesalahan dalam pengambilan link url . Gambar 4.43 dan 4.44 menunjukkan hasil link yang diperoleh. Dari pengamatan hasil pencarian KMP penyebab sama seperti percobaan sebelumnya pada saat hasil index pencarian dari pembatas link "|" memiliki indek ganjil maka hasil url yang ditemukan akan bermasalah sehingga mengakibatkan kesalahan hasil pencarian pada metode kompresi multilevel.

|mailto:?to=&subject=menonton laga sepak bola baik untuk kesehatan mental&body=:mail

|food.detik.com/info-kuliner/d-4105684/bukan-untuk-kesehatan-ini-alasan-utama-sophia-latjuba-jadi-vegan:info kuliner bukan untuk kesehatan, ini alasan utama sophia latjuba jadi vegan

Gambar 4. 43 Hasil pencarian dengan metode tanpa kompresi multilevel Inputan Dua Kata (Dasar dan Kata Berimbuhan)

|mailto:?to=&subject=menonton laga sepak bola baik untuk kesehatan mental&body=:mail

utama-sophia-latjuba-jadi-vegan:info kuliner bukan untuk kesehatan, ini alasan utama sophia latjuba jadi vegan

Gambar 4. 44 Hasil pencarian dengan metode kompresi multilevel Inputan Dua Kata (Dasar dan Kata Berimbuhan)

Kata pencarian selalu ditemukan dengan benar , namun untuk pengambilan 1 link yang utuh yang mengandung kata inputan terkadang bermasalah jika index pembatas yang ditemukan bernilai ganjil. Pada contoh diatas rentang index yang ditemukan dalam link yang tidak komplit adalah 5632639, 5632672, 5632710 dan link berikutnya dengan rentang index 5634579, 5634608, 5634646. Pada kata inputan "melangkah pasti" memiliki rentang index 9006707, 9006736, 9006778. Untuk pencarian kata dengan algoritma KMP dengan inputan dua kata dasar dan berimbuhan masih bisa dilakukan pada kedua metode yang ada.

4. Presisi dan Recall 2 Kata Inputan

Perhitungan nilai presisi dan recall, jenis presisi dan recall akan disamakan seperti pada percobaan 1 kata inputan. Perhitungan 2 kata inputan dengan tiga jenis 2 kata inputan (kata dasar-kata dasar, kata berimbuhan – kata berimbuhan, kata dasar –

kata berimbuhan) akan ditampilkan pada tiga tabel dibawah ini dimana tiap tabel akan menunjukkan presisi dan recall pada masing-masing jenis inputan 2 kata inputan. Tabel pertama 4.17 adalah tabel presisi dan recall 2 inputan kata berupa kata dasar.

Tabel 4. 17 Tabel Presisi dan Recall Inputan Dua Kata Dasar

No Percb	Kata	Presisi (sesuai inputan)		Presisi (ada pada kata lain)		Recall (sesuai inputan)		Recall (ada pada kata lain)	
		FWC	FWCM	FWC	FWCM	FWC	FWCM	FWC	FWCM
1	rumah sehat	0.000	0.000	1.000	0.000	0.00000	0.00000	0.00140	0.00000
2	udara segar	0.000	0.000	0.000	0.000	0.00000	0.00000	0.00000	0.00000
3	masuk rumah	1.000	0.500	0.000	0.000	0.00056	0.00028	0.00000	0.00000
4	cabang olahraga	1.000	0.000	0.000	0.000	0.00028	0.00000	0.00000	0.00000
5	listrik mati	0.000	0.000	0.000	0.000	0.00000	0.00000	0.00000	0.00000
6	insentif pajak	0.750	1.000	0.250	0.000	0.00084	0.00085	0.00028	0.00000
7	sisa uang	1.000	0.000	0.000	0.000	0.00670	0.00000	0.00000	0.00000
8	aktor hollywood	0.600	0.200	0.200	0.000	0.00084	0.00028	0.00028	0.00000
9	teroris bom	1.000	1.000	0.000	0.000	0.00028	0.00028	0.00000	0.00000
10	rumah zakat	0.096	0.000	0.692	0.000	0.00140	0.00000	0.01005	0.00000
11	orang tewas	0.965	0.826	0.000	0.000	0.03881	0.03363	0.00000	0.00000
12	investasi migas	0.438	0.313	0.375	0.375	0.00195	0.00141	0.00168	0.00170
13	kolesterol naik	1.000	1.000	0.000	0.000	0.00056	0.00057	0.00000	0.00000
14	piala dunia	0.713	0.621	0.230	0.199	0.66164	0.57773	0.21385	0.18457

Pada perhitungan presisi dan recall dua kata inputan dengan kata dasar, nilai presisi baik inputan yang sesuai atau yang ada didalam kata lain menunjukkan metode tanpa kompresi multilevel memiliki rata-rata nilai yang lebih tinggi daripada metode dengan kompresi multilevel. Pada jumlah link hasil dengan nilai yang sama pada kedua metode misalkan pada inputan kata “masuk rumah”,”cabang olaraga”, “actor hollywood” presisi metode tanpa kompresi multilevel lebih banyak karena hasil link pada metode kompresi multilevel terkadang masih ada masalah dalam pengambilan link secara utuh walaupun kata inputan bisa didapatkan.

Dalam perhitungan presisi dan recall link yang tidak lengkap dianggap sebagai halaman tidak sesuai, oleh karena itu nilai presisi maupun recall berpengaruh. Untuk kata inputan yang memiliki jumlah link sama dan memiliki nilai presisi dan recall sama

seperti inputan “udara segar”,“listrik mati”,” teroris bom”,dan “kolesterol naik” terjadi karena KMP tidak menemukan inputan didalam history link dan untuk nilai presisi 1 maka metode dengan kompresi multilevel berhasil menemukan kata inputan dan pengambilan link penuh sukses/tidak menemui pembatas link yang indexnya bernilai ganjil. Untuk selanjutnya nilai presisi dan recall untuk dua kata inputan berupa kata dasar dan kata berimbuhan. Selebihnya akan ditunjukkan pada tabel 4.18 dibawah ini.

Tabel 4. 18 Tabel Presisi dan Recall Inputan Dua Kata Berimbuhan

No Perc	**Kata**	**Presisi (sesuai inputan)**		**Presisi (ada pada kata lain)**		**Recall (sesuai inputan)**		**Recall (ada pada kata lain)**	
		FWC	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**
1	menangkal radikalisme	0.625	0.000	0.000	0.000	0.06667	0.00000	0.00000	0.00000
2	menunggu pesanan	1.000	1.000	0.000	0.000	0.13333	0.13699	0.00000	0.00000
3	berikan sosialisasi	1.000	0.000	0.000	0.000	0.01333	0.00000	0.00000	0.00000
4	terakhir pencarian	1.000	1.000	0.000	0.000	0.02667	0.02740	0.00000	0.00000
5	berikan sosialisasi	0.000	0.000	0.000	0.000	0.00000	0.00000	0.00000	0.00000
6	hadiri pertemuan	1.000	1.000	0.000	0.000	0.10667	0.10959	0.00000	0.00000
7	sebelum menyelam	1.000	1.000	0.000	0.000	0.10667	0.10959	0.00000	0.00000
8	mengulik konsolidasi	1.000	1.000	0.000	0.000	0.18667	0.17808	0.00000	0.00000
9	merespons pelemahan	1.000	1.000	0.000	0.000	0.01333	0.01370	0.00000	0.00000
10	ratusan tahanan	1.000	1.000	0.000	0.000	0.02667	0.02740	0.00000	0.00000
11	pastikan persiapan	1.000	1.000	0.000	0.000	0.01333	0.01370	0.00000	0.00000
12	melakukan penyisiran	1.000	1.000	0.000	0.000	0.02667	0.02740	0.00000	0.00000
13	ikuti kejuaraan	1.000	0.000	0.000	0.000	0.18667	0.00000	0.00000	0.00000
14	jenggotnya	1.000	1.000	0.000	0.000	0.02667	0.02740	0.00000	0.00000

	memutih								
15	atasi kekurangan	1.000	1.000	0.000	0.000	0.02667	0.02740	0.00000	0.00000

Pada perhitungan nilai presisi dan recall untuk dua kata inputan berupa kata berimbuhan dari 15 percobaan menunjukkan kedua metode memiliki nilai presisi, baik pada kata inputan sesuai dengan kata inputan. Untuk hasil pada link yang mirip atau kata inputan terdapat kata lain sudah tidak ditemukan. Ini terjadi karena pencarian KMP pada kata yang berimbuhan, hasil kata inputan yang ada di dalam kata lain jarang ditemukan bahkan pada percobaan ini tidak ditemukan lagi. Menemukan huruf kata inputan berimbuhan pada kata lain jarang atau bahkan tidak ada pada penggunaan dalam penulisan atau kaidah bahasa. Untuk metode kompresi multilevel juga memiliki kesamaan model pencarian array index kata, semakin banyak array yang dicari maka pencocokan pada array yang cocok di dalam data akan semakin sedikit karena urutan dari index array.

