



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - 141501

IDENTIFIKASI KARAKTERISTIK DATASET UNTUK FE- DERATED SPARQL QUERY

DATASET CHARACTERISTICS IDENTIFICATION FOR FE- DERATED SPARQL QUERY

LUTFI NUR FADZILAH
NRP 5213100059

Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

Halaman ini sengaja dikosongkan

TUGAS AKHIR - 141501

IDENTIFIKASI KARAKTERISTIK DATASET UNTUK FE- DERATED SPARQL QUERY

LUTFI NUR FADZILAH
NRP 5213100059

Dosen Pembimbing

Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

Halaman ini sengaja dikosongkan

UNDERGRADUATE THESIS - 141501

DATASET CHARACTERISTICS IDENTIFICATION FOR FEDERATED SPARQL QUERY

LUTFI NUR FADZILAH

NRP 5213100059

Supervisor

Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

DEPARTMENT OF INFORMATION SYSTEM

Faculty of Information Technology

Institut Teknologi Sepuluh Nopember

Surabaya, 2017

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**IDENTIFIKASI KARAKTERISTIK DATASET UNTUK
FEDERATED SPARQL QUERY**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Analisa Data dan Diseminasi Informasi
Program Studi S1 Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

LUTEFI NUR FADZILAH

NRP: 5213100059

Surabaya, September 2017

**KETUA
DEPARTEMEN SISTEM INFORMASI**

Dr. Ir. Aris Tjahyanto, M.Kom.

NIP. 19650310 199102 1 001

Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

**IDENTIFIKASI KARAKTERISTIK DATASET UNTUK
FEDERATED SPARQL QUERY**

TUGAS AKHIR

**Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Analisa Data dan Diseminasi Informasi
Program Studi SI Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh :

LUTFI NUR FADZILAH

NRP: 5213100059

**Disetujui Tim Penguji: Tanggal Ujian: 12 Juli 2017
Periode Wisuda: September 2017**

**Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D (Pembimbing
1)**

Faizal Johan Atletiko, S.Kom., M.T.

Renny Pradina K., S.T., M.T.

(Handwritten signature)
(Penguji 1)

(Handwritten signature)
(Penguji 2)

Halaman ini sengaja dikosongkan

IDENTIFIKASI KARAKTERISTIK DATASET UNTUK FEDERATED SPARQL QUERY

Nama : LUTFI NUR FADZILAH
NRP : 5213100059
Departemen : Sistem Informasi FTIf
Pembimbing I : Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Abstrak

Saat ini telah dikembangkan federated SPARQL query engine yang mempunyai kemampuan untuk melakukan query dari beberapa SPARQL endpoint yang terdistribusi, sehingga data yang berasal berbagai sumber memungkinkan untuk diperoleh. Ketika dijalankan untuk melakukan query, masing-masing query engine mempunyai kinerja yang berbeda-beda. Salah satu faktor yang berpengaruh terhadap kinerja dari query engine adalah karakteristik dari dataset RDF yang diakses, seperti jumlah triple, kelas, property, subjek, entity, objek, dan spreading factor dataset. Tugas Akhir ini dilakukan untuk mengidentifikasi karakteristik dataset RDF serta mengetahui karakteristik dataset yang berpengaruh terhadap kinerja dari query engine. Penelitian dilakukan dengan mengidentifikasi 10 dataset yang diambil dari jurnal penelitian lain. Sedangkan uji coba untuk mengetahui keterkaitan antara karakteristik dataset dengan kinerja dari query engine dilakukan menggunakan federated SPARQL query engine FedX. Dari hasil analisis, diketahui bahwa jumlah triple dan jumlah kelas yang terkait dengan query cenderung berpengaruh terhadap kinerja dari query engine. Sedangkan jumlah property yang terkait dengan dataset, spreading factor dataset, dan spreading factor dataset yang terkait dengan query cenderung tidak berpengaruh terhadap kinerja dari query engine.

Kata kunci: linked data, federated SPARQL query engine, dataset, RDF.

Halaman ini sengaja dikosongkan

DATASET CHARACTERISTICS IDENTIFICATION FOR FEDERATED SPARQL QUERY

Name : LUTFI NUR FADZILAH
NRP : 5213100059
Major : Information System FTIf
Supervisor I : Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Abstract

Federated SPARQL query engines that are able to query from multiple distributed SPARQL endpoints have been developed, so that data from multiple sources are possible to obtain. When it is used to execute a query, a query engine usually has different performance compared to the others. One of the factors that affect the performance of the query engine is the characteristic of the accessed RDF dataset, such as the number of triples, the number of classes, the number of properties, the number of subjects, the number of entities, the number of objects, and the spreading factor of dataset. This final project is done to identify the characteristic of RDF dataset and to know dataset characteristic which is able influence the performance of query engine. The study was conducted by identifying 10 datasets taken from other research journals. The test to determine the relationship between dataset characteristics and the performance of the query engine is done using federated SPARQL query engine FedX. From the analysis results, it is known that the number of triples and the number of classes associated with the query tend to affect the performance of the query engine. Meanwhile, the number of properties associated with the query, spreading factor of dataset, and spreading factor of dataset associated with the query tend not to have an effect on performance of query engine.

Keywords: linked data, federated SPARQL query engine, dataset, RDF.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Identifikasi Karakteristik Dataset untuk Federated SPARQL Query” dengan tepat waktu.

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi nyata bagi kampus Sistem Informasi, ITS, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin menyampaikan terimakasih kepada:

1. Ibu Nur Aini Rakhmawati, S.Kom., M.Sc., Eng. selaku dosen pembimbing penulis yang telah memberikan ide, bimbingan, saran, kritik, ilmu, dan pengalamannya yang sangat bermanfaat sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Bapak Bakti Cahyo Hidayanto, S.Si., M.Kom. selaku dosen wali penulis yang selalu membimbing dan memberikan arahan ke penulis.
3. Bapak Dr. Ir. Aris Tjahyanto, M.Kom. selaku Kepala Jurusan Sistem Informasi yang telah memberikan ilmu dan pengalaman kepada penulis.
4. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu pengetahuan dan pengalaman yang sangat berharga dan bermanfaat bagi penulis.
5. Seluruh keluarga besar saya khususnya Bapak, Ibu dan Kakak saya yang telah memberikan dukungan baik material maupun non material serta semangat kepada penulis hingga akhirnya

dapat menyelesaikan tugas akhir ini.

6. Teman-teman di Jurusan Sistem Informasi yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir.
7. Serta seluruh pihak-pihak lain yang tidak dapat disebutkan satu per satu yang telah banyak membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Tugas Akhir ini juga masih jauh dari kata sempurna, sehingga penulis mengharapkan saran dan kritik yang membangun dari pembaca untuk perbaikan ke depan. Semoga Tugas Akhir ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan semua pihak.

DAFTAR ISI

ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR TABEL	xxi
DAFTAR GAMBAR	xxiii
DAFTAR KODE	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Relevansi	4

2	TINJAUAN PUSTAKA	5
2.1	Penelitian Sebelumnya	5
2.2	Dasar Teori	8
2.2.1	RDF	8
2.2.2	Linked Data	10
2.2.3	SPARQL Query	11
2.2.4	Dataset RDF	17
2.2.5	Spreading Factor	18
2.2.6	FedX	20
3	METODOLOGI	23
3.1	Tahapan Pengerjaan Tugas Akhir	23
3.1.1	Studi Literatur	23
3.1.2	Pengumpulan Data	24
3.1.3	Identifikasi Karakteristik Dataset	25
3.1.4	Pembuatan Query	25
3.1.5	Pengujian	27
4	PERANCANGAN	29
4.1	Pengumpulan Dataset	29
4.2	Praproses Dataset	31
4.3	Identifikasi Karakteristik Dataset	31
4.3.1	Spreading Factor	37
4.4	Pembuatan Query	38
4.4.1	Hasil Pembuatan Query	41
4.4.2	Jumlah Kelas dan Property yang Terlibat dalam Query	43
4.4.3	Spreading Factor Berdasarkan Query	44
5	IMPLEMENTASI	49

5.1	Lingkungan Implementasi	49
5.2	Pengujian	50
5.2.1	Pengaturan <i>Federation</i> pada FedX	50
5.2.2	Eksekusi Query	54
6	HASIL DAN PEMBAHASAN	59
6.1	Hasil Pengujian	59
6.2	Pembahasan	61
6.2.1	Response Time dan Bandwidth Query 1, 2, dan 3 vs Triple	63
6.2.2	Response Time dan Bandwidth Query 1, 2, dan 3 vs Spreading Factor	66
6.2.3	Rata-Rata Response Time Query vs Triple .	70
6.2.4	Rata-Rata Bandwidth Query vs Triple . . .	71
6.2.5	Rata-Rata Response Time Query vs Spreading Factor	72
6.2.6	Rata-Rata Bandwidth Query vs Spreading Factor	73
6.2.7	Rata-Rata Response Time Query vs Spreading Factor Query	74
6.2.8	Rata-Rata Bandwidth Query vs Spreading Factor Query	75
7	KESIMPULAN DAN SARAN	77
7.1	Kesimpulan	77
7.2	Saran	78
	DAFTAR PUSTAKA	81
A	Kode Program	83

B Daftar Kelas dan Property Terbanyak pada Semua Dataset	85
C Hasil Pengujian	89
BIODATA PENULIS	93

DAFTAR TABEL

2.1	Hasil query SELECT setelah dijalankan	12
3.1	Daftar Jurnal dan Dataset yang Dipakai dalam Penelitian	26
4.1	Daftar dataset beserta sumber dan SPARQL endpoint	30
4.2	Hasil Identifikasi Karakteristik Dataset	37
4.3	Hasil Identifikasi Spreading Factor	39
4.4	Hasil Spreading Factor Berdasarkan Query	46
4.5	Rata-Rata dari Karakteristik Dataset Tiap Jurnal	47
5.1	Spesifikasi Perangkat Keras	49
5.2	Spesifikasi Perangkat Lunak	50
5.3	Pengaturan Federation pada Masing-Masing Jurnal	53
6.1	Hasil Pengujian Setelah Dirata-rata	62
6.2	Jumlah Kelas dan Property Tiap Jurnal yang Terkait dengan Query	69
B.2	Daftar Property Terbanyak	85
B.1	Daftar Kelas Terbanyak	85
B.2	Daftar Property Terbanyak	86
B.2	Daftar Property Terbanyak	87
B.2	Daftar Property Terbanyak	88

C.1 Hasil Pengujian Pertama	89
C.2 Hasil Pengujian Kedua	90
C.3 Hasil Pengujian Ketiga	91

DAFTAR GAMBAR

2.1	Hubungan "subjek-predikat-objek" [7]	9
2.2	Arsitektur FedX [12]	21
3.1	Alur pengerjaan tugas akhir	24
3.2	Ilustrasi pengujian dataset	27
4.1	Alur identifikasi 1	34
4.2	Alur identifikasi 2	34
4.3	Alur identifikasi 3	35
4.4	Alur identifikasi 4	36
4.5	Alur identifikasi 5	36
4.6	Alur mencari kelas dan property terbanyak	40
4.7	Star Query	41
4.8	Chain Query	42
5.1	Pengaturan Federation	51
5.2	Tampilan dari FedX	56
5.3	Tampilan dari NetBalancer	57
6.1	Grafik Response Time Query 1, 2, dan 3 vs Triple	63
6.2	Grafik Bandwidth Query 1, 2, dan 3 vs Triple	64
6.3	Grafik Response Time Query 1, 2, dan 3 vs Spreading Factor	66
6.4	Grafik Bandwidth Query 1, 2, dan 3 vs Spreading Factor	67

