

SKRIPSI

INCREMENTAL WEB CRAWLING PADA LINGKUNGAN
HADOOP



Melinda Nur Abianti

NPM: 2014730012

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2018

UNDERGRADUATE THESIS

**INCREMENTAL WEB CRAWLING ON HADOOP
ENVIRONMENT**



Melinda Nur Abianti

NPM: 2014730012

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2018**

LEMBAR PENGESAHAN



**INCREMENTAL WEB CRAWLING PADA LINGKUNGAN
HADOOP**

Melinda Nur Abianti

NPM: 2014730012

Bandung, 31 Mei 2018

Menyetujui,

Pembimbing

Gede Karya, M.T., CISA, IPM

Ketua Tim Penguji

Chandra Wijaya, M.T.

Anggota Tim Penguji

Rosa De Lima, M.Kom.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

INCREMENTAL WEB CRAWLING PADA LINGKUNGAN HADOOP

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 31 Mei 2018



Melinda Nur Abianti
NPM: 2014730012

ABSTRAK

Web crawler merupakan perangkat lunak yang melakukan *web scanning* dan pengindeksan *Uniform Resource Locator* (URL) secara periodis. Salah satu permasalahan dari *web crawler* tradisional adalah ketidak efektifan proses *crawling* dikarenakan perubahan konten *web* setiap periode tidak diperhitungkan untuk menentukan periode waktu *web scanning* selanjutnya. Pada penelitian ini dibangun *web crawler* yang menerapkan salah satu teknik *web crawling*, yaitu *incremental web crawling* untuk menyelesaikan permasalahan tersebut.

Perangkat *incremental web crawling* dibangun berdasarkan rancangan dari A.K. Sharma dan Ashutosh Dixit yang menerapkan algoritma *Self Adjusting Refresh Time Calculator Module* (SARTCM). Algoritma tersebut digunakan untuk menentukan periode waktu *web scanning* URL selanjutnya berdasarkan probabilitas perubahan konten *web* URL setiap periode *crawling*. Informasi perubahan konten *web* dan perubahan periode waktu *crawling* pada setiap versi sebuah URL disimpan pada basis data NoSQL HBase yang memiliki fitur *versioning* untuk penyimpanan sebuah data dengan banyak versi. Untuk meningkatkan kecepatan akses data, basis data HBase dibangun di atas Hadoop untuk melakukan penyimpanan terdistribusi.

Terdapat dua perangkat lunak yang dibangun pada penelitian ini, yaitu perangkat lunak situs induk dan agen *crawler*. Situs induk dibangun menggunakan *platform Java 2 Platform, Enterprise Edition* (J2EE) yang memiliki fitur untuk memasukkan URL untuk di-*crawl*, memasukkan informasi *crawler*, mencari konten berdasarkan URL yang di-*crawl*, melihat *log* pencarian dan proses *crawling*, melihat status URL. Agen *crawler* digunakan untuk melakukan proses *incremental web crawling*. Pengujian fungsional dilakukan untuk menguji fitur pada situs induk, dapat disimpulkan bahwa situs induk berhasil dibangun dan seluruh fungsi berjalan dengan baik. Pengujian performa dilakukan untuk menguji tingkat *scalability* pada HBase yang diterapkan pada agen *crawler* dan situs induk, dapat disimpulkan dari salah satu skenario pengujian bahwa semakin banyak *region server* digunakan, maka baris data URL yang dihasilkan akan semakin besar.

Kata-kata kunci: *web crawler*, *incremental web crawling*, Hadoop, NoSQL HBase, J2EE

ABSTRACT

Web crawler is a software which performs web scanning and *Uniform Resource Locator* (URL) indexing periodically. One of the problems regarding traditional web crawlers is the ineffectiveness of the crawling process due to the change of web content each period is not taken into account to determine the next web scanning time period. To resolve the problem, one of web crawling techniques, incremental web crawling, is being built in this research.

The incremental web crawling software is built based on the design of A.K. Sharma and Ashutosh Dixit that implements *Self Adjusted Refresh Time Calculator Module* (SARTCM) algorithm. The algorithm is used to determine the time period of the next URL's web scanning based on the probability of the URL's web content changes of each crawling period. All version of web content's information and crawling time period of an URL are stored in the HBase NoSQL database that has versioning features for storing a data with multiple versions. To increase the speed of data access, the HBase database is built on top of Hadoop for distributed storage.