Pada percobaan pertama yaitu kata inputan "menangkal radikalisme" untuk metode kompresi multilevel menghasilkan link namun presisi dan recall bernilai 0 ini terjadi karena hasil link yang dihasilkan tidak lengkap sehingga dianggap sebagai link tidak sesuai, penyebabnya adalah index pembatas link ditemukan pada index yang memiliki nilai ganjil yaitu rentang "674243, 674260, 674278" , "1548607, 1548624, 1548642". Untuk selanjutnya adalah nilai presisi dan recall pada dua kata inputan berupa kata dasar dan kata berimbuhan.Hasil akan ditampilkan pada tabel 4.19 dibawah ini:

Tabel 4. 19 Tabel Presisi dan Recall Inputan Dua Kata Berupa Kata Dasar dan Kata Berimbuhan

No Perc	**Kata**	**Presisi (sesuai inputan)**		**Presisi (ada pada kata lain)**		**Recall (sesuai inputan)**		**Recall (ada pada kata lain)**	
		FWC	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**
1	amankan wilayah	0.000	0.000	1.000	1.000	0.00000	0.00000	0.00402	0.00397
2	untuk pertumbuhan	1.000	1.000	0.000	0.000	0.00803	0.00794	0.00000	0.00000

3	penerbangan ekstra	1.000	1.000	0.000	0.000	0.02811	0.02778	0.00000	0.00000
4	balas ucapan	1.000	1.000	0.000	0.000	0.00402	0.00397	0.00000	0.00000
5	siaran langsung	0.903	0.897	0.000	0.000	0.52610	0.51587	0.00000	0.00000
6	untuk kesehatan	0.375	0.294	0.500	0.059	0.02410	0.01984	0.03213	0.00397
7	melangkah pasti	1.000	0.000	0.000	0.000	0.04819	0.00000	0.00000	0.00000
8	yang diminati	1.000	1.000	0.000	0.000	0.04016	0.03968	0.00000	0.00000
9	salurkan beasiswa	1.000	1.000	0.000	0.000	0.00803	0.00794	0.00000	0.00000
10	pasca akuisisi	0.000	0.000	1.000	0.000	0.00000	0.00000	0.00803	0.00000
11	meminimalisir nyeri	1.000	0.000	0.000	0.000	0.00803	0.00000	0.00000	0.00000
12	manfaat tidur	0.889	0.778	0.111	0.000	0.03213	0.02778	0.00402	0.00000
13	pemilihan umum	1.000	1.000	0.000	0.000	0.04418	0.04365	0.00000	0.00000
14	tetapkan pengganti	0.800	0.000	0.000	0.000	0.01606	0.00000	0.00000	0.00000
15	negara terkuat	0.667	0.667	0.000	0.000	0.01606	0.01587	0.00000	0.00000
16	penguatan ekonomi	1.000	0.333	0.000	0.000	0.01205	0.00397	0.00000	0.00000
17	kembali ditemukan	1.000	0.000	0.000	0.000	0.01205	0.00000	0.00000	0.00000
18	alat pertanian	1.000	1.000	0.000	0.000	0.00402	0.00397	0.00000	0.00000
19	percepat proses	1.000	1.000	0.000	0.000	0.00803	0.00794	0.00000	0.00000
20	rehabilitasi terumbu	0.778	0.800	0.111	0.100	0.02811	0.03175	0.00402	0.00397

Pada 20 percobaan yang dilakukan ditemukan nilai presisi dan recall pada pencarian link yang sesuai dan pencarian link yang terdapat/mengandung kata lain. Pada hasil percobaan dimana nilai presisi ada/mengandung kata lain seperti kata “amankan wilayah”,” untuk kesehatan”,” pasca akuisisi”,” manfaat tidur”, dan “rehabilitasi terumbu” kata inputan yang ada/mengandung lain lebih kepada hasil kata inputan yang dekat dengan karakter tanda baca seperti tanda titik, koma, tanda tanya. Contoh output terlihat pada gambar 4.45 dibawah :

health.detik.com/read/2018/07/05/080830/4099011/763/arang-aktif-masih-diminati-apa-benar-baik-untuk-kesehatan:arang aktif masih diminati, apa benar baik untuk kesehatan?
food.detik.com/info-kuliner/d-4105684/bukan-untuk-kesehatan-ini-alasan-utama-sophia-latjuba-jadi-vegan:info kuliner bukan untuk kesehatan, ini alasan utama sophia latjuba jadi vegan

Gambar 4. 45 Output Kata Inputan yang Ada/Mengandung Kata Lain pada Inputan Dua Kata (Dasar dan Berimbuhan)

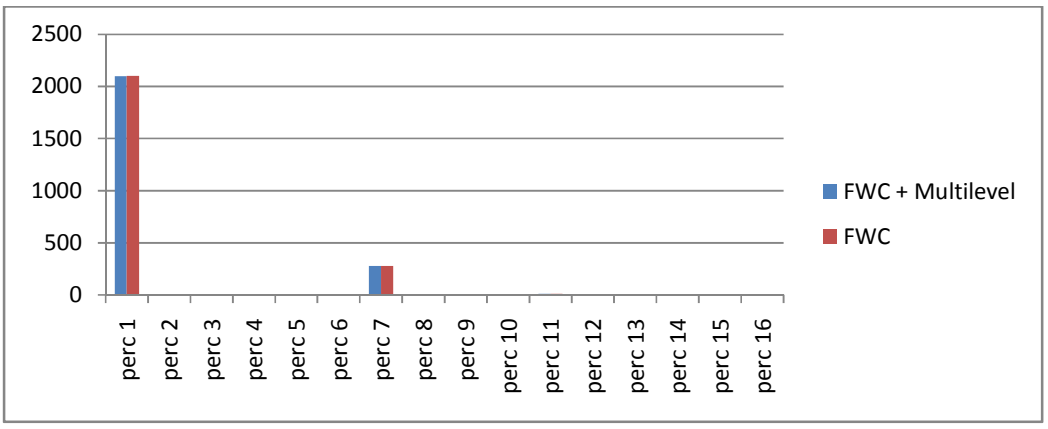
Beberapa hasil nilai presisi dan recall dimana dua metode yang memiliki perbedaan, nilai metode dengan kompresi multilevel memiliki rata-rata nilai yang lebih sedikit hal ini terjadi karena terjadinya kesalahan pengambilan link karena ketidaklengkapan link sehingga dianggap halaman yang tidak sesuai walaupun kata inputan berhasil ditemukan.

4.3.4.1.3 Uji Coba dan Analisa Tiga Kata Pencarian

Untuk uji coba selanjutnya adalah untuk tiga kata inputan pencarian. Dalam uji coba tiga kata percobaan akan dilakukan sama seperti percobaan sebelumnya yakni tiga kata inputan yang berupa kata dasar, dan tiga kata yang terdiri dari kata dasar dan berimbuhan. Untuk percobaan pertama akan dilakukan dengan melakukan percobaan dengan menggunakan tiga inputan yang masing-masing berupa kata dasar.

1. **Tiga Kata Dasar**

Pada percobaan pertama adalah dengan inputan tiga kata dasar. Percobaan yang dilakukan adalah sebanyak 15 percobaan dengan inputan yang berbeda. Dari 15 percobaan tersebut link yang dihasilkan oleh kedua metode focused web crawler ditampilkan pada gambar 4.46 dibawah :



Gambar 4. 46 Grafik Jumlah Link yang diperoleh pada Metode Focused Web Crawler dengan dan tanpa Mulitilevel Kompresi Inputan Tiga Kata Dasar.

Untuk mengetahui selisih hasil perolehan link dari masing-masing percobaan dari ke enam belas percobaan akan ditampilkan pada tabel 4.20.