6.5	Grafik Rata-Rata Response Time Query vs Triple . . .	70
6.6	Grafik Rata-Rata Bandwidth Query vs Triple	71
6.7	Grafik Rata-Rata Response Time Query vs Spreading Factor	72
6.8	Grafik Rata-Rata Bandwidth Query vs Spreading Factor	73
6.9	Grafik Rata-Rata Response Time Query vs Spreading Factor Query	74
6.10	Grafik Rata-Rata Bandwidth Query vs Spreading Factor Query	75

DAFTAR KODE

2.1	Struktur SPARQL	11
2.2	Query SELECT	12
2.3	Query ASK	12
2.4	Query CONSTRUCT	13
2.5	Hasil query CONSTRUCT setelah dijalankan . . .	13
2.6	Query DESCRIBE	15
2.7	Hasil query DESCRIBE setelah dijalankan	15
2.8	Contoh dataset	17
4.1	Jumlah triples	32
4.2	Jumlah entity	32
4.3	Jumlah kelas	32
4.4	Jumlah property	32
4.5	Jumlah distinct subjek	32
4.6	Jumlah distinct objek	33
4.7	Daftar kelas	33
4.8	Daftar property	33
4.9	Daftar subjek	33
4.10	Daftar entity	33
4.11	Daftar objek	33
4.12	Daftar kelas	38
4.13	Daftar property	38
4.14	Star Query 1	41
4.15	Star Query 2	42
4.16	Chain Query	42

4.17	Query untuk Mencari Jumlah Kelas foaf:person . .	44
4.18	Query untuk Mencari Jumlah Property rdf:type . .	44
5.1	Contoh Pengaturan Federation	52
5.2	Eksekusi Query	54
A.1	<i>Script</i> untuk Konversi Dataset GeoNames ke dalam Format N-Triples	83

BAB 1

PENDAHULUAN

Pada bab pedahuluan ini akan membahas terkait latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan relevansi terhadap pengerjaan tugas akhir.

1.1 Latar Belakang

Penemuan Semantic Web telah membuat pertukaran data pada web dapat dilakukan dengan lebih mudah. Semantic Web telah memunculkan standar baru yang membuat data-data pada web yang tersebar di seluruh dunia menjadi lebih terstruktur dan saling terhubung. Semantic Web juga membuat data dapat dimengerti oleh mesin sehingga memungkinkan pemanfaatan data oleh aplikasi.

Framework standar pertukaran data yang digunakan pada Semantic Web adalah RDF (Resource Description Framework). Saat ini telah banyak database online yang dibuat dalam format RDF, antara lain DBpedia dan Freebase. Untuk dapat memperoleh data RDF dari sumber tersebut, perlu dilakukan query menggunakan bahasa query spesifik untuk RDF, yaitu SPARQL (SPARQL Protocol and RDF Query Language). Sedangkan layanan web yang memungkinkan query data RDF melalui SPARQL disebut SPARQL endpoints.

Dari tahun ke tahun, jumlah data yang dipublikasikan dalam format RDF semakin banyak. Untuk mengimbangi perkembangan data RDF yang begitu pesat, maka dibuatlah sistem aplikasi yang memungkinkan query data RDF dari beberapa sumber data. Query data RDF dari beberapa SPARQL endpoint dapat dilakukan menggu-

nakan aplikasi yang disebut federated SPARQL query engine. Contoh dari federated SPARQL query engine antara lain FedX, ANA-PSID, dan ADERIS.

Ketika digunakan untuk melakukan query data RDF, query engine tersebut menampilkan kinerja yang berbeda-beda. Suatu query engine mungkin memiliki kinerja yang baik untuk melakukan query pada sumber data A, namun lambat ketika digunakan untuk melakukan query pada sumber data B. Sedangkan query engine lain mungkin bekerja sebaliknya, dimana memiliki kinerja yang baik untuk melakukan query pada sumber data B, namun lambat ketika digunakan untuk melakukan query pada sumber data A. Oleh karena itu, sulit untuk menentukan query engine mana yang paling baik, karena masing-masing query engine memiliki kinerja yang berbeda-beda. Salah satu faktor yang menyebabkan perbedaan kinerja tersebut ialah karakteristik dari dataset RDF yang diakses oleh query engine tersebut. Dataset RDF tentu memiliki karakteristik yang berbeda-beda antara dataset satu dengan yang lain, seperti jumlah triple dan kelas yang dimiliki. Perbedaan tersebut bisa berdampak pada kinerja query engine yang bervariasi ketika digunakan untuk mengakses dataset tersebut. Oleh karena itu, perlu dilakukan penelitian untuk mengetahui keterkaitan antara karakteristik dataset dengan kinerja dari federated SPARQL query engine yang mengaksesnya. Pada penelitian kali ini akan dilakukan identifikasi karakteristik dataset yang digunakan pada federated SPARQL query dan mengetahui keterkaitan antara karakteristik dataset dengan kinerja dari query engine.

Penelitian ini dimulai dengan identifikasi karakteristik yang dimiliki oleh dataset. Dataset yang digunakan pada penelitian ini diambil dari jurnal-jurnal penelitian sebelumnya yang memanfaatkan Linked Data dan memakai federated repository. Setelah karakteristik dataset berhasil diketahui, dilakukan pengujian dataset dengan cara mengeksekusi query pada federated SPARQL query engine Fe-

dX. Proses pengujian ini bertujuan untuk mengetahui apakah terdapat hubungan antara karakteristik dataset dengan kinerja dari query engine.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, tugas akhir yang akan diajukan ini menitikberatkan permasalahan pada beberapa hal sebagai berikut:

1. Bagaimana mengidentifikasi karakteristik dataset RDF berdasarkan metrik yang ditentukan?
2. Bagaimana karakteristik dataset RDF berpengaruh pada kinerja dari federated SPARQL query engine?

1.3 Batasan Masalah

Batasan-batasan dalam pembuatan Usulan Tugas Akhir adalah sebagai berikut:

1. Pengujian dataset RDF pada penelitian dilakukan menggunakan federated SPARQL query engine FedX.
2. Dataset RDF yang digunakan pada penelitian ini terbatas pada 10 dataset jurnal penelitian yang telah dipilih sebelumnya.

1.4 Tujuan

Tujuan dari penelitian ini adalah melakukan identifikasi karakteristik dataset RDF serta mengetahui keterkaitan antara karakteristik

dataset dengan kinerja dari federated SPARQL query engine.

1.5 Manfaat

Manfaat yang akan diperoleh dengan tugas akhir ini antara lain:

1. Bagi pengguna, dapat mengetahui proses identifikasi karakteristik dataset RDF serta dapat mengetahui apakah terdapat keterkaitan antara karakteristik dataset RDF dengan kinerja dari federated SPARQL query engine.
2. Bagi peneliti, hasil dari penelitian ini dapat dimanfaatkan sebagai referensi untuk pengembangan penelitian ini di masa mendatang.

1.6 Relevansi

Tugas akhir ini berkaitan dengan mata kuliah Pemrograman Integratif dan Pemrograman Berorientasi Objek. Tugas akhir ini layak dijadikan sebagai tugas akhir pada tingkat S1, karena tugas ini memecahkan masalah yang berhubungan dengan materi pada tingkat S1, yaitu dalam hal mencari keterkaitan antara karakteristik dataset RDF dengan kinerja dari federated SPARQL query engine yang mengaksesnya.

BAB 2

TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Dasar teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir ini.

2.1 Penelitian Sebelumnya

- Penelitian pertama dengan judul On Metrics For Measuring Fragmentation Of Federation over SPARQL Endpoints [10] oleh Nur Aini Rakhmawati, Marcel Karnstedt, Michael Hausenblas, dan Stefan Decker. Penelitian dilakukan untuk mengetahui keterkaitan antara distribusi data dan biaya komunikasi pada federated SPARQL query. Penulis mengenalkan metrik baru yang disebut dengan Spreading Factor yang berfungsi untuk mengetahui persebaran class dan property pada suatu dataset. Kemudian, untuk mengetahui keterkaitan antara spreading factor dengan biaya komunikasi, penulis melakukan evaluasi dengan menjalankan static query pada 9 dataset dengan berbagai jenis fragmentasi dalam batas waktu 1 jam, lalu dilakukan penghitungan biaya komunikasi. Hasil penelitian menunjukkan bahwa Spreading Factor yang kecil dapat meminimalkan biaya komunikasi, namun juga dapat mengurangi performa dari SPARQL endpoint. Selain itu, Spreading Factor yang kecil dapat mengakibatkan tingginya request yang diterima SPARQL endpoint. SPARQL endpoint yang menyimpan predikat yang populer juga akan menerima lebih banyak request. Hal ini dapat berakibat pada hasil qu-

ery yang tidak lengkap karena terjadi overloaded.

- Penelitian kedua dengan judul Querying over Federated SPARQL Endpoints – A State of the Art Survey [11] oleh Nur Aini Rakhmawati, Jurgen Umbrich, Marcel Karnstedt, Ali Hasnain dan Michael Hausenblas. Pada penelitian ini, penulis memberikan overview mengenai infrastruktur untuk melakukan query linked data serta langkah-langkah mengenai federation pada SPARQL Endpoints. Kemudian penulis melakukan perbandingan beberapa SPARQL federation framework. Berdasarkan perbandingan yang telah dilakukan, diketahui bahwa masing-masing federation framework mendukung fitur yang berbeda-beda. Penulis mengusulkan fitur berupa Hybrid Data Catalogue dan Link Predicate Awareness yang dapat ditambahkan untuk pengembangan lebih jauh terhadap federation framework. Penulis juga menjelaskan tantangan yang dihadapi pada saat penelitian yang perlu untuk diselesaikan pada pengembangan federation framework di masa mendatang, meliputi data access and security, data allocation, data freshness, benchmark, overlapping terminologies, dan provenance.
- Penelitian ketiga berjudul Benchmarking Federated SPARQL Query Engines: Are Existing Testbeds Enough? [9] oleh Gabriela Montoya, Maria-Esther Vidal, Oscar Corcho, Edna Ruckhaus, dan Carlos Buil-Aranda. Penulis melakukan benchmarking pada tiga federated query engine, yaitu ARQ, ANAPSID, dan FedX. Benchmarking dilakukan dengan mengukur Endpoint Selection Time, Execution Time, dan Answer Completeness. Selain itu juga dilakukan penambahan variabel dan konfigurasi, seperti query baru, konfigurasi network latency baru, dan parameter distribusi dataset baru untuk menghasilkan informasi yang lebih akurat. Hasil penelitian menunjukkan bahwa hasil benchmark dengan parameter tertentu menunjukkan hasil yang berbeda-beda dari masing-

masing federated query engine. Benchmark dengan parameter tertentu juga mampu memunculkan perilaku dari federated query engine yang tidak pernah teramati sebelumnya. Penelitian seperti ini masih perlu untuk dikembangkan lebih jauh dengan penambahan variabel dan konfigurasi baru untuk menghasilkan hasil yang semakin akurat.