There are two software that is built on this research, parent site and agent crawler. Parent site is built on top of *Java 2 Platform, Enterprise Edition* (J2EE) platform that has features to enter URLs for crawling, enter crawler information, search contents based on crawled URLs, views search and crawling processes logs, and views URL's status. Crawler agent is used for performing incremental web crawling processes. Functional testing is used for determine parent site feature, it can be concluded that parent site software runs well. Perform testing is used for testing HBase's scalability feature that is implemented on crawler agent and parent site, based on the result of one of the performance testing scenario, it can be concluded that the more region server used, the number of rows of data URLs generated will be bigger.

Keywords: *web crawler, incremental web crawling, Hadoop, NoSQL HBase, J2EE*

*Dipersembahkan untuk Tuhan, Orang tua, Dosen Pembimbing,
Kekasih, dan Teman-Teman*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas berkat rahmatNya penulis dapat menyelesaikan tugas akhir ini yang berjudul "Incremental Web Crawling pada Lingkungan Hadoop". Adapun tugas akhir ini disusun untuk memenuhi salah satu persyaratan untuk menyelesaikan pendidikan di Fakultas Teknologi Informasi dan Sains pada Program Studi Teknik Informatika di Universitas Katolik Parahyangan Bandung. Dalam menyusun skripsi ini penulis menerima banyak sekali bantuan baik secara jasmani dan rohani dari berbagai pihak yang terlibat baik secara langsung maupun tidak langsung. Oleh karena itu, melalui kesempatan ini penulis hendak mengucapkan terima kasih yang sebesar-besarnya kepada:

- Allah SWT, asal segala kelancaran dan kemudahan yang diberikan-Nya sehingga penulis dapat menyelesaikan kuliahnya tanpa kendala.
- Orang tua penulis yaitu Wiwit Eko Khristianto dan Eva Dianita, serta kakak penulis yaitu Maudy Nur Avianti yang telah mengajarkan kemandirian pada penulis dan mendoakan dan mendukung penulis selama penyusunan skripsi.
- Bapak Gede Karya, M.T., CISA, IPM selaku dosen pembimbing yang telah memberi dorongan mental agar penulis tetap teguh mengerjakan skripsi dan memberi masukan selama penyusunan skripsi.
- Bapak Chandra Wijaya, M.T. selaku dosen penguji utama yang telah memberikan kritik dan saran untuk skripsi, dan Ibu Rosa De Lima, M.Kom. selaku penguji pendamping yang juga telah memberikan kritik dan saran untuk skripsi dan juga rela mencurahkan waktu dan mendukung penulis saat penulis mengalami kegalauan saat menjalankan skripsi.
- Staf Tata Usaha Fakultas Teknologi Informasi dan juga Staf Pegawai yang telah melayani penulis dalam hal administrasi perkuliahan maupun tempat penulis mencurahkan keluh kesah.
- Kekasih penulis yaitu Reanta Indra Putra Pratama, yang telah menemani penulis dalam suka dan duka, teman untuk bertukar pikiran, teman yang memberi kritik, saran, dan dukungan, serta memotivasi penulis untuk segera menyelesaikan skripsi.
- Lina Afriyani, teman terdekat penulis yang selalu menyemangati penulis selama suka dan duka dan memberikan dukungan untuk terus berusaha dalam menyelesaikan tugas akhir dan juga tempat berkeluh kesah.
- Faza Hunafa, teman terdekat penulis yang juga selalu menyemangati penulis selama suka dan duka dan memberikan dukungan untuk terus berusaha dalam menyelesaikan tugas akhir dan juga tempat berkeluh kesah.
- Rekan-rekan di Informatika Unpar, yaitu Agina Rinda, Reza Reynaldi Hasan Haznam, Sapta Hadi Kesuma, Jovanka Helen, Kevin Pratama, Stephanie Tania, Prayogo Cendra, Daud Andrew Gorgha, dan lain-lain; yang telah mendukung untuk menyelesaikan tugas akhir ini hingga penulis lulus.

- Pihak-pihak lain yang belum disebutkan, yang telah memberikan bantuan dalam penyusunan skripsi.