Tabel 4. 20 Tabel Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Tiga Kata Dasar

No Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
1	terbang tanpa sayap	11	11	0
2	lampu mobil kusam	4	4	0
3	polisi di lampung	2	2	0
4	buka konser celine	6	6	0
5	kpk tahan anggota	1	1	0
6	asian games 2018	278	278	0
7	untuk ibu hamil	3	3	0
8	mantan agen rusia	1	1	0
9	minta proses hukum	4	4	0
10	pensiun dari timnas	13	13	0
11	sahabat putri diana	6	6	0
12	dapat bunga mawar	5	5	0
13	beli lahan tanah	1	1	0
14	harga jual dollar	3	3	0
15	korban bom pasuruan	5	5	0

Dari hasil link yang didapatkan pada kasus ketiga kata inputan adalah kata dasar, hasil antara kedua metode dapat menghasilkan link yang sama banyak. Untuk selanjutnya adalah melakukan pengamatan pada hasil dari link yang ditemukan apakah benar-benar sesuai dengan inputan atau masih ditemukan kesalahan link. Pengamatan dilakukan dengan melihat kemungkinan hasil link pencarian adalah benar-benar sesuai dengan kata inputan atau hasil link merupakan kata inputan yang mengandung kata lain atau berada pada kata lain. Hasil pengamatan akan ditampilkan pada tabel 4.21 dibawah ini:

Tabel 4. 21 Tabel Perbandingan Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Tiga Kata Dasar

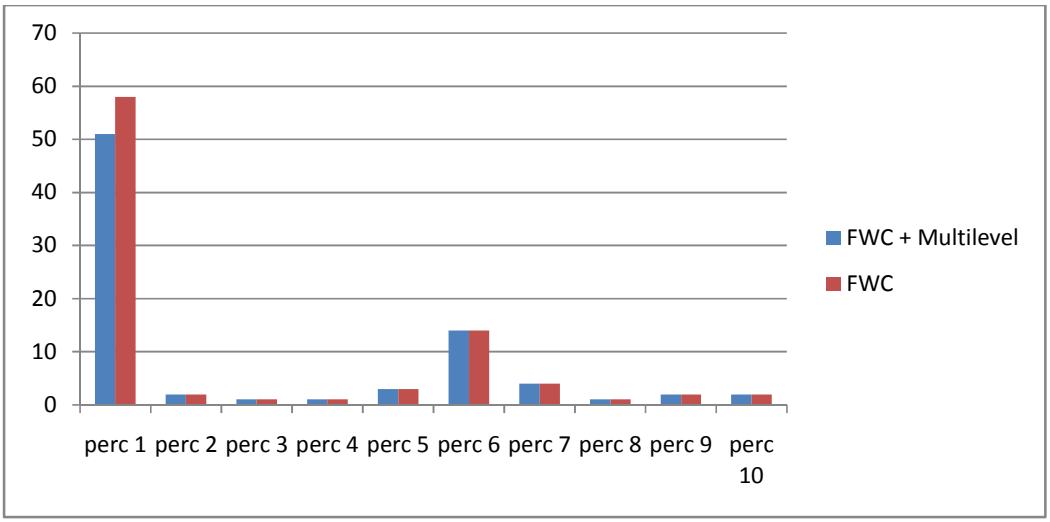
Kata	FCW					FCWM				
	Hasil link	LB (Sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)	Hasil link	LB (Sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)
terbang tanpa sayap	11	11	0	0	11	11	11	0	0	11
lampu mobil kusam	4	3	1	1	3	4	3	1	1	3
polisi di lampung	2	2	0	0	2	2	2	0	0	2
buka konser celine	6	6	0	0	6	6	6	0	0	6
kpk tahan anggota	1	0	1	1	0	1	0	1	1	0
asian games 2018	278	185	78	93	200	278	159	50	119	228
untuk ibu hamil	3	3	0	0	3	3	0	0	3	3
mantan agen rusia	1	1	0	0	1	1	1	0	0	1
minta proses hukum	4	4	0	0	4	4	4	0	0	4
pensiun dari timnas	13	13	0	0	13	13	13	0	0	13
sahabat putri diana	6	6	0	0	6	6	0	0	6	6
dapat bunga mawar	5	5	0	0	5	5	5	0	0	5
beli lahan tanah	1	1	0	0	1	1	1	0	0	1
harga jual dollar	3	3	0	0	3	3	0	0	3	3
korban bom pasuruan	5	5	0	0	5	5	5	0	0	5

Dari 15 percobaan yang dilakukan ternyata hasil pencarian masih menunjukkan adanya kata inputan yang berada atau mengandung kata lain yakni terjadi pada percobaan 2 dan percobaan 5. Kata inputan yang ditemukan pada dua percobaan ini sama-sama mengandung karakter yaitu " : ". Pada percobaan inputan "asian games 2018" jumlah link yang ditemukan pada metode dengan kompresi multilevel rata-rata memiliki jumlah lebih sedikit daripada metode tanpa kompresi multilevel. Hal ini terjadi dikarenakan ada kesalahan pengambilan pada link secara keseluruhan pada index kata inputan ditemukan yang disebabkan karena algortima pencarian KMP menemukan pembatas link pada nilai index ganjil. Karena hasil link tidak lengkap , walaupun kata inputan ditemukan maka hasil pencarian dianggap tidak sesuai. Hal ini menyebabkan jumlah link yang ditemukan pada metode dengan kompresi multilevel menjadi lebih sedikit, padahal jumlah penemuan kata pada kedua kata memiliki jumlah sama.

Pada percobaan dimana pada metode tanpa kompresi multilevel ditemukan dan metode dengan multilevel tidak ditemukan setelah dilakukan pengamatan pada hasil link pencarian ternyata link pencarian kata yang ditemukan tidak lengkap walaupun sebetulnya kata inputan juga ditemukan pada metode dengan kompresi multilevel namun karena link tidak lengkap maka hasil temuan dianggap halaman yang tidak sesuai.

2. Tiga Kata Dasar dan Berimbuhan

Percobaan selanjutnya adalah dengan melakukan perbedaan jenis inputan kata. Inputan kata yang akan merupakan tiga kata yang terdiri atas kata dasar dan kata berimbuhan. Dilakukan 10 percobaan dengan kata inputan yang berbeda, hasil dari kesepuluh percobaan kata inputan akan di tampilkan pada gambar 4.47 di bawah ini:



Gambar 4. 47 Grafik Jumlah Link yang diperoleh pada Metode Focused Web Crawler dengan dan tanpa Mulitilevel Kompresi Inputan Tiga Kata (Dasar dan Berimbuhan).

Detail hasil link yang diperoleh dari 10 percobaan yang dilakukan, akan di tunjukkan pada tabel 4.22 berikut:

Tabel 4. 22 Tabel Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Tiga Kata (Dasar dan Berimbuhan).

No. Percobaan	Kata	Hasil perolehan link		Selisih
		FWC	FWCM	
1	dipatuk king cobra	58	51	7

2	meminimalisir nyeri rahang	2	2	0
3	merusak terumbu karang	1	1	0
4	jus buah kemasan	1	1	0
5	janji perbaiki aplikasi	3	3	0
6	perizinan online terpadu	14	14	0
7	pecahkan rekor berenang	4	4	0
8	tak berpotensi tsunami	1	1	0
9	berhasil dievakuasi nasional	2	2	0
10	tingkatkan penerimaan negara	2	2	0

Dari jumlah link yang ditemukan 9 dari 10 percobaan menunjukkan hasil link yang sama. Percobaan yang menunjukkan metode tanpa kompresi multilevel memiliki lebih banyak hasil dikarenakan ada penemuan kata inputan yang memiliki tanda baca yang melekat dan ini menyebabkan pencarian dengan menggunakan metode dengan kompresi multilevel menganggap kata tersebut bukan merupakan kata inputan.

|video.tribunnews.com/view/55962/fakta-fakta-remaja-tewas-dipatuk-king-cobra-piaraannya-asal-ular-hingga-pesan-terakhir-kepada-bapak:fakta-fakta remaja tewas dipatuk king cobra…

Link diatas ditemukan pada metode tanpa kompresi multilevel namun tidak pada metode dengan kompresi multilevel, karena jumlah link yang memiliki kata "dipatuk king cobra…" ada 7 link maka selisih hasil link yang didapatkan adalah sebanyak 7 link tersebut.

Selanjutnya untuk melihat apakah hasil link mengandung kata inputan atau tidak serta apakah link merupakan link yang betul(link dan name tag) akan dilakukan pengamatan pada hasil link data. Data pengamatan dirangkum pada tabel 4.23 dibawah :

Tabel 4. 23 Tabel Perbandingan Hasil Link Temuan pada Metode Focused Web Crawler (FWC) dan Metode Focused Web Crawler dengan Mulitilevel Kompresi(FWCM) Inputan Tiga Kata Dasar.

Kata	**FWC**					**FWCM**				
	Hasil link	**LB (Sama)**	**LB (ada kata inputan)**	**LS (sama)**	**LS (ada kata inputan)**	**Hasil link**	**LB (Sama)**	**LB (ada kata inputan)**	**LS (sama)**	**LS (ada kata inputan)**
dipatuk king cobra	58	39	19	19	39	51	0	0	51	51
meminimalisir nyeri rahang	2	2	0	0	2	2	0	0	2	2
merusak terumbu	1	1	0	0	1	1	0	0	1	1

karang										
jus buah kemasan	1	0	1	1	0	1	0	0	1	1
janji perbaiki aplikasi	3	3	0	0	3	3	3	0	0	3
perizinan online terpadu	14	14	0	0	14	14	14	0	0	14
pecahkan rekor berenang	4	4	0	0	4	4	4	0	0	4
tak berpotensi tsunami	1	1	0	0	1	1	1	0	0	1
berhasil dievakuasi nasional	2	2	0	0	2	2	2	0	0	2
tingkatkan penerimaan negara	2	2	0	0	2	2	2	0	0	2

Untuk hasil link dengan metode tanpa kompresi multilevel masih bisa menemukan kata inputan yang mengandung kata/terdapat kata lain. Dari pengamatan 19 belas link yang tidak sama persis dengan inputan untuk kata inputan berupa 3 kata inputan kata lain yang ikut ditemukan adalah lebih kepada karakter bukan kata yang memiliki arti.

|video.tribunnews.com/view/55962/fakta-fakta-remaja-tewas-dipatuk-king-cobra-piaraannya-asal-ular-hingga-pesan-terakhir-kepada-bapak:fakta-fakta remaja tewas dipatuk king cobra...
|www.liputan6.com/regional/read/3583384/dipatuk-king-cobra-peliharaan-bagaimana-kondisi-rizki-sang-pawang-ular:dipatuk king cobra peliharaan, bagaimana kondisi rizki sang pawang ular?
|news.detik.com/berita/d-4106260/keluarga-yakin-rizky-mati-suri-dipatuk-king-cobra-ini-alasannya:keluarga yakin rizky mati suri dipatuk king cobra, ini alasannya

Gambar 4. 48 Hasil 19 Link Metode Tanpa Kompresi multilevel yang Menemukan Kata Inputan yang Memiliki Kata Lain Inputan Tiga Kata (Dasar dan Berimbuhan).