- Penelitian keempat berjudul Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets [5] oleh Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, dan Octavian Udrea. Penulis melakukan perbandingan antara RDF benchmark dataset dengan RDF dataset asli yang umum digunakan. RDF benchmark dataset yang dipakai adalah TPC-H, BSBM dataset, LUBM dataset, dan SP2Bench Dataset. Sedangkan RDF dataset asli yang digunakan yaitu DBpedia, UniProt, YAGO, Barton Library Dataset, WordNet, dan Linked Sensor Dataset. Metrik yang dipakai untuk perbandingan meliputi disk space, jumlah triple, average outdegree, outdegree distribution, average indegree, indegree distribution, jumlah subjek, jumlah objek, jumlah property, jumlah type, average type properties, type properties distribution, average number of instances per type, dan instance distribution. Hasil penelitian menunjukkan metrik primitif yang digunakan untuk perbandingan tidak cukup untuk menggambarkan perbedaan antara benchmark dataset dan dataset asli. Maka dari itu, penulis mengenalkan metrik yang disebut coherence, yang mengkombinasikan metrik-metrik primitif dan digunakan untuk mengukur tingkat kestrukturan dari dataset. Selain itu, benchmark dataset memiliki keterbatasan dalam hal tingkat kestrukturan, dimana tingkat kestrukturan merupakan hal yang penting. Oleh karena itu, penulis mengenalkan benchmark generator yang mampu membuat benchmark dataset yang meniru karakteristik dataset yang sebenarnya dalam hal struktur, ukuran, dan isi.

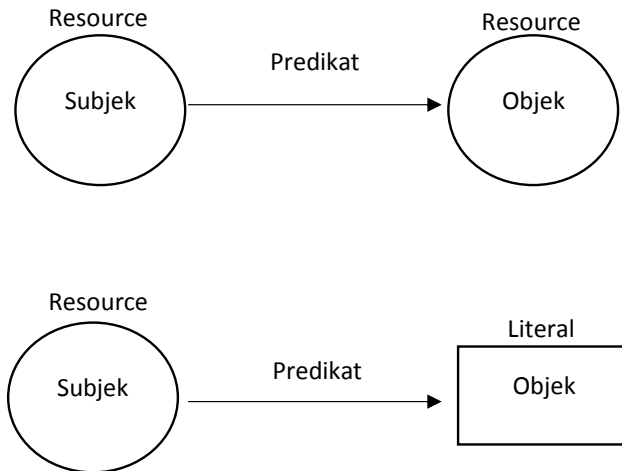
2.2 Dasar Teori

2.2.1 RDF

RDF merupakan singkatan dari Resource Description Framework. RDF merupakan suatu model standar yang digunakan untuk pertukaran data pada web [2]. RDF pertama kali diperkenalkan pada awal tahun 1999 oleh W3C sebagai standar pengkodean metadata. Sebelum RDF diciptakan, Web dibuat dalam bahasa yang hanya dimengerti oleh manusia dan tidak dapat dimengerti oleh mesin sama sekali. Hal ini menyebabkan otomatisasi pada web sangat sulit untuk dilakukan. Sehingga penggunaan metadata diusulkan untuk pertukaran data pada web, dimana metadata dapat dimengerti oleh mesin dan memungkinkan proses otomatisasi pada web [13]. RDF menampilkan informasi ke dalam bentuk statement yang terdiri atas subjek-predikat-objek, dimana subjek dan objek dapat berupa nama suatu hal, benda, orang (resource), sedangkan predikat (property) merupakan penghubung yang menghubungkan subjek dan objek. Suatu statement RDF selalu terdiri atas subjek, predikat, dan objek, sehingga disebut dengan triple. Misalnya, :john :mengendarai :motor merupakan contoh dari statement RDF. Ilustrasi RDF dapat dilihat pada gambar 2.1.

Komponen-komponen pada subjek, predikat, dan objek dapat berupa URI, Literal, maupun Blank Node.

- URI (Uniform Resource Identifier) dapat ditempatkan pada subjek, predikat, maupun objek. URI merupakan kumpulan karakter yang menjadi identifikasi sebuah nama. URI dapat berupa URL (Uniform Resource Locator) ataupun URN (Uniform Resource Name). Contoh dari URI: <http://dbpedia.org/resource/Jakarta>.
- Literal dapat digunakan pada posisi objek untuk menunjukk-



Gambar 2.1: Hubungan ”subjek-predikat-objek” [7]

an tanggal, angka, atau huruf. Pada RDF graph, literal digambarkan dengan kotak segi empat, seperti pada gambar 2.1. Ada dua jenis Literal, yaitu Plain Literal dan Typed Literal. Plain Literal yang terdiri atas karakter dan bisa diberi kode bahasa, misalnya `mathematic@en`. Sedangkan Typed Literal menerangkan tipe data dari karakter, misalnya `xsd:string`.

- Blank Node dapat digunakan pada subjek dan objek untuk menunjukkan node yang tidak memiliki nama, tidak dapat di-representasikan dalam URI maupun Literal. Penulisan Blank Node diawali dengan underscore.

Ada beberapa format yang digunakan untuk menuliskan RDF triple. Beberapa diantaranya yang paling terkenal adalah Notation-3, Terse RDF Triple Language (Turtle), N-triples, dan RDF/XML.

Agar suatu website dapat dibaca oleh mesin, maka harus terdapat versi RDF dari website tersebut. Ada beberapa spesifikasi yang dapat diikuti, seperti RDFa dan Microformat untuk menambahkan data RDF ke dalam dokumen web. Data RDF dapat ditempatkan pada bagian metadata dari halaman HTML, atau dapat ditambahkan sebagai atribut pada HTML tag [7].

2.2.2 Linked Data

Linked Data merupakan istilah yang digunakan untuk publikasi data pada web dengan cara tertentu sehingga membuat data tersebut dapat dimengerti oleh mesin. Data tersebut saling terhubung dengan dataset eksternal lain, begitu pula sebaliknya. Linked Data juga dapat diartikan sebagai praktik terbaik dalam mempublikasikan dan menghubungkan data terstruktur pada web.

Konsep dari Linked Data secara garis besar ialah sebagai berikut:

1. Menggunakan data model RDF untuk mempublikasikan data terstruktur pada web.
2. Menggunakan *link* RDF untuk menghubungkan data-data dari berbagai sumber

Penerapan kedua hal tersebut membuat terciptanya Web of Data yang dapat dimengerti oleh mesin. Web of Data ini dapat dikatakan sebagai salah satu bentuk realisasi dari Semantic Web. Saat ini, Linked Data dan Semantic Web merupakan dua konsep yang dapat saling dipertukarkan [13].

2.2.3 SPARQL Query

SPARQL merupakan akronim rekursif dari SPARQL Protocol and RDF Query Language. SPARQL adalah bahasa query yang digunakan untuk mendapatkan dan memanipulasi data yang tersimpan dalam format RDF. SPARQL mirip seperti SQL, dimana SQL digunakan untuk memperoleh data dari database. Sebagaimana SQL, SPARQL juga memungkinkan hasil query untuk dilakukan sorting, filter, dan lain-lain [4]. SPARQL mendukung triple pattern, conjunction, disjunction, dan optional pattern [7]. SPARQL memungkinkan untuk dijalankan pada semua format RDF, seperti RDF/XML, Turtle, N-triples, dan lain-lain [8]. Versi terbaru dari SPARQL saat ini adalah versi 1.1 yang dikeluarkan pada Maret 2013.

Pada umumnya, SPARQL memiliki struktur sebagai berikut:

Kode 2.1: Struktur SPARQL

```
PREFIX ...
FROM ...
SELECT/ASK/CONSTRUCT/DESCRIBE ...
?variabel konstanta
ORDER BY ...
```

PREFIX digunakan untuk mendeklarasikan awalan untuk memper singkat penulisan alamat sehingga mempermudah penulisan query. FROM digunakan untuk menentukan sumber data. SELECT/ASK/CONSTRUCT/DESCRIBE merupakan perintah yang dapat dijalankan pada SPARQL. Sedangkan ORDER BY digunakan untuk menentukan urutan hasil yang ditampilkan, apakah ascending (dari kecil ke besar) atau descending. Perlu diperhatikan bahwa di akhir dari setiap baris pada query perlu ditambahkan tanda titik.

SPARQL memiliki beberapa perintah yang dapat dijalankan, diantaranya SELECT, ASK, CONSTRUCT, dan DESCRIBE. :

Tabel 2.1: Hasil query SELECT setelah dijalankan

Negara	Ibukota
http://dbpedia.org/resource/Brunei	http://dbpedia.org/resource/Bandar_Seri_Begawan
http://dbpedia.org/resource/Cambodia	http://dbpedia.org/resource/Phnom_Penh
http://dbpedia.org/resource/East_Timor	http://dbpedia.org/resource/Dili
http://dbpedia.org/resource/Indonesia	http://dbpedia.org/resource/Jakarta
http://dbpedia.org/resource/Laos	http://dbpedia.org/resource/Vientiane
http://dbpedia.org/resource/Malaysia	http://dbpedia.org/resource/Kuala_Lumpur
http://dbpedia.org/resource/Malaysia	http://dbpedia.org/resource/Putrajaya
http://dbpedia.org/resource/Philippines	http://dbpedia.org/resource/Manila
http://dbpedia.org/resource/Singapore	http://dbpedia.org/resource/City-state
http://dbpedia.org/resource/Thailand	http://dbpedia.org/resource/Bangkok
http://dbpedia.org/resource/Vietnam	http://dbpedia.org/resource/Hanoi
http://dbpedia.org/resource/Myanmar	http://dbpedia.org/resource/Naypyidaw

- SELECT digunakan untuk mengambil data dari SPARQL endpoint. Berikut ini contoh query yang dijalankan pada DBpedia SPARQL endpoint untuk mencari negara-negara di Asia Tenggara beserta ibu kotanya:

Kode 2.2: Query SELECT

```

PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dct:<http://purl.org/dc/terms/>
SELECT ?Negara ?Ibukota {
  ?Negara dct:subject <http://dbpedia.org/
    resource/Category:
      Southeast_Asian_countries >.
  ?Negara dbo:capital ?Ibukota.
}

```

Hasil query setelah dijalankan dapat dilihat pada Tabel 2.1.

- ASK digunakan untuk memastikan ada tidaknya suatu data pada SPARQL endpoint. Hasil query bernilai true atau false. Berikut ini contoh query yang dijalankan pada DBpedia SPARQL endpoint untuk mengetahui ada tidaknya data mengenai Jakarta:

Kode 2.3: Query ASK

```
ASK {
<http://dbpedia.org/resource/Jakarta> ?p ?o
}
```

Setelah query tersebut dijalankan maka akan muncul tulisan "true" yang menunjukkan terdapat data yang berhubungan dengan Jakarta.

- CONSTRUCT memiliki fungsi yang hampir sama seperti SELECT, yaitu untuk mengambil data dari SPARQL endpoint. Yang membedakannya dengan SELECT yaitu hasil dari query yang ditampilkan dalam RDF graph (sekumpulan triple). Berikut contoh query yang dijalankan pada DBpedia SPARQL endpoint untuk menampilkan negara-negara di Asia Tenggara beserta ibu kotanya:

Kode 2.4: Query CONSTRUCT

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dct:<http://purl.org/dc/terms/>
CONSTRUCT {
?Negara dbo:capital ?Ibukota
}
WHERE {
?Negara dbo:capital ?Ibukota .
?Negara dct:subject <http://dbpedia.org/
resource/Category:
Southeast_Asian_countries >.
}
```

Berikut hasil query setelah dijalankan:

Kode 2.5: Hasil query CONSTRUCT setelah dijalankan

```
This HTML5 document contains 12 embedded RDF statements represented using
HTML+Microdata notation.