Akhir kata, penulis menyadari bahwa skripsi ini tidak lepas dari kekurangan. Namun penulis berharap skripsi ini dapat memberikan kontribusi baik untuk penelitian atau pembelajaran selanjutnya.

Bandung, Mei 2018

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 <i>Web Crawler</i>	5
2.1.1 Definisi <i>Web Crawler</i>	5
2.1.2 Algoritma <i>Web Crawler</i>	5
2.1.3 Arsitektur <i>Web Crawler</i>	6
2.1.4 Jenis <i>Web Crawler</i>	7
2.2 Struktur Data <i>Queue</i>	7
2.2.1 Definisi <i>Queue</i>	7
2.2.2 Pengaplikasian <i>Queue</i>	8
2.3 <i>Incremental Web Crawler</i>	8
2.3.1 Arsitektur <i>Incremental Web Crawler</i>	9
2.3.2 Penghitungan Frekuensi Perubahan Elemen (P_c)	12
2.4 Hadoop	13
2.4.1 Deskripsi Hadoop	13
2.4.2 Ekosistem Hadoop	14
2.4.3 HDFS	15
2.5 NoSQL HBase	18
2.5.1 <i>Not Only SQL</i> (NoSQL)	18
2.5.2 HBase	19
2.6 Lingkungan Pengembangan Aplikasi Enterprise Berbasis J2EE	24
2.6.1 Servlet	25
2.6.2 <i>Java Server Pages</i> (JSP)	26
2.7 Format Komunikasi <i>Hypertext Transfer Protocol</i> (HTTP)	27
2.7.1 HTTP <i>Request</i>	27
2.7.2 HTTP <i>Response</i>	27
2.8 Jsoup	29
2.9 CSV	30

3	ANALISIS	31
3.1	Deksripsi Masalah	31
3.2	Analisis Algoritma pada Perangkat Lunak	31
3.2.1	Analisis Algoritma <i>Incremental Crawler</i>	32
3.2.2	Analisis Algoritma Pencarian Konten	40
3.3	Analisis Kebutuhan Perangkat Lunak	41
3.3.1	Analisis Kebutuhan Aktor	41
3.3.2	Diagram <i>Use Case</i> dan Skenario Perangkat Lunak	41
3.3.3	Kebutuhan Masukan Perangkat Lunak	45
3.3.4	Kebutuhan Keluaran Perangkat Lunak	47
3.3.5	Kebutuhan Atribut pada Penyimpanan	47
3.3.6	Arsitektur Perangkat Lunak dan Diagram Kelas Sederhana	50
3.4	Eksplorasi Hadoop dan Lingkungannya	54
3.4.1	Instalasi Hadoop	54
3.4.2	Instalasi Zookeeper	56
3.4.3	Instalasi HBase	56
3.4.4	Pembuatan Basis Data dan Manipulasi Data Sederhana	58
4	PERANCANGAN	67
4.1	Perancangan Antarmuka	67
4.2	Perancangan Basis Data Fisik	80
4.3	Diagram Kelas Rinci	82
4.3.1	Diagram Kelas Situs Induk J2EE	82
4.3.2	Diagram Kelas Agen <i>Crawler</i> J2SE	107
5	IMPLEMENTASI DAN PENGUJIAN	117
5.1	Arsitektur Implementasi	117
5.1.1	Lingkungan Implementasi Perangkat Keras	117
5.1.2	Lingkungan Implementasi Perangkat Lunak	117
5.1.3	Implementasi Perangkat Lunak	118
5.2	Pengujian	137
5.2.1	Pengujian Fungsional	137
5.2.2	Pengujian Performa	147
6	KESIMPULAN DAN SARAN	153
6.1	Kesimpulan	153
6.2	Saran	153
	DAFTAR REFERENSI	155
A	KONFIGURASI EKSPLORASI HADOOP DAN LINGKUNGANNYA	157
A.1	Konfigurasi Hadoop	157
A.2	Konfigurasi Zookeeper	160
A.3	Konfigurasi HBase	160
B	KELAS DIAGRAM RINCI	163
B.1	Diagram Kelas Situs Induk J2EE	163
B.2	Diagram Kelas Agen <i>Crawler</i> J2SE	164
C	GAMBAR PENGUJIAN PERFORMA	165
C.1	Lingkungan Pengujian Performa	165
C.2	Informasi Proses <i>Crawling</i>	169
C.3	Hasil Pengujian Performa	170