Pada gambar 4.48 menunjukkan 3 dari 19 hasil link pencarian dan ditemukan bahwa karakter yang ditemukan dan menempel pada kata inputan adalah karakter " , ", " : ", dan "…" selain karakter tidak ditemukan kata lain yang memiliki arti. Untuk percobaan lain yang juga menghasilkan kata inputan mengandung kata lain untuk percobaan ke 4 yakni "jus buah kemasan" link yang ditemukan adalah sebanyak 1 yaitu

|www.republika.co.id/berita/gaya-hidup/info-sehat/16/03/14/o40c7q384-lupakan-jus-buah-kemasan-mengapa:lupakan jus buah kemasan, mengapa?

Dari link yang ditemukan kata inputan diikuti karakter "," sehingga dianggap memiliki kata lain pada data tabel pengamatan 4.23. Untuk 4 percobaan yang menunjukkan metode dengan kompresi multilevel tidak menunjukkan hasil, baik memiliki nilai sama persis dengan inputan atau hasil yang mengandung kata inputan. Setelah dilakukan pengamatan, hasil link menemukan kata inputan namun tidak menghasilkan link dan nametag yang utuh. Sama seperti percobaan sebelumnya jika link dan name tag tidak lengkap maka dianggap halaman link yang dihasilkan tidak sesuai. Penyebab link dan name tag tidak dapat ditampilkan secara utuh adalah karena penemuan index karakter pembatas link "l" pada kata pencarian yang dicari memiliki nilai *index* ganjil. Untuk percobaan lain memiliki nilai yang benar dan tidak ada karakter yang menempel pada kata inputan pada history link sehingga kata inputan yang mengandung kata lain tidak ditemukan.

3. Presisi dan Recall 3 Kata Inputan

Selanjutnya adalah perhitungan presisi dan recall pada inputan 3 kata. Untuk nilai presisi dan recall akan dibedakan menjadi dua yakni nilai presisi dan recall untuk 3 kata inputan yang berupa kata dasar, yang kedua adalah nilai presisi dan recall untuk 3 kata inputan yang merupakan gabungan antara kata dasar dan kata berimbuhan. Untuk perhitungan pertama adalah nilai presisi pada 3 kata dasar. Untuk nilai masing-masing percobaan akan ditampilkan pada tabel 4.24 dibawah :

Tabel 4. 24 Tabel Presisi dan Recall 3 Inputan Kata Berupa Kata Dasar

No Perc	Kata	Presisi (sesuai inputan)		Presisi (ada pada kata lain)		Recall (sesuai inputan)		Recall (ada pada kata lain)	
		FWC	FWCM	FWC	FWCM	FWC	FWCM	FWC	FWCM
1	terbang tanpa sayap	1.000	1.000	0.000	0.000	0.03207	0.03207	0.00000	0.00000
2	lampu mobil kusam	0.750	0.750	0.250	0.250	0.00875	0.00875	0.00292	0.00292
3	polisi di lampung	1.000	1.000	0.000	0.000	0.00583	0.00583	0.00000	0.00000
4	buka konser celine	1.000	1.000	0.000	0.000	0.01749	0.01749	0.00000	0.00000
5	kpk tahan anggota	0.000	0.000	1.000	1.000	0.00000	0.00000	0.00292	0.00292
6	asian games 2018	0.665	0.572	0.281	0.180	0.53936	0.46356	0.22741	0.14577
7	untuk ibu hamil	1.000	0.000	0.000	0.000	0.00875	0.00000	0.00000	0.00000
8	mantan agen rusia	1.000	1.000	0.000	0.000	0.00292	0.00292	0.00000	0.00000
9	minta proses hukum	1.000	1.000	0.000	0.000	0.01166	0.01166	0.00000	0.00000

10	pensiun dari timnas	1.000	1.000	0.000	0.000	0.03790	0.03790	0.00000	0.00000
11	sahabat putri diana	1.000	0.000	0.000	0.000	0.01749	0.00000	0.00000	0.00000
12	dapat bunga mawar	1.000	1.000	0.000	0.000	0.01458	0.01458	0.00000	0.00000
13	beli lahan tanah	1.000	1.000	0.000	0.000	0.00292	0.00292	0.00000	0.00000
14	harga jual dollar	1.000	0.000	0.000	0.000	0.00875	0.00000	0.00000	0.00000
15	korban bom pasuruan	1.000	1.000	0.000	0.000	0.01458	0.01458	0.00000	0.00000

Pada hasil perhitungan nilai recall untuk 3 kata inputan berupa kata dasar, nilai presisi mencapai angka 1 rata-rata terjadi pada hasil link kata yang memiliki hasil pencarian sama persis dengan kata inputan. Untuk nilai presisi pada kata inputan yang mengandung kata lain memiliki nilai yang lebih sedikit dan hanya sedikit percobaan yang memiliki hasil link dengan kata inputan yang mengandung kata lain. Ini terjadi karena jumlah link yang ditemukan selalu lebih sedikit daripada total link yang dihasilkan dalam sekali pencarian kata inputan sehingga pembagi nilai link yang sesuai lebih banyak dan menghasilkan nilai yang lebih sedikit. Untuk nilai recall terbanyak dihasilkan pada link yang memiliki nilai relevan banyak pula. Untuk nilai presisi dan recall kedua metode memiliki nilai yang rata-rata adalah sama. Pada beberapa kasus yang mengakibatkan ketidak sesuaian link menyebabkan metode dengan kompresi multilevel memiliki nilai presisi dan recall yang lebih sedikit daripada metode tanpa kompresi multilevel.

Perhitungan kedua adalah nilai presisi dan recall untuk tiga kata inputan dengan kata dasar dan kata berimbuhan. Untuk nilai presisi dan recall akan di tampilkan pada tabel 4.25 dibawah:

Tabel 4. 25 Tabel Presisi dan Recall 3 Inputan Kata Berupa Kata Dasar dan Kata Berimbuhan.

No Perc	**Kata**	**Presisi (sesuai inputan)**		**Presisi (ada pada kata lain)**		**Recall (sesuai inputan)**		**Recall (ada pada kata lain)**	
		FWC	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**	**FWC**	**FWCM**
1	dipatuk king cobra	0.672	0.000	0.328	0.000	0.44318	0.00000	0.21591	0.00000
2	meminimalisir nyeri rahang	1.000	0.000	0.000	0.000	0.02273	0.00000	0.00000	0.00000
3	merusak terumbu karang	1.000	0.000	0.000	0.000	0.01136	0.00000	0.00000	0.00000
4	jus buah	0.000	0.000	1.000	0.000	0.00000	0.00000	0.01136	0.00000

	kemasan								
5	janji perbaiki aplikasi	1.000	1.000	0.000	0.000	0.03409	0.03704	0.00000	0.00000
6	perizinan online terpadu	1.000	1.000	0.000	0.000	0.15909	0.17284	0.00000	0.00000
7	pecahkan rekor berenang	1.000	1.000	0.000	0.000	0.04545	0.04938	0.00000	0.00000
8	tak berpotensi tsunami	1.000	1.000	0.000	0.000	0.01136	0.01235	0.00000	0.00000
9	berhasil dievakuasi nasional	1.000	1.000	0.000	0.000	0.02273	0.02469	0.00000	0.00000
10	tingkatkan penerimaan negara	1.000	1.000	0.000	0.000	0.02273	0.02469	0.00000	0.00000

Untuk nilai presisi dan recall pada tiga kata inputan yang berupa kata dasar dan berimbuhan rata-rata nilai presisi dan recall antara kedua metode adalah memiliki nilai yang sama karena hasil link yang ditemukan oleh pencarian memiliki nilai yang sama. Untuk nilai presisi tertinggi adalah 1 dan nilai ini hampir dimiliki pada setiap percobaan. Untuk beberapa percobaan yang memiliki nilai presisi kurang dari 1 atau bahkan 0 dipengaruhi oleh jumlah link tidak sesuai yang ditemukan oleh metode pencarian. Ketidak sesuaian pada pencarian ini memiliki penyebab antara lain link yang dihasilkan masih mengandung kata/karakter lain (presisi sesuai inputan) / link yang dihasilkan sama persis dengan inputan (presisi mengandung kata lain) , selain itu link yang ditemukan bukan link dan nametag lengkap sehingga dianggap bukan link sesuai. Untuk hasil recall karena yang digunakan juga link sesuai maka penyebab nilai presisi juga menjadi penyebab nilai recall tidak bisa begitu tinggi. Secara keseluruhan untuk nilai presisi, pada kedua metode memiliki nilai rata-rata baik untuk link yang memiliki hasil sama persis dengan inputan kata. Untuk hasil kata inputan mengandung kata lain metode tanpa kompresi multilevelsi memiliki nilai presisi yang lebih baik, terbukti ditemukan nilai presisi pada 2 percobaan yang dilakukan.