The embedded RDF content will be recognized by any processor of HTML5
Microdata.
Prefix Namespace IRI
dbo http://dbpedia.org/ontology/
rdf http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```

xsdh    http://www.w3.org/2001/XMLSchema#
dbr     http://dbpedia.org/resource/

```

```

Subject Item
  dbr:Brunei
dbo:capital
  dbr:Bandar_Seri_Begawan

```

```

Subject Item
  dbr:Cambodia
dbo:capital
  dbr:Phnom_Penh

```

```

Subject Item
  dbr:East_Timor
dbo:capital
  dbr:Dili

```

```

Subject Item
  dbr:Indonesia
dbo:capital
  dbr:Jakarta

```

```

Subject Item
  dbr:Laos
dbo:capital
  dbr:Vientiane

```

```

Subject Item
  dbr:Malaysia
dbo:capital
  dbr:Putrajaya dbr:Kuala_Lumpur

```

```

Subject Item
  dbr:Philippines
dbo:capital
  dbr:Manila

```

```

Subject Item
  dbr:Singapore
dbo:capital
  dbr:City-state

```

```

Subject Item
  dbr:Thailand
dbo:capital
  dbr:Bangkok

```

```

Subject Item
  dbr:Vietnam
dbo:capital
  dbr:Hanoi

```

```

Subject Item
  dbr:Myanmar
dbo:capital
  dbr:Naypyidaw

```

- DESCRIBE digunakan untuk menampilkan seluruh data yang berhubungan dengan data yang dicari. Berikut contoh query yang dijalankan pada DBpedia SPARQL endpoint untuk me-

nampilkan seluruh data yang berhubungan dengan Terminal Purabaya:

Kode 2.6: Query DESCRIBE

```
DESCRIBE <http://dbpedia.org/resource/Purabaya_Bus_Station>
```

Berikut hasil query setelah dijalankan:

Kode 2.7: Hasil query DESCRIBE setelah dijalankan

```
This HTML5 document contains 42 embedded RDF statements represented using
HTML+Microdata notation.

The embedded RDF content will be recognized by any processor of HTML5
Microdata.
Prefix Namespace IRI
dct http://purl.org/dc/terms/
yago-res http://yago-knowledge.org/resource/
dbo http://dbpedia.org/ontology/
foaf http://xmlns.com/foaf/0.1/
dbpedia-wikidata http://wikidata.dbpedia.org/resource/
n17 http://en.wikipedia.org/wiki/Purabaya_Bus_Station?oldid=
yago http://dbpedia.org/class/yago/
schema http://schema.org/
rdfs http://www.w3.org/2000/01/rdf-schema#
n19 http://rdf.freebase.com/ns/m.
rdf http://www.w3.org/1999/02/22-rdf-syntax-ns#
owl http://www.w3.org/2002/07/owl#
wikipedia-en http://en.wikipedia.org/wiki/
dbp http://dbpedia.org/property/
dbc http://dbpedia.org/resource/Category:
prov http://www.w3.org/ns/prov#
xsdh http://www.w3.org/2001/XMLSchema#
dbpedia-id http://id.dbpedia.org/resource/
wikidata http://www.wikidata.org/entity/
dbr http://dbpedia.org/resource/

Subject Item
  dbr:Purabaya_Bus_Station
rdf:type
  owl:Thing wikidata:Q41176 yago:PhysicalEntity100001930 schema:Place
  yago:Whole100003553 yago:YagoGeoEntity yago:
  YagoPermanentlyLocatedEntity dbo:Building yago:Building102913152
  dbo:Location yago:Artifact100021939 dbo:ArchitecturalStructure
  yago:Object100002684 yago:
  WikicatBuildingsAndStructuresInSurabaya yago:Structure104341686
  dbo:Place
rdfs:label
  Purabaya Bus Station
rdfs:comment
  Terminal Purabaya (English:Purabaya Bus Station), or more popularly
  known as Terminal Bungurasih is the busiest bus terminal in
  Indonesia (up to 120,000 passengers per day). This terminal is
  located in the outside of Surabaya, actually in Waru, Sidoarjo.
  This bus station serves local and inter-island route.
owl:sameAs
```

```

yago-res: Purabaya_Terminal dbpedia-id: Terminal_Purabaya dbpedia-
wikidata: Q7260794 wikidata: Q7260794 n19: 07sb4ds
prov: wasDerivedFrom
n17: 677654917
dbo: abstract
Terminal Purabaya (English: Purabaya Bus Station), or more popularly
known as Terminal Bungurasih is the busiest bus terminal in
Indonesia (up to 120,000 passengers per day). This terminal is
located in the outside of Surabaya, actually in Waru, Sidoarjo.
This bus station serves local and inter-island route.
dbo: address
Jl. Letjen Sutoyo, Bungurasih, Waru, Sidoarjo
dbo: location
dbr: Surabaya
dbo: type
dbr: Bus_station
foaf: name
Terminal Purabaya (Purabaya Bus Station)
dbp: name
Terminal Purabaya
foaf: isPrimaryTopicOf
wikipedia-en: Purabaya_Bus_Station
dct: subject
dbc: Transport_in_East_Java dbc: Bus_transport_in_Indonesia dbc:
Buildings_and_structures_in_Surabaya
dbo: wikiPageID
24388209
dbo: wikiPageRevisionID
677654917
dbp: address
Jl. Letjen Sutoyo, Bungurasih, Waru, Sidoarjo
dbp: buildingType
dbr: Bus_station
dbp: locationTown
dbr: Surabaya

Subject Item
dbr: Bungurasih
dbo: wikiPageRedirects
dbr: Purabaya_Bus_Station

Subject Item
dbr: Purabaya_Terminal
dbo: wikiPageRedirects
dbr: Purabaya_Bus_Station

Subject Item
wikipedia-en: Purabaya_Bus_Station
foaf: primaryTopic
dbr: Purabaya_Bus_Station

```

:

2.2.4 Dataset RDF

Dataset RDF merupakan bentuk publikasi dari sekumpulan data yang terdapat pada Web of Data yang terdiri atas triple dalam jumlah banyak [6]. Dataset RDF tersimpan dalam database RDF yang disebut dengan triple store.

Dataset RDF memiliki karakteristik yang menentukan struktur dan sifat dari dataset tersebut. Karakteristik tersebut termasuk jumlah dan jenis dari atribut atau variabel yang terdapat pada dataset beserta penghitungan statistik lainnya. Sebagai contoh, terdapat sebuah dataset sebagai berikut:

Kode 2.8: Contoh dataset