C.3.1	Pengujian Skenario <i>Crawling</i>	170
C.3.2	Pengujian Skenario Pencarian	171
C.4	Penyimpanan Data HBase pada Hadoop	177
C.5	Bukti Pengujian Dilakukan	179

DAFTAR GAMBAR

2.1	Arsitektur <i>Web Crawler</i> [1]	6
2.2	Penggambaran Proses <i>Enqueue</i> dan <i>Dequeue</i> ¹	8
2.3	Arsitektur <i>Incremental Web Crawler</i> [2]	9
2.4	Ekosistem Hadoop [3]	14
2.5	Arsitektur <i>HDFS</i> [3]	15
2.6	Pembuatan Replikasi	17
2.7	Penggambaran Replikasi	17
2.8	Arsitektur <i>HBase</i> [3]	19
2.9	Struktur Data <i>HBase</i> (1)	20
2.10	Struktur Data <i>HBase</i> (2)	20
2.11	Struktur <i>Column Family</i> dan <i>Column Qualifier</i>	21
2.12	Struktur <i>Versions/Timestamp</i>	21
2.13	Penyimpanan pada <i>HBase</i> [4]	21
2.14	Cara Kerja Servlet <i>container</i> [5]	25
2.15	Siklus Hidup Servlet [5]	26
3.1	Legenda <i>Flowchart</i>	32
3.2	<i>Flowchart</i> Algoritma <i>Incremental Crawler</i>	33
3.3	<i>Flowchart</i> Algoritma Usulan	36
3.4	<i>Flowchart</i> Pencarian URL	40
3.5	<i>Use Case</i> Perangkat Lunak <i>Incremental Crawler</i>	42
3.6	Contoh Ilustrasi <i>Depth</i>	46
3.7	Arsitektur Perangkat Lunak <i>Incremental Crawler</i>	51
3.8	Diagram Kelas Sederhana Situs Induk	53
3.9	Diagram Kelas Sederhana Agen <i>Crawler</i>	54
3.10	Hadoop yang Aktif	55
3.11	Halaman Awal Web Monitor Hadoop	55
3.12	Zookeeper yang Aktif	56
3.13	Folder <i>/hbase</i> di dalam <i>hdfs</i>	57
3.14	<i>HBase</i> yang Aktif	57
3.15	Halaman Awal Web Monitor <i>HBase</i>	58
3.16	Pembuatan Tabel <i>HBase</i> API	59
3.17	Penambahan <i>Column Families</i> pada Tabel <i>HBase</i> API	59
3.18	Hasil Pembuatan Tabel	60
3.19	Mendapatkan Tabel <i>HBase</i> API	61
3.20	Hasil Mendapatkan Tabel	61
3.21	Menambahkan Data <i>HBase</i> API	62
3.22	Mencoba Mengambil Data <i>HBase</i> API	62
3.23	Hasil Mengambil Data	63
3.24	Meng- <i>enable</i> dan <i>Disable</i> Tabel	63
3.25	Hasil <i>Enable</i> dan <i>Disable</i> Tabel	63
3.26	Pendefinisian <i>Version/Timestamp</i> pada <i>HBase</i>	64