Sedangkan untuk nilai recall pada hasil link sesuai dengan kata inputan, metode dengan menggunakan kompresi multilevel memiliki nilai rata-rata recall lebih besar daripada metode tanpa kompresi multilevel. Untuk hasil kata inputan masih mengandung kata lain metode tanpa kompresi multilevel memiliki nilai recall, sedangkan metode dengan kompresi multilevel tidak memiliki nilai recall atau 0.

Dari ketiga jenis pengujian untuk nilai presisi dan recall ditemukan ada beberapa (tidak semua) data test yang memiliki nilai presisi dan recall yang berbeda walaupun hasil link output memiliki nilai yang sama. Berikut pada tabel 4.26 adalah beberapa contoh kasus yang terjadi dimana hasil link output data test memiliki nilai sama namun nilai presisi atau recall memiliki nilai yang berbeda.

Tabel 4. 26 Contoh Kasus Output Link Sama dengan nilai Presisi dan Recall Berbeda.

Kata	FCW					FCWM				
	Hasil link	LB (Sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)	Hasil link	LB (Sama)	LB (ada kata inputan)	LS (sama)	LS (ada kata inputan)
manfaat tidur	9	8	1	1	8	9	7	0	2	9
meminimalisir nyeri rahang	2	2	0	0	2	2	0	0	2	2

Kata	Presisi (sesuai inputan)		Presisi (ada pada kata lain)		Recall (sesuai inputan)		Recall (ada pada kata lain)	
	FCW	FCWM	FCW	FCWM	FCW	FCWM	FCW	FCWM
manfaat tidur	0.889	0.778	0.111	0.000	0.03213	0.02778	0.00402	0.00000
meminimalisir nyeri rahang	1.000	0.000	0.000	0.000	0.02273	0.00000	0.00000	0.00000

Dari dua contoh kasus diatas yang mempengaruhi nilai presisi dan recall adalah LB (link benar) dan LS (link salah). Jika hasil LB lebih besar maka nilai presisi dari data tersebut akan semakin besar dan sebaliknya. Sedangkan untuk nilai recall selain LB dan LS jumlah semua link output yang didapat mempengaruhi nilai recall. Semakin banyak hasil output dan merupakan LB maka nilai recall makin besar dan sebaliknya makin banyak output dan LS banyak maka nilai recall semakin kecil.

Dalam pengklasifikasian link benar seperti yang disebutkan sebelumnya link benar adalah link yang bisa di lakukan pencarian melalui browser, karena hasil link output di bedakan menjadi dua jenis yang sama dengan inputan dan yang mengandung inputan maka klasifikasi hasil output akan disesuaikan dengan dua jenis tersebut. Pada dua kolom dibawah merupakan hasil link output data tes “manfaat tidur” yang memiliki perbedaan.

```
-siang-yang-tak-banyak-diketahui:7 manfaat tidur siang yang tak banyak
diketahui
Hasil rentang index: [4885127, 4885152, 4885178]
-siang:manfaat tidur siang
Hasil rentang index: [4885499, 4885504, 4885514]
|gayahidup.republika.co.id/berita/gaya-hidup/info-sehat/18/07/08/pbjfz2313-
empat-manfaat-tidur-dalam-posisi-miring:empat manfaat tidur dalam posisi
miring
Hasil rentang index: [5075802, 5075884, 5075902]
|gayahidup.republika.co.id/berita/gaya-hidup/info-sehat/18/07/08/pbjfz2313-
empat-manfaat-tidur-dalam-posisi-miring:empat manfaat tidur dalam posisi
miring
Hasil rentang index: [5707534, 5707616, 5707634]
```

```
|www.liputan6.com/health/read/3379189/7-manfaat-tidur-siang-yang-tak-banyak-
diketahui:7 manfaat tidur siang yang tak banyak diketahui
Hasil rentang index: [7672934, 7673022, 7673067]
|www.liputan6.com/tag/manfaat-tidur-siang:manfaat tidur siang
Hasil rentang index: [7673503, 7673545, 7673564]
|gayahidup.republika.co.id/berita/gaya-hidup/info-sehat/18/07/08/pbjfz2313-
empat-manfaat-tidur-dalam-posisi-miring:empat manfaat tidur dalam posisi
miring
Hasil rentang index: [7975202, 7975323, 7975356]
|gayahidup.republika.co.id/berita/gaya-hidup/info-sehat/18/07/08/pbjfz2313-
empat-manfaat-tidur-dalam-posisi-miring:empat manfaat tidur dalam posisi
miring
Hasil rentang index: [8979899, 8980020, 8980053]
```

Pada dua kolom dibawah selanjutnya merupakan hasil link output data tes "meminimalisir nyeri rahang" yang memiliki perbedaan.

```
kampus/18/07/04/pbc8vj399-gusta-headset-untuk-meminimalisir-nyeri-
rahang:gusta, headset untuk meminimalisir nyeri rahang
Hasil: [7181559, 7181620, 7181630]
kampus/18/07/04/pbc8vj399-gusta-headset-untuk-meminimalisir-nyeri-
rahang:gusta, headset untuk meminimalisir nyeri rahang
Hasil: [8914455, 8914516, 8914526]
```

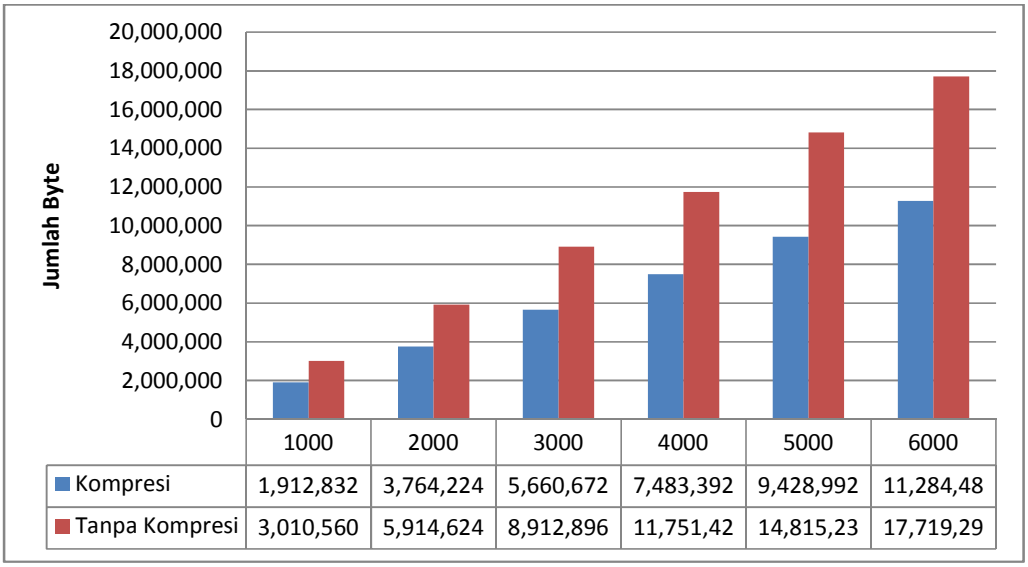
```
|www.republika.co.id/berita/pendidikan/dunia-kampus/18/07/04/pbc8vj399-gusta-
headset-untuk-meminimalisir-nyeri-rahang:gusta, headset untuk meminimalisir
nyeri rahang
Hasil: [11274153, 11274292, 11274318]
|www.republika.co.id/berita/pendidikan/dunia-kampus/18/07/04/pbc8vj399-gusta-
headset-untuk-meminimalisir-nyeri-rahang:gusta, headset untuk meminimalisir
nyeri rahang
Hasil: [13997506, 13997645, 13997671]
```

Dua contoh hasil kesalahan output dari dua data testing menjadikan link output yang memiliki kata inputan menjadi link salah. Kesalahan pada pencarian + kompresi multilevel ditemukan pada rentang index hasil pencarian karakter pembatas link yang dimulai dengan nilai ganjil menyebabkan nilai link yang ditemukan tidak lengkap/tidak benar. Sedangkan pada metode tanpa multilevel rentang index hasil pencarian karakter

pembatas link nilai ganjil tidak ada masalah. Pada metode pencarian + kompresi multilevel index nilai pembatas link ganjil terjadi kesalahan karena dalam pencarian metode yang diusulkan pencocokan KMP dilakukan per 2 byte rentang index, sedangkan pada index metode sebelumnya pencocokan KMP dilakukan per byte. Karena link salah digunakan dalam perhitungan nilai presisi dan recall, maka pencarian rentang index dalam metode KMP pada usulan metode perlu di lakukan perubahan algoritma untuk mengatasi kondisi tersebut agar nilai presisi dan recall tidak terpengaruh oleh hasil pencarian KMP yang terkadang memberikan link salah.