```
( person0 , name , Eric )  
( person0 , office , BA7430 )  
( person0 , ext , x4402 )  
( person1 , name , Kenny )  
( person1 , office , BA7349 )  
( person1 , office , BA7350 )  
( person1 , ext , x5304 )  
( person2 , name , Kyle )  
( person2 , ext , x6282 )  
( person3 , name , Timmy )  
( person3 , major , C . S . )  
( person3 , GPA , 3 . 4 )  
( person4 , name , Stan )  
( person4 , GPA , 3 . 8 )  
( person5 , name , Jimmy )  
( person5 , GPA , 3 . 7 )
```

Dari dataset di atas, dapat diambil beberapa karakteristik dataset sebagai berikut:

- Jumlah subjek

Terdapat 6 subjek pada dataset di atas, yaitu person1, person2, person3, person4, dan person5.

- Jumlah objek
Terdapat 16 objek pada dataset di atas, yaitu Eric, BA7430, x4402, Kenny, BA7349, BA7350, x5304, Kyle, x6282, Timmy, C.S., 3.4, Stan, 3.8, Jimmy, dan 3.7
- Jumlah property
Terdapat 5 property atau predikat pada dataset di atas, yaitu name, office, ext, major, dan GPA
- Jumlah triple
Jumlah triple menunjukkan jumlah data pada dataset yang diekspresikan dalam bentuk subjek-predikat-objek. Terdapat 16 triple pada dataset di atas.

2.2.5 Spreading Factor

Spreading factor adalah metrik yang digunakan untuk mengetahui persebaran kelas dan property pada suatu dataset [10]. Diketahui dataset D berisi sekumpulan dataset d , maka spreading factor dapat dihitung menggunakan rumus berikut

$$\Gamma(D) = \frac{(1 + \beta^2)OCP(D) \times OCC(D)}{\beta^2 \times OCP(D) + OCC(D)} \quad (2.1)$$

dimana $\beta = 0.5$.

$OCP(D)$ merupakan tingkat kemunculan property pada dataset D yang dapat dihitung dengan rumus berikut:

$$OCP(D) = \frac{\sum_{\forall d \in D} |P_d(d, D)|}{|P(D)| \times |D|} \quad (2.2)$$

Sedangkan $OCC(D)$ merupakan tingkat kemunculan kelas pada da-

taset D yang dapat dihitung dengan rumus berikut:

$$OCC(D) = \frac{\sum_{\forall d \in D} |C_d(d, D)|}{|C(D)| \times |D|} \quad (2.3)$$

Nilai dari spreading factor berkisar dari 0 sampai 1. Nilai yang semakin tinggi menunjukkan bahwa kelas dan property semakin menyebar dalam dataset.

Selain spreading factor yang sudah dijelaskan di atas, terdapat spreading factor yang dihitung berdasarkan query. Berbeda dengan spreading factor yang telah dijelaskan sebelumnya yang digunakan untuk menghitung persebaran seluruh kelas dan property, spreading factor berdasarkan query digunakan untuk menghitung persebaran kelas dan property pada dataset, dimana kelas dan property yang dihitung hanya kelas dan property yang terdapat pada query yang akan dieksekusi saja [10]. Diketahui terdapat suatu set query $Q = \{q1, q2, \dots, qn\}$, maka spreading factor γ dari dataset D yang berhubungan dengan query Q dapat dihitung menggunakan rumus berikut:

$$\gamma(Q, D) = \sum_{\forall q \in Q} \frac{\sum_{\forall \tau \in q} OC(\tau, D)}{|Q|} \quad (2.4)$$

τ merupakan baris (tuple) yang terdapat pada query, dimana mengandung kelas atau property. $OC(\tau, D)$ merupakan kemunculan kelas dan property dari tuple τ pada dataset, yang dapat dihitung sebagai berikut:

- Apabila property pada suatu tuple berupa `rdf:type` dan objek pada tuple tidak berupa variabel, maka $OC(\tau, D)$ dapat diketahui dengan rumus berikut:

$$OC(\tau, D) = \frac{ofD(o\tau, D)}{|D|} \quad (2.5)$$

dimana

$$ofd(o, D) = \sum_{\forall d \in D} ofd(o, d, D) \quad (2.6)$$

Apabila terdapat kemunculan objek o pada dataset d , maka $ofd(o, d, D)$ bernilai 1, jika tidak maka bernilai 0.

- Apabila property pada suatu tuple tidak berupa rdf:type dan property tidak berupa variabel, maka $OC(\tau, D)$ dapat diketahui dengan rumus berikut:

$$OC(\tau, D) = \frac{pfD(p\tau, D)}{|D|} \quad (2.7)$$

dimana

$$pfD(p, D) = \sum_{\forall d \in D} pfD(p, d, D) \quad (2.8)$$

Apabila terdapat kemunculan property o pada dataset d , maka $ofd(o, d, D)$ bernilai 1, jika tidak maka bernilai 0.

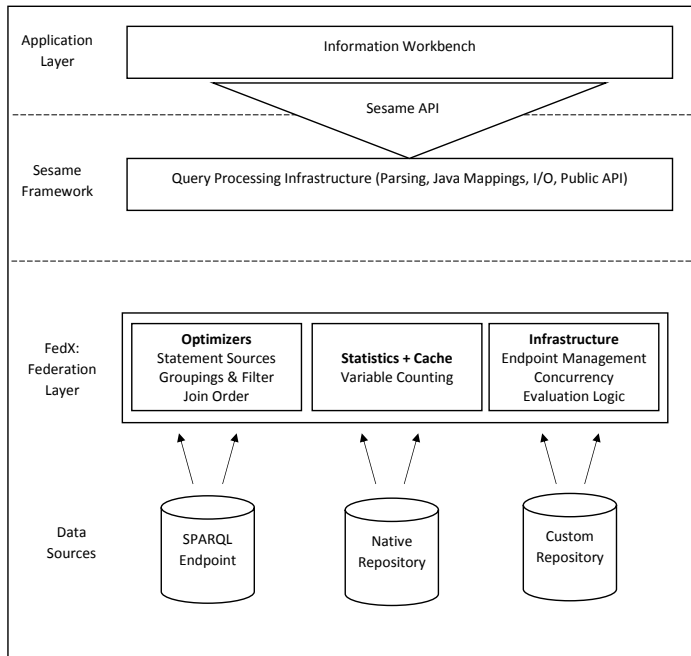
- Apabila dua poin di atas tidak terpenuhi, maka $OC(\tau, D)$ dapat diketahui dengan rumus berikut:

$$OC(\tau, D) = \frac{\sum_{\forall d \in D} |Pd(d, D)|}{|D|} \quad (2.9)$$

2.2.6 FedX

FedX merupakan federated SPARQL query engine yang dikembangkan untuk menyediakan solusi efisien untuk pemrosesan distributed query pada Linked Open Data. FedX diimplementasikan dalam bahasa Java dan merupakan ekstensi dari Sesame framework dengan federation layer. FedX mengimplementasikan Sesame framework sebagai SAIL (Storage and Inference Layer), dimana hal ini memungkinkan integrasi yang baik dengan repositori RDF standar maupun yang terkustomisasi. Adanya infrastruktur Sesame me-

mungkinkan sumber data yang heterogen untuk dijadikan endpoint dalam federation. Arsitektur dari FedX dapat dilihat pada gambar 2.2.



Gambar 2.2: Arsitektur FedX [12]

Layer pertama merupakan Application Layer yang menyediakan frontend untuk pemrosesan query. Bagian ini dibutuhkan untuk proses interaksi. Pada gambar, digunakan Information Workbench sebagai contoh aplikasi. Layer kedua merupakan Sesame Framework yang menyediakan infrastruktur untuk pemrosesan query,

seperti parsing, java mappings, komponen I/O, dan API untuk interaksi dengan klien. Layer ketiga adalah Federation Layer yang memiliki fungsi hampir sama seperti layer kedua, namun terdapat penambahan fungsionalitas seperti optimalisasi pemrosesan distributed query, manajemen sumber data, dan komunikasi endpoint. Pada layer ini dapat ditambahkan sumber data dalam berbagai bentuk yang didukung oleh Sesame, seperti native repository, SPARQL repository, dan repository yang dikustomisasi.

Langkah-langkah pemrosesan query pada FedX adalah sebagai berikut. Pertama adalah perumusan global query untuk federation pada sumber data. Kemudian global query akan diparse dan dioptimisasi, dimana dilakukan pembagian menjadi subquery lokal yang dapat dimengerti oleh masing-masing sumber data. Hasil dari masing-masing local query tersebut kemudian akan digabungkan dan dikembalikan dalam bentuk agregat [12].

BAB 3

METODOLOGI

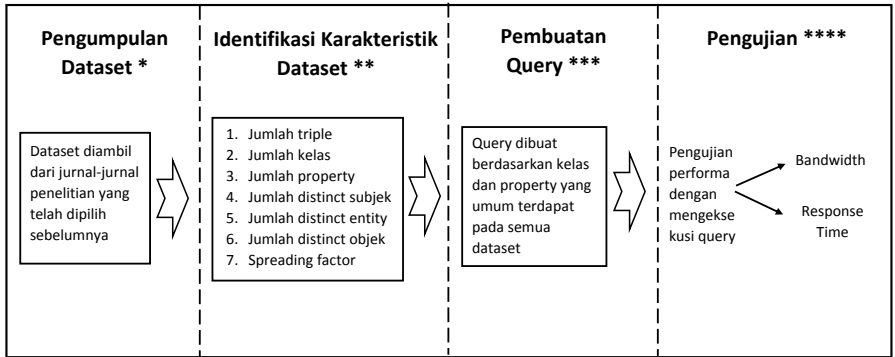
Pada bab ini akan dijelaskan bagaimana langkah pengerjaan tugas akhir dengan disertakan deskripsi dari setiap penjelasan untuk masing-masing tahapan beserta jadwal kegiatan pengerjaan tugas akhir.

3.1 Tahapan Pengerjaan Tugas Akhir

Sub bab ini akan menjelaskan mengenai proses pengerjaan tugas akhir. Tahapan pengerjaan tugas akhir dapat dilihat pada gambar 3.1

3.1.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan literatur yang mendukung dalam menyelesaikan tugas akhir ini. Literatur adalah penjelasan konsep-konsep atau penelitian sebelumnya yang pernah dilakukan dan didokumentasikan dalam buku, jurnal, maupun website. Pada pengerjaan tugas akhir ini, studi literatur dilakukan dengan mengumpulkan literatur yang berhubungan dengan topik tugas akhir, yaitu literatur tentang RDF, Linked Data, SPARQL, maupun federated query engine. Output atau keluaran proses ini adalah pemahaman mengenai konsep dan knowledge gap pada penelitian sebelumnya.



- * Dataset yang digunakan yaitu 10 dataset jurnal penelitian yang telah dipilih sebelumnya. Kriteria pemilihan dapat dilihat pada subbab 3.1.2
- ** Jenis karakteristik yang diidentifikasi mengacu pada referensi [10] dan [5].
- *** Daftar kelas dan property terdapat pada bagian lampiran.
- **** Pengujian performa dilakukan menggunakan federated SPARQL query engine FedX.

Gambar 3.1: Alur pengerjaan tugas akhir

3.1.2 Pengumpulan Data

Dataset yang digunakan pada penelitian ini diambil dari jurnal penelitian lain. Oleh karena itu, tahap pengumpulan data dilakukan dengan mengumpulkan jurnal untuk mendapatkan dataset yang dibutuhkan. Jurnal yang dikumpulkan adalah jurnal yang memiliki kriteria sebagai berikut:

- Jurnal berhubungan dengan Linked Data
- Jurnal memakai federated repository
- Dataset yang digunakan pada jurnal dapat diakses secara gratis

Jurnal diperoleh dari beberapa sumber, antara lain Google Scholar,

Science Direct, Semantic Web Journal, dan International Semantic Web Conference. Berikut ini 10 jurnal dan dataset yang berhasil dikumpulkan, dapat dilihat pada Tabel 3.1.

3.1.3 Identifikasi Karakteristik Dataset

Pada tahap ini dilakukan proses identifikasi karakteristik dataset yang terdapat pada masing-masing paper. Proses ini dilakukan dengan tujuan untuk mengetahui struktur dan sifat dari dataset. Proses identifikasi karakteristik dataset meliputi hal-hal sebagai berikut:

- Jumlah triple
- Jumlah kelas
- Jumlah property
- Jumlah distinct subjek
- Jumlah distinct entity
- Jumlah distinct object
- Spreading factor dataset

3.1.4 Pembuatan Query

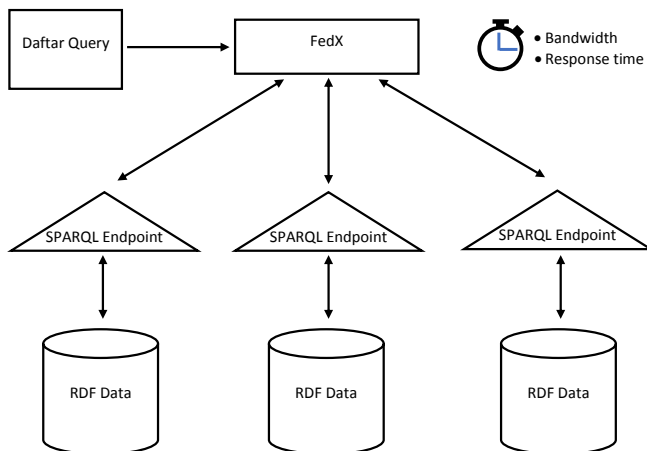
Pada tahap ini dilakukan proses *generate* query. Pembuatan query ini dilakukan karena query dibutuhkan pada saat proses pengujian kinerja dari query engine, dimana pengujian dilakukan dengan cara mengeksekusi query tersebut pada query engine untuk mengakses dataset yang terdapat pada masing-masing jurnal. Pembuatan query dibuat berdasarkan kelas dan predikat yang paling umum terdapat pada dataset-dataset tersebut. Hal tersebut dimaksudkan agar proses eksekusi query pada setiap dataset bisa memunculkan hasil, sehingga memungkinkan untuk dilakukan pengujian kinerja.

Tabel 3.1: Daftar Jurnal dan Dataset yang Dipakai dalam Penelitian

Nomor	Judul Jurnal	Dataset
1	Countering language attrition with PanLex and the Web of Data	<ul style="list-style-type: none"> ● Lexvo ● DBpedia
2	How Redundant Is It? - An Empirical Analysis on Linked Datasets	<ul style="list-style-type: none"> ● LinkedMDB ● Linked Open Vocabularies (LOV) ● DBLP
3	iRap - An interest-based RDF Update Propagation Framework	<ul style="list-style-type: none"> ● DBpedia ● LinkedGeoData
4	LED: curated and crowdsourced Linked Data on Music Listening Experiences	<ul style="list-style-type: none"> ● DBpedia ● British National Bibliography
5	Lexvo.org: Language Information for the Linguistic Linked Data Cloud	<ul style="list-style-type: none"> ● DBpedia ● YAGO ● WordNet
6	Luzzu - A Framework for Linked Data Quality Assessment	<ul style="list-style-type: none"> ● Democratic city RDF dataset ● OCD : Cidade Democrática Ontology
7	Property Path over Linked Data: Can It be Done and How to Start?	<ul style="list-style-type: none"> ● Yago ● DBpedia ● Wikidata
8	Semantic Hadith: Leveraging Linked Data Opportunities for Islamic Knowledge	<ul style="list-style-type: none"> ● QuranOntology ● SemanticQuran
9	Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	<ul style="list-style-type: none"> ● GeoNames ● DBpedia ● LinkedGeoData
10	The Semantic Web Journal as Linked Data	<ul style="list-style-type: none"> ● DBLP ● Citeseer

3.1.5 Pengujian

Pada tahap uji coba pengujian dataset, dilakukan eksekusi query yang telah dibuat pada tahap sebelumnya menggunakan federated SPARQL query engine FedX. Pada saat eksekusi query dijalankan, dilakukan analisis untuk mengetahui karakteristik dataset yang berpengaruh pada kinerja dari FedX. Analisis yang dilakukan meliputi response time dan bandwidth pada masing-masing dataset. Ilustrasi pengujian dataset dapat dilihat pada gambar 3.2



Gambar 3.2: Ilustrasi pengujian dataset

Halaman ini sengaja dikosongkan

BAB 4

PERANCANGAN

Pada bab ini akan dibahas mengenai perancangan dari pengerjaan tugas akhir ini yang meliputi identifikasi karakteristik dataset dan pembuatan query.

4.1 Pengumpulan Dataset

Pengumpulan dataset merupakan proses paling awal yang dilakukan dalam pengerjaan tugas akhir ini. Pada penelitian ini, dataset yang dibutuhkan diperoleh dari jurnal penelitian lain yang menggunakan federated repository. Oleh karena itu, langkah pertama yang dilakukan pada tahap ini ialah mengumpulkan 8 jurnal untuk memperoleh dataset RDF yang diperlukan. Jurnal yang terpilih ialah jurnal yang memenuhi kriteria-kriteria berikut, yaitu berhubungan dengan linked data, menggunakan federated repository, dan dataset yang digunakan dapat diperoleh secara gratis. Daftar jurnal yang telah berhasil dikumpulkan telah dijelaskan pada tabel 3.1. Setelah jurnal yang memenuhi kriteria terkumpul, selanjutnya dilakukan proses download dataset RDF yang terdapat pada jurnal-jurnal tersebut. Proses download dilakukan melalui website yang menyediakan dataset tersebut secara gratis. Berikut ini daftar dataset RDF yang telah dikumpulkan beserta sumber dan SPARQL endpoint, dapat dilihat pada tabel 4.1. Dataset diakses pada tanggal 16-23 Maret 2017.

Tabel 4.1: Daftar dataset beserta sumber dan SPARQL endpoint

Dataset	Sumber	SPARQL Endpoint
Quran Ontology	http://www.quranontology.com/	http://www.quranontology.com/Query
Semantic Quran	https://datahub.io/dataset/semanticquran	http://semanticquran.aksw.org/sparql
DBpedia	http://wiki.dbpedia.org/downloads-2016-04	https://dbpedia.org/sparql
LinkedGeoData	http://downloads.linkedgeodata.org/releases/2015-11-02/	http://linkedgeodata.org/sparql
Lexvo	http://www.lexvo.org/linkeddata/resources.html	-
Yago	https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/	https://linkeddata1.calcul.u-psud.fr/sparql
WordNet	http://wordnet-rdf.princeton.edu/	http://wordnet-rdf.princeton.edu/sparql/
GeoNames	http://www.geonames.org/ontology/documentation.html	-
British National Bibliography	http://www.bl.uk/bibliographic/download.html#lodbnb	http://bnb.data.bl.uk/flint-sparql
Wikidata	https://dumps.wikimedia.org/wikidatawiki/entities/	https://query.wikidata.org/
DBLP	https://datahub.io/dataset/l3s-dblp	http://dblp.l3s.de/d2r/snorql/
Citeseer	https://datahub.io/dataset/rkb-explorer-citeseer	http://citeseer.rkbexplorer.com/sparql/
Democratic City RDF Dataset	https://datahub.io/dataset/democratic-city/resource/25bc850b-bcfb-4203-a834-f1c744a1eda7?inner_span=True	-
OCD : Cidade Democratica Ontology	http://vocab.e.gov.br/2014/10/OCD.ttl	-
LinkedMDB	http://www.cs.toronto.edu/~sim\$oktie/linkedmdb/	http://www.linkedmdb.org/snorql/
Linked Open Vocabularies (LOV)	http://lov.okfn.org/dataset/lov/sparql	http://lov.okfn.org/dataset/lov/sparql

4.2 Praproses Dataset

Sebelum dataset yang telah berhasil didownload diidentifikasi, perlu dilakukan praproses data pada beberapa dataset tertentu. Hal ini dikarenakan tidak semua dataset yang baru saja didownload dapat dibaca dengan baik oleh aplikasi pemroses data RDF. Pada umumnya hal tersebut disebabkan oleh adanya karakter yang tidak sesuai (*bad character*) pada beberapa dataset, sehingga error akan muncul ketika dataset tersebut diload pada aplikasi pemroses data RDF. Untuk menyelesaikan masalah tersebut, dilakukan perubahan *bad character* pada dataset menjadi karakter yang sesuai agar dataset dapat diload dengan baik pada aplikasi pemroses data RDF. Perubahan karakter pada dataset dilakukan menggunakan aplikasi editor teks EmEditor, dimana aplikasi ini mampu membaca file yang berukuran hingga beberapa GB.

Selain itu, terdapat pula permasalahan pada dataset GeoNames dimana format dataset tidak dikenali oleh aplikasi pemroses data RDF. Penyebabnya adalah format penulisan RDF yang terdapat pada dataset merupakan format yang tidak formal, sehingga aplikasi pemroses data RDF tidak dapat membaca dataset tersebut. Untuk menyelesaikannya, dilakukan konversi data menjadi format N-triples yang dapat dibaca oleh aplikasi pemroses data RDF. Proses konversi dilakukan menggunakan *script* pada Python [1] yang menggunakan *library* RDFLib.

4.3 Identifikasi Karakteristik Dataset

Setelah dataset yang dibutuhkan berhasil didownload, dataset-dataset tersebut kemudian akan diidentifikasi karakteristiknya, meliputi jumlah triple, jumlah entity, jumlah kelas, jumlah distinct subjek, jum-

lah distinct objek, jumlah property, daftar kelas, dan daftar property. Agar proses identifikasi tersebut dapat dilakukan, dibutuhkan software yang mampu menganalisis dan melakukan query pada data RDF. Salah satu software yang dapat melakukan hal tersebut ialah Stardog. Stardog dipilih karena software tersebut cukup mudah dalam penggunaannya, memiliki fitur yang cukup lengkap, serta memiliki kecepatan pemrosesan yang relatif lebih cepat dibanding software sejenis seperti Apache Jena.

Proses identifikasi dilakukan dengan cara menjalankan query-query [3] yang berfungsi untuk memperoleh karakteristik dataset yang diinginkan. Berikut ini query-query yang digunakan untuk mengidentifikasi karakteristik dataset:

Kode 4.1: Jumlah triples