3.27	Contoh Pembuatan Data dengan Menambahkan Versi	64
3.28	Contoh Hasil <i>Scan Data Version/Timestamp</i>	64
3.29	Contoh Hasil <i>Scan Data Version/Timestamp</i> dengan <i>Filter</i>	65
3.30	Contoh Data <i>Version Waktu</i>	65
4.1	<i>Layout</i> Halaman Utama	67
4.2	<i>Layout</i> Halaman <i>Sign Up</i>	68
4.3	<i>Layout</i> Halaman <i>Log In</i>	68
4.4	<i>Layout</i> Halaman Utama Admin	69
4.5	<i>Layout Input</i> Informasi URL	70
4.6	<i>Layout Input</i> URL Berhasil	70
4.7	<i>Layout</i> Halaman Cari Konten URL	71
4.8	<i>Layout</i> Halaman Hasil Cari Konten URL	72
4.9	<i>Layout</i> Halaman Detil Cari Konten URL	72
4.10	<i>Layout</i> Halaman Ubah Pengaturan <i>Crawling</i>	73
4.11	<i>Layout</i> Halaman Lihat <i>Log</i>	74
4.12	<i>Layout</i> Halaman <i>Log</i> Pencarian	75
4.13	<i>Layout</i> Halaman Detil <i>Log</i> Pencarian Berdasarkan Tanggal	76
4.14	Informasi <i>Log</i> Pencarian dengan Waktu Spesifik	76
4.15	<i>Layout</i> Halaman <i>Log Crawling</i>	77
4.16	<i>Layout</i> Halaman Detil <i>Log Crawling</i> Berdasarkan Tanggal	78
4.17	Informasi <i>Log Crawling</i> dengan Waktu Spesifik	78
4.18	<i>Layout</i> Halaman Lihat Status URL	79
4.19	<i>Layout</i> Halaman Detil Status URL	80
4.20	Kelas Diagram Kelas HBaseConfig	83
4.21	Kelas Diagram Kelas searchLogJava	83
4.22	Kelas Diagram Kelas UserJava	84
4.23	Kelas Diagram Kelas StatusURLJava	85
4.24	Kelas Diagram Kelas URLSettingJava	86
4.25	Kelas Diagram Kelas URLJava	88
4.26	Kelas Diagram Kelas CrawlLogJava	92
4.27	Kelas Diagram <i>Package Controller</i> Situs Induk J2EE	94
4.28	Kelas Diagram <i>Package WebPage</i> Situs Induk J2EE	97
4.29	Kelas Diagram <i>Package Handler</i> Situs Induk J2EE	99
4.30	Kelas Diagram <i>Package Helper</i> Situs Induk J2EE	101
4.31	Kelas Diagram <i>Package Handler</i> Agen <i>Crawler</i> J2SE	107
4.32	Kelas Diagram <i>Package Helper</i> Agen <i>Crawler</i> J2SE	108
4.33	Kelas Diagram Kelas CrawlProcessor	111
4.34	Kelas Diagram Kelas Main	116
5.1	Arsitektur Implementasi Perangkat Lunak	118
5.2	<i>Layout</i> Halaman Utama	119
5.3	<i>Layout</i> Halaman <i>Sign Up</i>	119
5.4	<i>Layout</i> Halaman <i>Log In</i>	120
5.5	<i>Layout</i> Halaman Utama Admin	120
5.6	<i>Layout Input</i> Informasi URL	121
5.7	<i>Layout Input</i> URL Berhasil	121
5.8	<i>Layout Input</i> URL Gagal	122
5.9	<i>Layout</i> Halaman Cari Konten URL	122
5.10	<i>Layout</i> Halaman Hasil Cari Konten URL	122
5.11	<i>Layout</i> Halaman Detil Cari Konten URL	123
5.12	Nilai konten	123