4.3.4.2 Uji Coba dan Analisa Rasio Kompresi Data link URL

Dalam tahap ini yang dilakukan adalah melakukan percobaan untuk melakukan kompresi terhadap data link url yang telah dikumpulkan untuk kemudian dilakukan penyimpanan *history* pada link url. Tujuan dari tahap ini adalah untuk mengetahui seberapa besar rasio dari metode kompresi yang diajukan. Dalam uji coba ini beberapa file kumpulan data link dan *tag name* dengan jumlah link berbeda dari 1000, 2000, 3000, 4000, 5000, dan 6000 akan dibandingkan seberapa banyak ukuran yang dapat di kurangi. Gambar 4.49 menunjukkan ukuran penyimpanan byte pada proses kompresi



	1000	2000	3000	4000	5000	6000
Kompresi	1,912,832	3,764,224	5,660,672	7,483,392	9,428,992	11,284,48
Tanpa Kompresi	3,010,560	5,914,624	8,912,896	11,751,42	14,815,23	17,719,29

Gambar 4. 49 Grafik Perbandingan Ukuran Penyimpanan Byte Sebelum dan Setelah Kompresi

Dari hasil pencatatan pengompresan yang dilakukan yang dicatat adalah jumlah kebutuhan memori penyimpanan dalam disk. Angka yang ditunjukkan berupa jumlah

byte dari data untuk sumbu x,dan pada sumbu y merupakan banyaknya jumlah halaman link web yang telah di kumpulkan dalam proses *crawling*. Perhitungan rasio dari data sebelum dan sesudah akan disajikan pada tabel 4.27.

Tabel 4. 27 Tabel Rasio Kompresi Data Link Url Hasil *Crawling*

Banyak Halaman	Ukuran sebelum (byte)	Ukuran setelah(byte)	Rasio
1000	3,010,560	1,912,832	36.463
2000	5,914,624	3,764,224	36.357
3000	8,912,896	5,660,672	36.489
4000	11,751,424	7,483,392	36.319
5000	14,839,808	9,445,376	36.351
6000	17,719,296	11,284,480	36.315

Pada semua hasil percobaan menunjukkan metode mampu menghemat ukuran file rata-rata dengan rasio pada angka 36%. Angka rata-rata 36% rasio kompresi didapat karena kompresi menuliskan tiap index kamus kata sebagai 2 byte pada sistem, untuk membuktikan dilakukan proses kompresi dengan metode kamus kata dengan penyimpanan tiap index kamus kata adalah 3 byte didapatkan hasil pada tabel 4.28:

Tabel 4. 28 Tabel Rasio Kompresi Data Link Url Hasil *Crawling* (3 byte per index kata)

Banyak Halaman	Ukuran sebelum (byte)	Ukuran setelah(byte)	Rasio
1000	3,010,560	2,867,200	4.762
2000	5,914,624	5,644,288	4.571
3000	8,912,896	8,486,912	4.779
4000	11,751,424	11,223,040	4.496
5000	14,839,808	14,165,724	4.542
6000	17,719,296	16,924,672	4.485

Dari hasil yang diperoleh menunjukkan rata-rata rasio kompresi adalah sebesar berkisar 4 % , besarnya penyimpanan byte pada setiap index kamus kata mempengaruhi rasio kompresi pada metode kompresi multilevel yang diusulkan. Semakin besar ukuran byte yang dialokasikan untuk sebuah index maka ukuran rasio akan semakin kecil.

Selain mencatat keberhasilan kompresi hasil pengompresan, akan dilakukan percobaan untuk melihat kebenaran hasil kompresi yaitu dengan melakukan proses decoding pada file terencode. Parameter yang digunakan untuk penilaian adalah ukuran file sebelum dan sesudah beserta kesesuaian konten file hasil decode.Untuk hasil di sajikan didalam tabel 4.29 dibawah ini:

Tabel 4. 29 Tabel Perbandingan Hasil Decoding dengan Metode Kompresi Multilevel

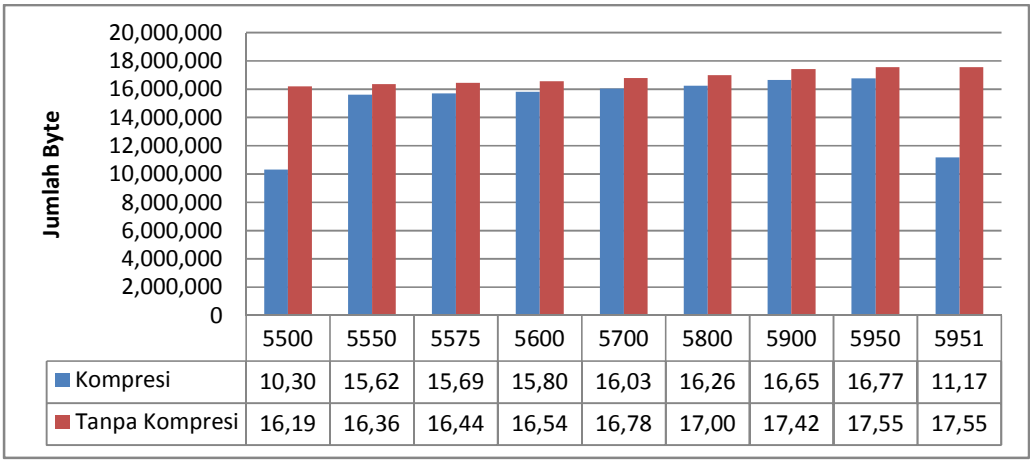
Banyak Halaman	Ukuran sebelum(byte)	Ukuran setelah(byte)	Selisih(byte)	Hasil Link
1000	3,010,560	3,010,560	0	sesuai
2000	5,914,624	5,910,528	4,096	sesuai
3000	8,912,896	8,904,704	8,192	sesuai
4000	11,751,424	11,743,232	8,192	sesuai
5000	14,839,808	14,827,520	12,288	sesuai
6000	17,719,296	46,620,672	-28,901,376	Tidak sesuai

Pada tabel 4.29 perbandingan jumlah ukuran file sebelum dan sesudah rata-rata ukuran sebelum memiliki size yang lebih besar namun selisih ukuran tidak terlalu jauh berbeda. Perbedaan terjadi karena untuk proses decoding pada data terencode karakter pergantian baris didalam file text tidak dilakukan sehingga mengurangi sedikit size dari file asli. Namun untuk kesesuaian isi file semua sesuai, hal ini terbukti dengan sudah dilakukannya pengamatan pada hasil data decoding secara manual. Pada hasil percobaan crawl 6000 link, hasil decoding melebihi data tanpa kompresi dimana besar nya data menjadi 1,6 kali lipat dari ukuran sebelumnya. Selain ukuran data yang membesar hasil decode dari data yang terencode juga mengalami kesalahan, sedangkan lima percobaan lainnya tidak.

Setelah di lakukan pengamatan, ternyata penyebab hal ini terjadi adalah pada kompresi data hasil crawl 6000 link jenis kata yang dimasukkan dalam kamus kata telah melebihi batas maksimal jenis kata yaitu sebesar 65.536. Sedangkan pada kompresi 6000 link tabel kata yang ada sebanyak 66.239. Oleh karena itu terjadi masalah karena pembacaan untuk konversi integer menjadi byte maupun sebaliknya pada proses encode maupun decode hanya disediakan sampai 2 byte dengan maksimum jumlah nilai *index* maksimum adalah 2^16 atau sebanyak 65.536. Untuk kompresi 5000 link sebelumnya belum ada masalah karena pada kompresi 5000 link banyak jenis kata unik yang dibuat kamus kata adalah sebanyak 59.032. Karena masih cukup *index* tersisa untuk kamus kata yaitu batas maksimum(65.536) – *index* maksimum(59.032) sekitar 6.504 *index* tersisa.

Untuk mengetahui maksimum link crawl yang dapat di kompresi oleh metode maka dilakukan kompresi pada data link hasil crawl pada 5000-6000 link. Percobaan

dilakukan dengan mencoba melakukan kompresi dengan menambahkan jumlah link antara crawl 25-100 link pada pergantian percobaan. Percobaan akan di tampilkan pada gambar 4.50 dibawah ini:



Gambar 4. 50 Grafik Perbandingan Ukuran Penyimpanan Byte Sebelum dan Setelah Kompresi pada 5000-6000 Crawl Link

Untuk rasio kompresi menunjukkan tidak adanya masalah, namun perlu dilakukan proses decoding untuk melihat kebenaran hasil kompresi. Untuk pengamatan dilakukan dengan mencatat perbandingan ukuran file sebelum dan setelah terdecode serta melihat kebenaran nilai decoding. Tabel 4.30 menunjukkan hasil ukuran file dan nilai kebenaran link hasil.