```
SELECT (COUNT(*) AS ?no) { ?s ?p ?o }
```

Kode 4.2: Jumlah entity

```
SELECT (COUNT(distinct ?s) AS ?no) { ?s a [] }
```

Kode 4.3: Jumlah kelas

```
SELECT (COUNT(distinct ?o) AS ?no) { ?s rdf:type  
?o }
```

Kode 4.4: Jumlah property

```
SELECT (COUNT(distinct ?p) AS ?no) { ?s ?p ?o }
```

Kode 4.5: Jumlah distinct subjek

```
SELECT (COUNT(DISTINCT ?s ) AS ?no) { ?s ?p ?o  
}
```

Kode 4.6: Jumlah distinct objek

```
SELECT (COUNT(DISTINCT ?o ) AS ?no) { ?s ?p ?o
      filter (! isLiteral(?o)) }
```

Kode 4.7: Daftar kelas

```
SELECT DISTINCT ?type { ?s a ?type }
```

Kode 4.8: Daftar property

```
SELECT DISTINCT ?p { ?s ?p ?o }
```

Kode 4.9: Daftar subjek

```
SELECT DISTINCT ?s { ?s ?p ?o }
```

Kode 4.10: Daftar entity

```
SELECT distinct ?s { ?s a [] }
```

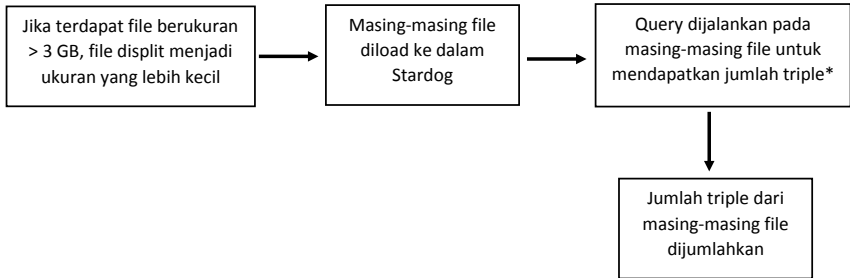
Kode 4.11: Daftar objek