5.13	<i>Layout</i> Halaman Ubah Pengaturan <i>Crawling</i>	124
5.14	<i>Layout Input</i> Ubah Pengaturan <i>Crawling</i> Berhasil	124
5.15	<i>Layout</i> Halaman Lihat <i>Log</i>	124
5.16	<i>Layout</i> Halaman <i>Log</i> Pencarian	125
5.17	<i>Layout</i> Halaman Detil <i>Log</i> Pencarian Berdasarkan Tanggal	125
5.18	Informasi <i>Log</i> Pencarian dengan Waktu Spesifik	125
5.19	<i>Layout</i> Halaman <i>Log Crawling</i>	126
5.20	<i>Layout</i> Halaman Detil <i>Log Crawling</i> Berdasarkan Tanggal	126
5.21	Informasi <i>Log Crawling</i> dengan Waktu Spesifik	127
5.22	<i>Layout</i> Halaman Lihat Status URL	127
5.23	<i>Layout</i> Halaman Detil Status URL	128
5.24	Informasi Hasil Ekstaksi URL dan <i>Depth</i>	128
5.25	<i>Refresh Time</i> Wikipedia (1)	140
5.26	<i>Refresh Time</i> Wikipedia (2)	140
5.27	Konten Wikipedia (1)	141
5.28	Konten Wikipedia (2)	141
5.29	<i>Child Info</i> Wikipedia	142
5.30	<i>Refresh Time</i> ITB	142
5.31	Konten ITB (1)	143
5.32	Konten ITB (2)	143
5.33	Konten ITB (3)	143
5.34	<i>Child Info</i> ITB	144
5.35	<i>Refresh Time</i> Kompas	145
5.36	Konten Kompas (1)	145
5.37	Konten Kompas (2)	146
5.38	<i>Child Info</i> Kompas	146
5.39	Skema Komunikasi antar Komputer	148
5.40	Skema Konfigurasi Jaringan	148
5.41	Hasil Pengujian Performa <i>Crawler</i>	149
5.42	Hasil Pengujian Performa Pencarian Kata	150
C.1	Sistem Operasi Komputer yang Digunakan	165
C.2	Informasi NameNode yang Digunakan	166
C.3	Informasi DataNode yang Digunakan	166
C.4	Informasi <i>Region Server</i> pada HBase (3 <i>Region Server</i> Aktif)	167
C.5	Kondisi MasterHBase (3 <i>Region Server</i> Aktif)	167
C.6	Kondisi slave1 (3 <i>Region Server</i> Aktif)	167
C.7	Kondisi slave2 (3 <i>Region Server</i> Aktif)	168
C.8	Kondisi slave3 (3 <i>Region Server</i> Aktif)	168
C.9	Informasi <i>seed</i> URL dan Pengaturan <i>Crawler</i> yang Digunakan	169
C.10	Proses <i>Crawling</i> pada Sebuah Komputer	170
C.11	Banyak URL dengan 1 <i>Region Server</i>	170
C.12	Banyak URL dengan 2 <i>Region Server</i>	170
C.13	Banyak URL dengan 3 <i>Region Server</i>	170
C.14	Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata hati	171
C.15	Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata hati	171
C.16	Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata hati	171
C.17	Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata abu	171
C.18	Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata abu	171
C.19	Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata abu	172
C.20	Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata bulan	172
C.21	Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata bulan	172

C.22 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata bulan	172
C.23 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata gesit	172
C.24 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata gesit	173
C.25 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata gesit	173
C.26 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata langit	173
C.27 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata langit	173
C.28 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata langit	173
C.29 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata light	174
C.30 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata light	174
C.31 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata light	174
C.32 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata lincah	174
C.33 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata lincah	175
C.34 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata lincah	175
C.35 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata mahasiswa	175
C.36 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata mahasiswa	175
C.37 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata mahasiswa	175
C.38 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata terima	176
C.39 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata terima	176
C.40 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata terima	176
C.41 Waktu Pencarian Konten dengan 1 <i>Region Server</i> atas nama kata tidak	176
C.42 Waktu Pencarian Konten dengan 2 <i>Region Server</i> atas nama kata tidak	176
C.43 Waktu Pencarian Konten dengan 3 <i>Region Server</i> atas nama kata tidak	177
C.44 Penyimpanan Data HBase di Hadoop	177
C.45 Contoh Penyimpanan Tabel url_hbase pada file Hadoop (1)	178
C.46 Contoh Penyimpanan Tabel url_hbase pada file Hadoop (2)	178
C.47 Bukti Penulis Melakukan Pengujian	179
C.48 Komputer yang Digunakan untuk Pengujian Performa	179

DAFTAR TABEL

4.1	Perancangan Fisik pada Tabel URL	81
4.2	Perancangan Fisik pada Tabel URL <i>Setting</i>	81
4.3	Perancangan Fisik pada Tabel Status URL	81
4.4	Perancangan Fisik pada Tabel <i>User Admin</i>	81
4.5	Perancangan Fisik pada Tabel <i>Log Search</i>	82
4.6	Perancangan Fisik pada Tabel <i>Log Crawl</i>	82
5.1	Daftar Implementasi Kelas	131
5.2	Pengujian Fungsional Situs Induk J2EE	138
5.3	Konfigurasi IP 4 Komputer	147
5.4	Waktu Pencarian Kata per <i>Region Server</i> (detik)	150

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Web crawler merupakan perangkat lunak yang melakukan *web scanning* dan membuat indeks dari *Uniform Resource Locator* (URL) yang dicari dengan metode tertentu. *Web crawler* dapat digunakan di berbagai tempat, contohnya di dalam *search engine* untuk mengumpulkan informasi yang berguna saat mencari data, sebagai perangkat lunak untuk menyalisir *web* untuk mencari kata/kumpulan kata tertentu yang digunakan, dan melihat *trend* pasar atas dasar pencarian kata tertentu.