Tabel 4. 30 Tabel Ukuran Hasil Sebelum dan Sesudah Decoding File pada 5500-6000 Crawl Link

Banyak Halaman	**Ukuran sebelum(byte)**	**Ukuran setelah(byte)**	**Selisih(byte)**	**Hasil Link**
5500	16,199,680	16,187,392	12,288	sesuai
5575	16,445,440	16,433,152	12,288	sesuai
5600	16,543,744	16,535,552	8,192	sesuai
5700	16,781,312	16,769,024	12,288	sesuai
5800	17,006,592	16,998,400	8,192	sesuai
5900	17,420,288	17,408,000	12,288	sesuai
5950	17,551,360	17,539,072	12,288	sesuai

5951	17,555,456	50,868,224	-33,312,768	tidak sesuai

Pada hasil pengamatan dari percobaan beberapa link diatas menunjukkan kesalahan link terjadi pada saat link berada data ke 5951 crawl link. Jadi maksimum halaman yang dapat terkompresi dalam kondisi data link pada saat kompresi hasil 5951 crawl link data sekitar 123.992 link url dan name tag.

Untuk mencoba mengatasi kelemahan dari metode multilevel kompresi dengan menggunakan kamus kata dengan *index* 2 byte yang hanya mampu menampung sebanyak 65.536 jenis *index* kata,maka dilakukan percobaan dengan membuat kompresi dengan menggunakan kamus kata dengan penyimpanan per *index* sebesar 3 byte. Untuk 3 byte diharapkan mampu menampung lebih banyak jenis kata untuk dijadikan kamus kata , maksimal jenis *index* jika menggunakan 3 byte penyimpanan adalah sebesar 2^24 yakni sebanyak 16.777.216 . Untuk percobaan kita menggunakan data hasil crawl pada 6000 link dimana pada kompresi 2 byte data masih terjadi kesalahan pada hasil decoding. Hasil kompresi menunjukkan bahwa setelah jumlah maksimum *index* kata lebih banyak pada kompresi 3 byte, hasil decode yang dihasilkan berhasil atau tidak terjadi kesalahan. Pada gambar dibawah akan ditunjukkan hasil decode. Untuk perbandingan ukuran file akan ditampilkan pada tabel 4.31 dibawah:

Tabel 4. 31 Tabel Perbandingan Ukuran Hasil Sebelum dan Sesudah Decoding File pada 6000 Crawl Link

Hal.	2 byte kompresi		Hasil Link	3 byte kompresi		Hasil Link
	Uk.sebelum (byte)	Uk setelah (byte)		Uk sebelum (byte)	Uk setelah (byte)	
6000	17,719,296	46,620,672	tidak sesuai	17,719,296	17,702,912	sesuai

|connect.detik.com/accounts/register/option/?redirecturl=https%3a%2f%2fwww.cnnindonesia.com%2fauthorize%3fu%3dhttps%253a%252f%252fwww.cnnindonesia.com%252fnasional%252f20180706190129-12-312128%252fanak-buah-rita-widyasari-divonis-8-tahun-penjara&clientid=10027&ui=popup:daftar|connect.detik.com/oauth/authorize?redirecturl=https%3a%2f%2fwww.cnnindonesia.com%2fauthorize%3fu%3dhttps%253a%252f%252fwww.cnnindonesia.com%252fnasional%252f20180706190129-12-312128%252fanak-buah-rita-widyasari-divonis-8-tahun-penjara&clientid=10027&ui=popup:masuk|

comnasional20171026150144410362613282938559961328292015062004l243ou0g4w4l4309574khofifahou0g4w414comolahraga20180708071013ou0g4w4l44103529ou0g4w4l420180630003531201806271831311069380181306pbdxjc399201807051959 5lpbdxjc399p9lnmz415pbdxjc399409368813282938903161328292015062004l243pbdxjc399410361 4pbdxjc399201804061922 12pbdxjc399410359

|connect.detik.com/accounts/register/option/?redirecturl=https%3a%2f%2fwww.cnnindonesia.com%2fauthorize%3fu%3dhttps%253a%252f%252fwww.cnnindonesia.com%252fnasional%252f20180706190129-12-312128%252fanak-buah-rita-widyasari-divonis-8-tahun-penjara&clientid=10027&ui=popup:daftar|connect.detik.com/oauth/authorize?redirecturl=https%3a%2f%2fwww.cnnindonesia.com%2fauthorize%3fu%3dhttps%253a%252f%252fwww.cnnindonesia.com%252fnasional%252f20180706190129-12-312128%252fanak-buah-rita-widyasari-divonis-8-tahun-penjara&clientid=10027&ui=popup:masuk|

Gambar 4. 51 Hasil Decoding Pada Kompresi 2 byte dan Kompresi 3 byte

Dari gambar 4.51 menunjukkan hasil decoding pada kompresi, potongan link pada kolom pertama adalah inputan asli data file, untuk kolom kedua adalah kondisi hasil decoding kompresi pada 2 byte yang melebihi *index* maksimum,untuk kolom ketiga adalah decoding kompresi pada 3 byte penyimpanan. Dari hasil proses kompresi 3 byte penyimpanan bisa dilihat bahwa pada maksimum data *index* percobaan 6000 link dengan nilai *index* 66.239(pada 2 byte) dan 66.224 (pada 3 byte) masih menghasilkan hasil decoding yang benar. Namun perbedaan yang terjadi adalah size kedua hasil data kompresi memiliki selisih,ditunjukkan pada gambar 4.52. Rata2 pengematan pada kompresi 2 byte adalah memiliki rasio sebesar 37% sedangkan untuk 3 byte rasio menjadi hanya 5% saja. Ini terjadi karena dalam kompresi sebelum dan sesudah byte yang disimpan didalam proses encode naik 1 byte dari awal mula 2 byte data menjadi 3 byte data, namun kelebihannya jenis *index* data maksimum yang dihasilkan lebih banyak yakni sebesar 16.777.216 sehingga kompresi masih bisa dilakukan untuk jumlah data crawl link dan name tag yang lebih banyak daripada 5950 crawl link pada kompresi 2 byte penyimpanan.

Encoding Data
compress ratio %37
Searching Data
Took 942 miliseconds.

```
Encoding Data
compress ratio %5
Searching Data
Took 168215 miliseconds.
```

Gambar 4. 52 Perbandingan penghematan size kompresi 2 byte dan 3 byte

4.3.4.3 Uji Coba Waktu Pencarian

Percobaan selanjutnya adalah melakukan pengamatan terhadap waktu yang diperlukan untuk melakukan sekali proses pencarian yang dibutuhkan oleh metode focused web crawler dengan kompresi multilevel dan metode focused web crawler tanpa kompresi multilevel. Dalam uji coba waktu pencarian data inputan yang digunakan adalah kata inputan yang terdiri dari tiga kata masukan. Tiga kata inputan di coba karena dari kata inputan ini memiliki hasil pencarian link dengan jumlah yang sama besar, sedangkan untuk kata inputan lain masih banyak perbedaan jumlah kata pencarian terutama jika kata inputan hanya 1 dan merupakan kata dasar. Dari pencarian kata akan dilakukan, pencatatan waktu dan akan ditampilkan pada tabel 4.32 dibawah.

Tabel 4. 32 Tabel Perbandingan Waktu Pencarian Metode Focused Web Crawler (sampai muncul hasil link dan name tag)

Kata	Focused web crawler + Kompresi multilevel (milisecond)	Focused web crawler (milisecond)	Selisih
dipatuk king cobra	3963	541	3422
meminimalisir nyeri rahang	564	548	16
merusak terumbu karang	614	615	-1
jus buah kemasan	548	508	40
janji perbaiki aplikasi	2048	830	1218
tak berpotensi tsunami	641	600	41
berhasil dievakuasi nasional	666	577	89
tingkatkan penerimaan negara	716	550	166

Dari pencatatan waktu pencarian untuk kedua metode terlihat bahwa metode focused web crawler memiliki waktu yang lebih lama untuk sama-sama menghasilkan link dan name tag. Ini terjadi karena perbedaan step dalam menghasilkan link dan name tag, perbedaannya adalah jika metode focused web crawler dengan kompresi multilevel setelah dilakukan pencarian pada data terencode link dan name tag harus dilakukan proses decoding untuk bisa ditampilkan seperti sebelum terkompresi, sedangkan pada

metode tanpa kompresi multilevel setelah pencarian kata inputan dilakukan hasil link dan name tag langsung merupakan hasil akhir yang bisa dibaca oleh user.