```
SELECT DISTINCT ?o { ?s ?p ?o filter (!
      isLiteral(?o)) }
```

Dataset yang akan diidentifikasi memiliki jumlah dan ukuran yang bervariasi. Suatu dataset dapat terdiri dari satu hingga puluhan file RDF. Sedangkan ukuran file RDF juga bervariasi dari beberapa kilobyte hingga mencapai puluhan gigabyte. Dataset-dataset tersebut diidentifikasi menggunakan komputer dengan spesifikasi processor Intel Core i5 3.1 GHz, RAM 8 GB, Harddisk 3 TB, serta sistem operasi Windows 10. Dengan keterbatasan spesifikasi komputer yang dimiliki serta untuk memudahkan proses identifikasi, maka proses identifikasi dataset dilakukan dengan langkah-langkah berikut:

1. Jika terdapat lebih dari satu file dalam satu dataset:

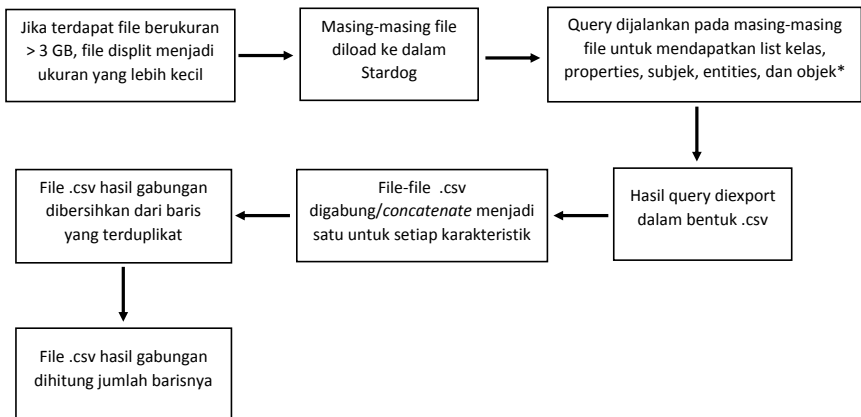
(a) Identifikasi jumlah triple:



* Query yang dimaksud adalah query pada kode A.1

Gambar 4.1: Alur identifikasi 1

(b) Identifikasi jumlah kelas, property, subjek, entity, dan objek:

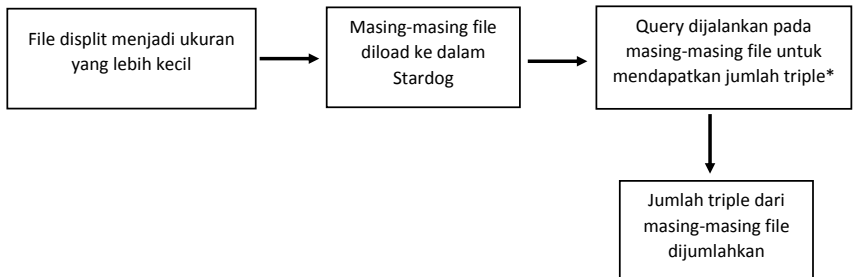


* Query yang dimaksud adalah query pada kode 4.7, kode 4.8, kode 4.9, kode 4.10, dan kode 4.11

Gambar 4.2: Alur identifikasi 2

2. Jika hanya terdapat satu file dalam satu dataset dengan ukuran lebih dari 3 GB:

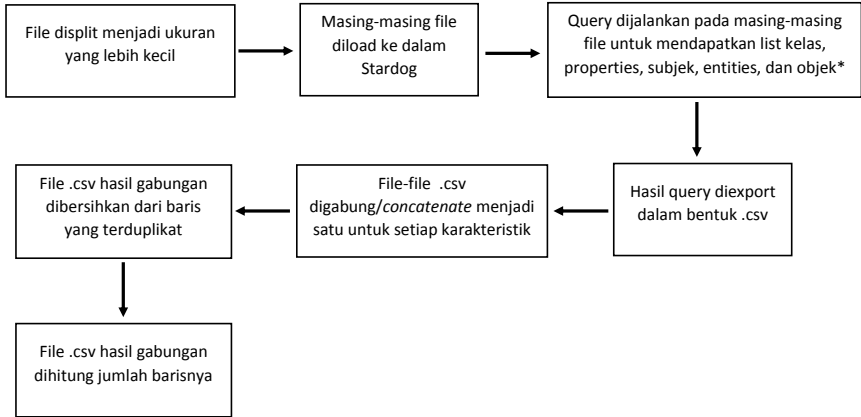
(a) Identifikasi jumlah triple:



* Query yang dimaksud adalah query pada kode A.1

Gambar 4.3: Alur identifikasi 3

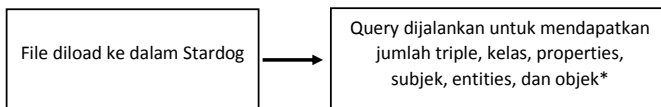
(b) Identifikasi jumlah kelas, property, subjek, entity, dan objek:



* Query yang dimaksud adalah query pada kode 4.7, kode 4.8, kode 4.9, kode 4.10, dan kode 4.11

Gambar 4.4: Alur identifikasi 4

3. Jika hanya terdapat satu file dalam satu dataset dengan ukuran kurang dari 3 GB, identifikasi jumlah triple, kelas, property, subjek, entity, dan objek dilakukan dengan:



* Query yang dimaksud adalah query pada kode A.1, kode 4.3, kode 4.4, kode 4.5, kode 4.2, dan kode 4.6

Gambar 4.5: Alur identifikasi 5

Pada pengerjaan langkah-langkah di atas, proses *splitting* file RDF dilakukan menggunakan software RDFPro. Sedangkan proses untuk menggabungkan file-file .csv dilakukan menggunakan command "copy" yang dijalankan melalui Command Prompt. Sementara itu,

Tabel 4.2: Hasil Identifikasi Karakteristik Dataset

Dataset	Triple	Kelas	Property	Subjek	Entity	Objek
British National Bibliography	144.754.861	29	63	19.100.626	12.682.101	21.284.676
Citeseer	6.282.235	4	19	859.175	859.175	859.468
DBLP	116.815.316	14	27	5.593.196	5.593.196	27.045.002
DBpedia	845.852.363	2.424	63.100	50.483.918	22.669.052	105.020.130
Democratic City	1.379.344	21	57	216.147	216.127	59.875
GeoNames	169.462.379	1	26	11.341.387	11.341.387	35.267.801
Lexvo	726.674	5	19	128.945	107.517	240.334
Linked Open Vocabularies	65.829	9	45	10.998	8.227	9.566
LinkedGeoData	1.292.933.812	1.136	32.491	284.098.269	267.737.503	268.402.435
LinkedMDB	6.147.996	53	222	694.400	665.441	1.049.261
OCD : Cidade Democratica Ontology	822	5	12	220	177	220
Quran Ontology	1.290.773	45	82	111.004	110.946	32.454
Semantic Quran	15.741.614	10	65	1.463.769	1.446.873	497.586
Wikidata	2.269.619.296	443	14.596	262.057.530	259.212.513	182.702.002
WordNet	5.557.709	5	64	647.215	645.761	1.193.996
Yago	1.090.372.899	484.085	88.736	331.212.386	15.368.508	17.412.052

proses pembersihan file .csv dari baris yang terduplikat dilakukan menggunakan command Unix "sort|uniq -i". Software Gnu-Win32 CoreUtils dibutuhkan agar command tersebut dapat berjalan di Windows.

Hasil identifikasi karakteristik masing-masing dataset dapat dilihat pada tabel 4.2:

4.3.1 Spreading Factor

Spreading Factor digunakan untuk mengetahui tingkat persebaran kelas dan property pada dataset yang digunakan oleh suatu jurnal. Nilai dari Spreading Factor berkisar dari 0 sampai 1. Nilai yang semakin tinggi menunjukkan bahwa kelas dan property semakin tersebar dalam dataset. Hasil penghitungan Spreading Factor pa-

da masing-masing jurnal dapat dilihat pada tabel 4.3:

4.4 Pembuatan Query

Pada tahap ini dilakukan pembuatan query, dimana query ini akan digunakan pada tahap pengujian. Pembuatan query dilakukan dengan mengacu pada kelas dan property yang paling banyak muncul di semua dataset. Gambar 4.6 menjelaskan mengenai langkah-langkah yang dilakukan untuk mencari kelas dan property terbanyak.

Kode 4.12: Daftar kelas

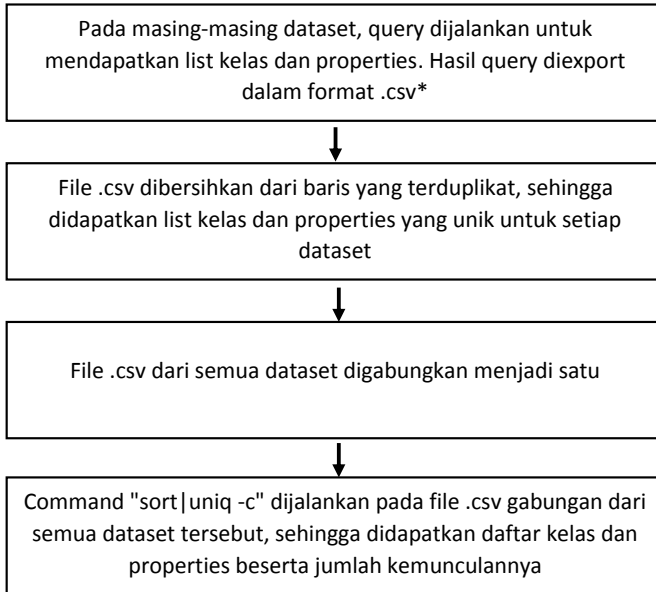
```
SELECT DISTINCT ?type { ?s a ?type }
```

Kode 4.13: Daftar property

```
SELECT DISTINCT ?p { ?s ?p ?o }
```

Table 4.3: Hasil Identifikasi Spreading Factor

Nomor	Judul Jurnal	Spreading Factor
1	Countering language attrition with PanLex and the Web of Data	0,50002535
2	How Redundant Is It? - An Empirical Analysis on Linked Datasets	0,346470172
3	iRap - An interest-based RDF Update Propagation Framework	0,500025108
4	LED: curated and crowdsourced Linked Data on Music Listening Experiences	0,500226496
5	Lexvo.org: Language Information for the Linguistic Linked Data Cloud	0,333347652
6	Luzzu - A Framework for Linked Data Quality Assessment	0,505865103
7	Property Path over Linked Data: Can It be Done and How to Start?	0,333349631
8	Semantic Hadith: Leveraging Linked Data Opportunities for Islamic Knowledge	0,527769074
9	Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	0,333364014
10	The Semantic Web Journal as Linked Data	0,537383178

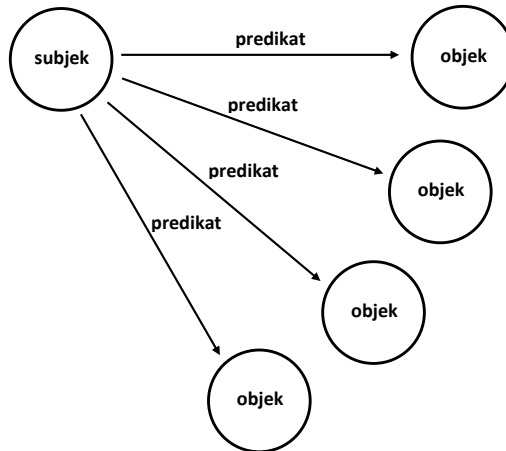


* Query yang dimaksud adalah query pada kode 4.12 dan kode 4.13.

Gambar 4.6: Alur mencari kelas dan property terbanyak

Daftar kelas dan property yang paling banyak muncul di semua dataset dapat dilihat pada tabel B.1 dan tabel B.2 pada bagian lampiran.

Pada proses pembuatan query ini, terdapat dua jenis query yang akan dibuat, yaitu star query dan chain query. Star query digunakan untuk mendapatkan satu subjek yang memiliki beberapa predikat dan objek. Sedangkan chain query digunakan untuk menampilkan beberapa subjek dan objek, dimana suatu objek dapat berperan menjadi subjek untuk memperoleh objek lain. Ilustrasi dari dua jenis query tersebut dapat dilihat pada gambar 4.7 dan gambar 4.8.



Gambar 4.7: Star Query

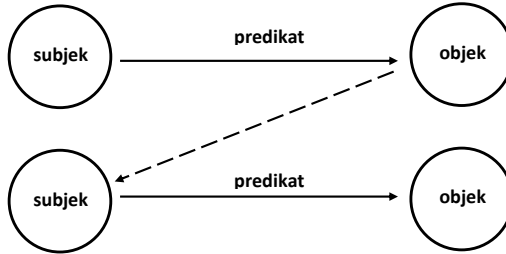
4.4.1 Hasil Pembuatan Query

Dengan mengacu pada kelas dan property terbanyak pada tabel B.1 dan tabel B.2 pada bagian lampiran, maka dapat dibuat star query dan chain query seperti yang terlihat pada kode 4.14, kode 4.15, dan kode 4.16. Pada query juga diberi batasan (*limit*) hasil yang ditampilkan untuk mengurangi kemungkinan waktu pemrosesan query yang terlalu lama.

Kode 4.14: Star Query 1

```

select * {
?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#
  type> <http://xmlns.com/foaf/0.1/Person>.
?s <http://www.w3.org/2000/01/rdf-schema#label>
  ?label.
?s <http://www.w3.org/2002/07/owl#sameAs> ?
  sameas.
?s <http://www.w3.org/2000/01/rdf-schema#comment
  
```



Gambar 4.8: Chain Query

```

    > ?comment .
}
LIMIT 5

```

Kode 4.15: Star Query 2

```

select * {
?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#
  type> <http://xmlns.com/foaf/0.1/Organization
  >.
?s <http://xmlns.com/foaf/0.1/name> ?name .
?s <http://xmlns.com/foaf/0.1/homepage> ?
  homepage .
}
LIMIT 5

```

Kode 4.16: Chain Query