Dalam melakukan *crawling*, *web crawler* memiliki metode-metode yang dapat digunakan, yang masing-masing memiliki kelebihan tersendiri. Contohnya adalah *distributed crawling*, *incremental crawling*, *focused crawling*, *deep crawling*, dan lainnya. Metode *incremental crawling* merupakan metode *crawling* yang secara *increment* memperbaharui konten *web* dengan durasi waktu yang tidak tetap, tergantung dari seberapa banyak perubahan konten *web* yang telah terjadi sebelumnya. Jika dibandingkan dengan metode *crawling* tradisional yang secara periodik mengganti dokumen lama dengan dokumen baru, penggunaan metode *incremental crawling* berguna untuk mengurangi nilai probabilitas konten yang sama dibandingkan konten terdahulunya, yang mengakibatkan proses *crawling* dianggap tidak terlalu efektif karena mengunduh informasi konten yang sama.

Pada penelitian ini, dibuat perangkat lunak yang menerapkan metode *incremental crawling* untuk menangani salah satu permasalahan *crawling*, yaitu perubahan konten *web* seiring waktu. Perubahan konten dalam suatu *web* belum tentu dilakukan secara periodik dan pada saat *crawling* dilakukan, konten tidak dapat dipastikan pengubahannya. Hal ini mengakibatkan metode *crawling* tradisional menjadi tidak efisien. Maka dari itu, penelitian ini menggunakan *incremental crawling* sebagai acuan metode *crawling* untuk menangani permasalahan yang telah dijelaskan di atas.

Algoritma *incremental crawling* pada penelitian ini diimplementasikan berdasarkan rancangan A.K. Sharma dan Ashutosh Dixit. Pada rancangan tersebut, diimplementasikan algoritma *Self Adjusting Refresh Time Calculator Module* (SARTCM). Algoritma tersebut berfungsi untuk menentukan frekuensi *refresh time* sebuah URL pada setiap periode proses *crawling* berdasarkan probabilitas perubahan elemen (p_c), batas bawah probabilitas perubahan elemen (p_l), dan batas atas probabilitas perubahan elemen (p_g) yang didapatkan dari akumulasi seluruh periode frekuensi perubahan elemen konten web URL tersebut. Probabilitas perubahan elemen pada sebuah konten web dapat ditentukan dengan mengamati perubahan konten web saat ini dengan konten web pada periode-periode *crawling* sebelumnya, juga dapat ditentukan dengan mengamati elemen pada *Hypertext Transfer Protocol* (HTTP) Response URL, misalnya adalah elemen ETag dan Last-Modified.

Perangkat lunak dibuat dengan bantuan *framework* Hadoop. Hadoop adalah *framework open source* berbasis Java di bawah lisensi Apache untuk *support* aplikasi yang berjalan pada *big data*. Hadoop menyediakan beragam ekosistem yang memiliki beragam kegunaan berbeda. Di antara beragam ekosistem tersebut, terdapat ekosistem yang menyediakan mekanisme *versioning* yang menyimpan versi pembaharuan dari suatu nilai yang dimasukkan ke dalam suatu sel dalam basis data Hadoop, yang berguna dalam menyelesaikan permasalahan perubahan konten seiring waktu. Dengan adanya *versioning*, dapat diketahui kapan saja dan perubahan apa saja yang terjadi

dalam suatu konten *web*, sehingga membantu pembuatan perangkat lunak yang menerapkan metode *incremental crawling*. Selain itu, HBase memiliki sifat *scalable* yang dapat dimanfaatkan untuk mengolah data berukuran besar, salah satunya adalah mengolah informasi URL pada perangkat lunak yang dibangun.

Pada penelitian ini, dilakukan pengimplementasian *incremental crawling* pada perangkat lunak. Di dalam perangkat lunak tersebut, pengguna dapat masuk sebagai admin untuk memasukkan URL untuk di-*crawl*, melihat hasil proses *incremental crawling* dari URL yang dimasukkan, dan memasukkan informasi *crawler*. Selain itu, pengguna juga dapat mencari konten yang terkandung pada URL yang di-*crawl* tanpa perlu melakukan akses sebagai admin.