Pada metode focused web crawler dengan kompresi multilevel berguna untuk mengurangi jumlah ukuran byte data history link, dari pemikiran ini maka akan dilakukan pengamatan pada waktu pencarian algoritma KMP dengan tahap dimana kedua metode sama-sama harus melalui tahap tersebut yakni hingga menemukan *index* kata didalam file. Percobaan menggunakan data inputan sama seperti pencarian untuk menghasilkan link dan name tag sebelumnya. Berikut hasil pengamatan waktu pencarian tabel 4.33 dibawah :

Tabel 4. 33 Tabel Perbandingan Waktu Pencarian KMP (tahap 1) Metode Focused Web Crawler

Kata	Focused web crawler + Kompresi multilevel (millisecond)	Focused web crawler (millisecond)	Selisih
dipatuk king cobra	519	380	139
meminimalisir nyeri rahang	415	388	27
merusak terumbu karang	413	470	-57
jus buah kemasan	475	478	-3
janji perbaiki aplikasi	462	500	-38
tak berpotensi tsunami	488	544	-56
berhasil dievakuasi nasional	486	542	-56
tingkatkan penerimaan negara	492	502	-10

Dari hasil pengamatan pada hasil waktu untuk proses pencarian dengan KMP pada metode focused web crawler dengan kompresi multilevel rata-rata memiliki nilai waktu yang lebih sedikit daripada focused web crawler tanpa multilevel. Dari percobaan ini menjelaskan bahwa yang membuat pencarian focused web crawler dengan kompresi multilevel lebih lama adalah proses untuk melakukan decoding pada data terencode yang telah disimpan.

[Halaman ini sengaja dikosongkan]

BAB 5
KESIMPULAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Berdasarkan uji coba dan analisa hasil, maka dapat ditarik beberapa kesimpulan antara lain:

1. Proses kompresi dilakukan untuk mengurangi ukuran file penyimpanan *history* pencarian website cara yang dilakukan adalah dengan melakukan teknik kompresi multilevel yaitu dengan melakukan pengurangan kata yang tidak penting dan pembuatan index kamus kata yang nantinya disimpan kedalam sistem sebagai nilai byte.
2. Metode kompresi multilevel yang diusulkan memiliki rasio penyimpanan rata-rata sekitar 36.4% pada disk. Maksimum jenis kata yang bisa disimpan untuk kamus kata adalah sebanyak 65.536 jenis kata (2^8). Semakin banyak jenis kata yang ada pada link web maka semakin terbatas pula kapasitas maksimum kamus kata yang dimiliki,dan jika maksimum kata sudah terlewati hasil encoding akan mengalami kesalahan. Besarnya byte pada index kamus kata mempengaruhi nilai rasio kompresi file.
3. Untuk membuktikan bahwa metode yang diusulkan tidak mengurangi nilai kebenaran terhadap hasil link yang dicari dilakukan perhitungan nilai presisi dan recall pada output link hasil pencarian. Dari hasil output link pencarian diketahui bahwa pada metode yang diusulkan, untuk percobaan 1 inputan kata dengan pencarian kata yang sama persis dengan inputan menghasilkan nilai presisi sebesar 1 dan nilai recall yang didapatkan 0.73. Sedangkan metode sebelumnya memiliki nilai presisi sebesar 0.002 dan nilai recall sebesar 0.0001 untuk kata inputan yang sama.

Beberapa saran yang bisa dilakukan untuk pengembangan selanjutnya dari metode pencarian focused web crawler dengan kompresi multilevel:

1. Penyesuaian pencarian pada metode KMP untuk penemuan *index* karakter pembatas agar tidak terjadi kesalahan pengambilan *index* pada link dan name tag.

2. Proses tokenizing pada link dan name tag dengan memperhatikan karakter yang sering muncul sehingga bisa mengurangi keragaman jenis kata sehingga proses kompresi dapat berjalan pada banyak link inputan.

DAFTAR PUSTAKA

Avraam, I. (2011) 'A Comparison over Focused Web Crawling Strategies', in *Panhellenic Conference on Informatics*. IEEE, pp. 245–249. doi: 10.1109/PCI.2011.53.

Bell, T., Witten, I. H. and Cleary, J. G. (1989) 'Modeling for text compression', *ACM Computing Surveys*, 21(4), pp. 557–591. doi: 10.1145/76894.76896.

Dewanjee, J. (2016) 'Heuristic Approach for Designing a Focused Web Crawler using Cuckoo Search', *International Journal of Computer Sciences and Engineering*, 04(09), pp. 59–63.

Dwivedi, S. K. and Arya, C. (2017) 'News web page classification using url content and structure attributes', in *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016*, pp. 317–322. doi: 10.1109/NGCT.2016.7877434.

Ganguly, B. and Raich, D. (2014) 'Performance optimization of focused web crawling using content block segmentation', in *Proceedings - International Conference on Electronic Systems, Signal Processing, and Computing Technologies, ICESC 2014*, pp. 365–370. doi: 10.1109/ICESC.2014.69.

Kalajdzic, K., Ali, S. H. and Patel, A. (2015) 'Rapid lossless compression of short text messages', *Computer Standards & Interfaces*. Elsevier B.V., 37(JUNE), pp. 53–59. doi: 10.1016/j.csi.2014.05.005.

Kan, M. (2005) 'Fast webpage classification using URL features', in *Proceedings of the 14th ACM international conference on Information and knowledge management*. Bremen, Germany, pp. 325–326. doi: 10.1145/1099554.1099649.

Kanda, S., Morita, K. and Fuketa, M. (2017) 'Practical String Dictionary Compression Using String Dictionary Encoding', 0, pp. 4–11. doi: 10.1109/Innovate-Data.2017.9.

Kodabagi, M. M. (2015) 'Multilevel Security and Compression of Text Data using Bit Stuffing and Huffman Coding', in *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. Davangere, India: IEEE, pp. 800–804. doi: 10.1109/ICATCCT.2015.7456992.

Lelewer, D. a and Hirschberg, D. S. (2004) *Data Compression*, *ACM Computing Surveys (CSUR)*. doi: 10.1007/b97635.

Local, P. C. *et al.* (2018) *Metaheuristic*.

Mahmood, A. and Hasan, K. M. A. (2017) 'An Efficient 6 Bit Encoding Scheme for Printable Characters by Table Look Up', in *International Conference on Electrical, Computer and Communication Engineering (ECCE), February 16-18, 2017, Cox's Bazar, Bangladesh An*. Cox's Bazar, Bangladesh: IEEE, pp. 468–472. doi: 10.1109/ECACE.2017.7912950.

Pant, G. and Srinivasan, P. (2006) 'Link contexts in classifier-guided topical crawlers', *IEEE Transactions on Knowledge and Data Engineering*, 18(1), pp. 107–122. doi: 10.1109/TKDE.2006.12.

Pratt, V. (2012) *Knuth – Morris – Pratt algorithm*.

Rawat, S. (2012) 'Efficient Focused Crawling based on Best First Search', pp. 908–911.

S, G. E. S. (2015) *Kecerdasan Buatan (Metode Heuristic)*.

Salomon, D. (2004) *Data compression*, *Departmen of computer Science*

california State University, Northridge. doi: 10.1002/9781118256053.ch13.

Wei, Z. *et al.* (2016) 'Mining and Harvesting High Quality Topical Resources from the Web □', 25(1). doi: 10.1049/cje.2016.01.008.

Wikipedia (2018) *Heuristik*.

Wol, J. G. (1999) 'Probabilistic Reasoning as Information Compression by Multiple Alignment , Uni cation and Search : An Introduction and Overview', 5(7), pp. 418–462.

Yang, X. *et al.* (2018) *Cuckoo search*.

Yang, X., Deb, S. and Behaviour, A. C. B. (2009) 'Cuckoo Search via L ´ evy Flights', pp. 210–214.

Yohanes, B. W., Handoko and Wardana, H. K. (2011) 'Focused Crawler Optimization Using Genetic Algorithm', *Telkomnika*, 9(3), pp. 403–410. doi: 10.12928/telkomnika.v9i3.730.

BIOGRAFI PENULIS

Dian Septiani Santoso lahir di Blitar, Jawa Timur pada tanggal 28 September 1990.Penulis menempuh pendidikan dasar pada tahun 1997 di SD Karangsari IV, pendidikan menengah pertama pada tahun 2003 di SMPN 1 Blitar dan menengah atas pada tahun 2006 di SMAN 1 Blitar di kota Blitar, Jawa Timur. Selanjutnya meneruskan pendidikan jenjang sarjana pada tahun 2009 di Politeknik Elektronika Surabaya di Kota Surabaya, Jawa Timur dengan jurusan Teknik Informatika. Setelah menyelesaikan pendidikan sarjana penulis sempat bekerja selama 2 tahun sebagai karyawan dan melajutkan Pendidikan Magister (S2) di Institut Teknologi Sepuluh Nopember jurusan Teknik Informatika pada tahun 2016. Untuk menghubungi penulis silahkan menghubungi alamat email berikut dieant0947@gmail.com.