1.2 Rumusan Masalah

Berikut ini adalah rumusan masalah dari penelitian ini:

1. Bagaimana mekanisme *incremental crawling* untuk menyelesaikan permasalahan perubahan konten *web* seiring waktu?
2. Bagaimana mekanisme penyimpanan informasi *versioning* pada ekosistem Hadoop untuk menyelesaikan permasalahan perubahan konten *web* seiring waktu?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka tujuan dari penelitian ini adalah:

1. Memahami mekanisme dan hasil dari *incremental crawling* untuk menyelesaikan permasalahan perubahan konten *web* seiring waktu.
2. Memahami mekanisme penyimpanan informasi *versioning* pada ekosistem Hadoop untuk menyelesaikan permasalahan perubahan konten *web* seiring waktu.

1.4 Batasan Masalah

Rumusan masalah yang telah disebutkan di atas masih memiliki ruang lingkup yang cukup luas. Karena keterbatasan waktu dan kemampuan yang dimiliki, maka penelitian ini hanya memfokuskan pada batasan masalah sebagai berikut:

1. Informasi yang diperoleh pada hasil *crawling* bersifat tekstual. Oleh karena itu, konten hasil *crawling* yang dapat dicari bersifat tekstual.
2. Informasi data diri (nama, *username*, dan *password*) dari admin perangkat lunak tidak dapat diperbaharui untuk menyederhanakan fungsionalitas perangkat lunak.
3. Satuan *refresh time* pada proses *crawling* memiliki satuan menit.
4. Penghitungan p_1 dan p_g menggunakan metode penghitungan kuartil Mendenhall and Sincich.
5. Nilai HTTP *Response* yang digunakan untuk memeriksa perubahan elemen adalah ETag dan Last-Modified.
6. Konten *Hypertext Markup Language* (HTML) yang disimpan pada basis data merupakan seluruh elemen yang berada pada *tag body* HTML.
7. Fokus pengerjaan pada penelitian ini adalah melihat perubahan *refresh time* setiap versi URL berdasarkan perubahan elemen yang terjadi pada sebuah URL.

1.5 Metodologi

Berikut ini adalah langkah-langkah yang dilakukan dalam membangun perangkat lunak:

1. Studi pustaka mengenai konsep *web crawler*.
2. Studi pustaka dan eksplorasi mengenai sistem terdistribusi berbasis Hadoop.
3. Studi pustaka mengenai *Java 2 Platform, Enterprise Edition (J2EE)*
4. Analisis pemilihan *database* pada ekosistem Hadoop.
5. Studi teknik mengenai *incremental crawling*.
6. Studi penyimpanan informasi *versioning* dan penyimpanannya pada ekosistem Hadoop.
7. Merancang dan mengembangkan sistem perangkat lunak *web crawler*, yang terdiri atas situs induk dan agen *crawler*.
8. Menguji sistem perangkat lunak *web crawler* baik fungsional maupun *performance*-nya.
9. Menulis dokumen penelitian.

1.6 Sistematika Pembahasan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

- Bab 1 Pendahuluan
Bab ini membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, serta sistematika pembahasan.
- Bab 2 Dasar Teori
Bab ini membahas tentang teori-teori dasar yang menunjang penelitian yang dilakukan yaitu teori *web crawler*, struktur data *queue*, *incremental web crawler*, Hadoop, *Not only SQL (NoSQL)* HBase, J2EE, format komunikasi HTTP, dan Jsoup.
- Bab 3 Analisis
Bab ini membahas tentang analisis mengenai teknik *incremental crawling* yang digunakan untuk melakukan *incremental crawling*, analisis perangkat lunak, serta hasil eksplorasi mengenai Hadoop dan lingkungannya yang terdiri atas Hadoop, Zookeeper, dan HBase.
- Bab 4 Perancangan
Bab ini membahas tentang perancangan yang dibutuhkan untuk membangun perangkat lunak, yaitu perancangan antarmuka, perancangan basis data, diagram kelas rinci dan fungsi di dalamnya.
- Bab 5 Implementasi dan Pengujian
Bab ini membahas tentang lingkungan implementasi, implementasi perangkat lunak (baik secara antarmuka, basis data, maupun fungsi), pengujian fungsional, pengujian *performance*, dan kesimpulan hasil pengujian.
- Bab 6 Kesimpulan dan Saran
Bab ini membahas tentang kesimpulan dari penelitian yang dilakukan dan saran untuk pengembangan penelitian selanjutnya.