```

select * {
?s <http://www.w3.org/2002/07/owl#sameAs> ?
  sameas .
?sameas <http://www.w3.org/2000/01/rdf-schema#
  seeAlso> ?seealso .
}
LIMIT 5

```


Query pertama (star query 1) terdiri dari empat baris, dimana terdapat satu kelas dan empat property yang digunakan. Query kedua (star query 2) terdiri dari tiga baris, dimana terdapat satu kelas serta tiga property yang digunakan. Sedangkan query ketiga (chain query) terdiri dari dua baris, dimana terdapat dua property yang digunakan. Tidak terdapat kelas pada query ketiga, karena query bertipe chain, dimana objek harus selalu berupa variabel.

Pembuatan query tersebut dilakukan dengan mengusahakan bahwa kelas dan property yang digunakan pada query ditemukan pada setidaknya satu dataset dari masing-masing jurnal. Sehingga ketika query tersebut dijalankan, maka akan memunculkan hasil sehingga response time dan bandwidth dapat diketahui. Namun terdapat dua jurnal yang datasetnya hanya memiliki kelas dan property yang spesifik dan kebanyakan tidak ditemukan pada jurnal lain, sehingga pembuatan query sulit untuk menjangkau kedua jurnal tersebut. Kedua jurnal tersebut ialah jurnal dengan judul "Luzzu - A Framework for Linked Data Quality Assessment" dan jurnal dengan judul "Semantic Hadith Leveraging Linked Data Opportunities for Islamic Knowledge". Sehingga diputuskan bahwa kedua jurnal tersebut tidak diikuti pada tahap pengujian.

4.4.2 Jumlah Kelas dan Property yang Terlibat dalam Query

Setelah query selesai dibuat, maka selanjutnya dilakukan identifikasi karakteristik dataset yang terkait dengan query. Identifikasi jumlah kelas dan property pada bagian ini berbeda dengan identifikasi jumlah kelas dan property yang telah dilakukan pada subbab 4.3, dimana pada bagian ini, penghitungan jumlah kelas dan property hanya dilakukan terhadap kelas dan property yang terlibat dalam query saja, serta jumlah kelas dan property yang diidentifikasi merupakan jumlah non-distinct. Jumlah kelas dan property tiap jurnal

yang terlibat dalam query akan digunakan dalam proses analisis hasil pengujian, dimana akan dibandingkan dengan response time dan bandwidth hasil pengujian untuk mengetahui apakah jumlah kelas dan property yang terlibat dalam query berpengaruh terhadap response time dan bandwidth atau tidak.

Proses identifikasi jumlah kelas dan property yang terlibat dalam query dilakukan dengan menjalankan query tertentu pada masing-masing dataset tiap jurnal. Berikut ini salah satu query SPARQL yang digunakan untuk mengetahui jumlah kelas foaf:person dan property rdf:type:

Kode 4.17: Query untuk Mencari Jumlah Kelas foaf:person

```
SELECT (COUNT(?s) AS ?count ) { ?s a <http://
  xmlns.com/foaf/0.1/Person> }
```

Kode 4.18: Query untuk Mencari Jumlah Property rdf:type

```
SELECT (COUNT(?s) AS ?count ) { ?s <http://www.
  w3.org/1999/02/22-rdf-syntax-ns#type> ?o }
```

Penghitungan dilakukan juga untuk kelas dan property lain yang terdapat pada query pada kode 4.14, kode 4.15, dan kode 4.16.

4.4.3 Spreading Factor Berdasarkan Query

Selain penghitungan spreading factor yang telah dilakukan pada subbab 4.3.1, dilakukan juga penghitungan spreading factor yang terkait dengan query. Spreading factor berdasarkan query nantinya akan digunakan pada tahap analisis untuk dibandingkan dengan response time dan bandwidth hasil pengujian. Hasil penghitungan spreading factor berdasarkan query dapat dilihat pada tabel 4.4. Jumlah jurnal yang diidentifikasi berjumlah 8, karena terdapat 2 jurnal

tidak diikutkan dalam tahap pengujian.

Sementara itu, hasil identifikasi karakteristik dataset tiap jurnal dalam bentuk rata-rata beserta spreading factor dan spreading factor berdasarkan query dapat dilihat pada tabel 4.5.

Tabel 4.4: Hasil Spreading Factor Berdasarkan Query

Nomor	Judul Jurnal	Spreading Factor Berdasarkan Query
1	Countering Language Attrition with PanLex and the Web of Data	2,166667
2	How Redundant Is It - An Empirical Analysis on Linked Datasets	2
3	iRap - An interest-based RDF Update Propagation Framework	1,833333
4	LED curated and crowdsourced Linked Data on Music Listening Experiences	2,5
5	Lexvo.org Language Information for the Linguistic Linked Data Cloud	1,777778
6	Property Path over Linked Data Can It be Done and How to Start	1,777778
7	Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	1,333333
8	The Semantic Web Journal as Linked Data	1,166667

Tabel 4.5: Rata-Rata dari Karakteristik Dataset Tiap Jurnal

Nomor	Judul Jurnal	Rata-Rata							SF	SFQ
		Triple	Kelas	Property	Subjek	Entity	Objek			
1	Countering language attrition with PanLex and the Web of Data	423.289.518,5	1.214,5	31.559,5	25.306.431,5	11.388.284,5	52.630.232	0,50002535	2,1666666667	
2	How Redundant Is It? - An Empirical Analysis on Linked Datasets	41.009.713,67	25,33333333	98	2.099.531,333	2.088.954,667	9.367,943	0,346470172	2	
3	iRap - An interest-based RDF Update Propagation Framework	1.069.393,088	1.780	47.795,5	167.291,093,5	145.203,277,5	186.711,282,5	0,500025108	1,833333333333	
4	LED: curated and crowdsourced Linked Data on Music Listening Experiences	495.303,612	1.226,5	31.581,5	34.792,272	17.675,576,5	63.152,403	0,500226496	2,5	
5	Lexvo.org: Language Information for the Linguistic Linked Data Cloud	647.260,990,3	162.171,3333	50.633,33333	127.447,839,7	12.894,440,33	41.208,726	0,333347652	1,7777777778	
6	Luzzu - A Framework for Linked Data Quality Assessment	690,083	13	34,5	108,183,5	108,152	30,047,5	0,505865103	-	
7	Property Path over Linked Data: Can It be Done and How to Start?	1.401.948,186	162.317,3333	55.477,33333	214.584,611,3	99.083,357,67	101.711,394,7	0,333349631	1,7777777778	
8	Semantic Hadith: Leveraging Linked Data Opportunities for Islamic Knowledge	8.516,193,5	27,5	73,5	787,386,5	778,909,5	265,020	0,527769074	-	
9	Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	769.416,184,7	1.187	31.872,33333	115.307,858	100.582,647,3	136.230,122	0,333364014	1,333333333333	
10	The Semantic Web Journal as Linked Data	61.548,775,5	9	23	3.226,185,5	3.226,185,5	13.952,235	0,537383178	1,1666666667	

SF : Spreading factor

SFQ : Spreading factor berdasarkan query

Halaman ini sengaja dikosongkan

BAB 5

IMPLEMENTASI

Pada bab ini akan dijelaskan terkait proses-proses pada pengujian dataset.

5.1 Lingkungan Implementasi

Pada bagian ini akan dibahas mengenai lingkungan pengujian yang digunakan dalam implemetasi tugas akhir, meliputi perangkat yang digunakan baik itu perangkat keras maupun perangkat lunak. Tabel 5.1 berisi spesifikasi perangkat keras yang digunakan pada proses implementasi tugas akhir.

Tabel 5.1: Spesifikasi Perangkat Keras

Perangkat	Spesifikasi
Jenis	Desktop PC
Processor	Intel Core i5 3.1 GHz
RAM	8GB
Hard Disk Drive	3000GB
Sistem Operasi	Windows 10

Sedangkan perangkat lunak yang digunakan pada proses implementasi tugas akhir ini dapat dilihat pada tabel 5.2.

Tabel 5.2: Spesifikasi Perangkat Lunak

Nama Perangkat Lunak	Kegunaan dalam Implementasi
FedX 3.1	Melakukan federated query
Virtuoso Open Source	Local SPARQL Endpoint
RDF4J	Pembuatan Native Store
EmEditor	Text Editor
NetBalancer	Mengetahui penggunaan bandwidth

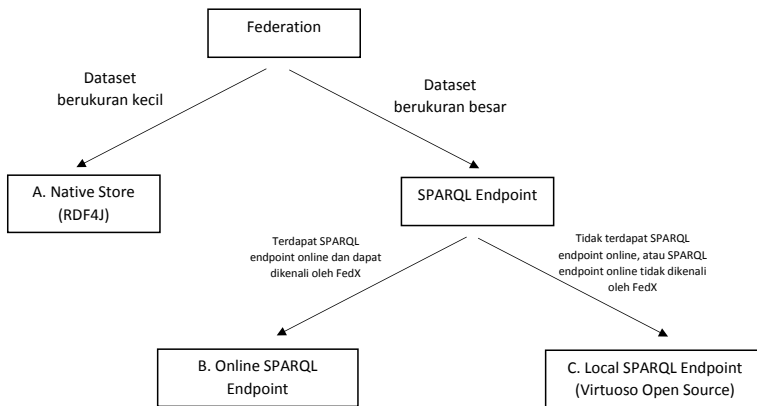
5.2 Pengujian

Pada tahap ini dilakukan pengujian dataset dengan menjalankan query yang telah dibuat pada kode 4.14, kode 4.15, dan kode 4.16 menggunakan aplikasi FedX. Pengujian ini dilakukan untuk mengetahui keterkaitan karakteristik dataset dengan kinerja dari federated SPARQL query engine FedX. Pengujian dilakukan pada dataset tiap-tiap jurnal, dimana terdapat delapan jurnal yang akan diuji.

5.2.1 Pengaturan *Federation* pada FedX

Sebelum query dapat dijalankan pada FedX, perlu dilakukan pengaturan *federation* terlebih dahulu. Pengaturan ini bertujuan untuk menentukan lokasi sumber-sumber data yang nantinya akan diakses. Proses pengaturan *federation* dilakukan untuk setiap jurnal yang akan diuji, karena dataset yang terdapat pada suatu jurnal dengan jurnal yang lain berbeda-beda. FedX mendukung tiga jenis *federation*, yaitu melalui SPARQL endpoint, Sesame remote repositories, dan Local repositories (Native Store RDF4J), serta kombinasi dari ketiganya. Pemilihan jenis *federation* yang akan digunakan tergantung dari kondisi dataset itu sendiri. Pada pengerjaan tugas akhir kali ini, *federation* dilakukan dengan mengkombi-

nasikan SPARQL endpoint dan Native Store, seperti yang terlihat pada gambar 5.1. Sementara pengaturan *federation* untuk masing-masing jurnal dapat dilihat pada tabel 5.3. Pada tabel 5.3, kolom "Pengaturan Federation" merujuk pada gambar 5.1.



Gambar 5.1: Pengaturan Federation

- **Native Store**
Native Store merupakan bentuk repositori RDF yang dimiliki oleh RDF4J dimana penyimpanan data dilakukan pada disk dan tidak pada memory. RDF4J itu sendiri merupakan software yang mampu melakukan analisis dan query data rdf, mirip seperti Apache Jena maupun Stardog. Pada pengaturan *federation* ini, pemilihan Native Store dilakukan apabila dataset yang akan diuji memiliki ukuran yang relatif kecil, sehingga memungkinkan untuk diupload pada Native Store. Untuk menyimpan dataset dalam bentuk Native Store, maka dataset harus diupload melalui software Sesame atau RDF4J.
- **SPARQL Endpoint**
SPARQL endpoint digunakan apabila dataset yang akan diuji memiliki ukuran yang besar dimana dataset dengan ukur-

an tersebut tidak memungkinkan untuk diupload pada local (Native Store) karena biasanya akan mengakibatkan penuhnya memory. Daftar SPARQL endpoint dari dataset dapat dilihat pada tabel 4.1. Namun terdapat suatu permasalahan, yaitu tidak semua dataset berukuran besar memiliki SPARQL endpoint online yang dapat diakses. Selain itu, tidak semua SPARQL endpoint yang dapat diakses melalui web browser dapat dikenali oleh FedX. Beberapa SPARQL Endpoint dapat diakses dan dilakukan query melalui web browser, namun ketika endpoint tersebut diakses oleh FedX, muncul pesan error yang menyatakan bahwa format output tidak didukung. Untuk mengatasi kedua permasalahan tersebut, dapat dilakukan pembuatan SPARQL endpoint yang dapat diakses melalui localhost menggunakan software Virtuoso Open Source. Virtuoso Open Source mendukung dataset dengan ukuran besar.

Hasil pengaturan *federation* kemudian disimpan dalam file .ttl dengan format yang telah ditentukan. File .ttl tersebut nantinya akan di-load oleh FedX ketika eksekusi query dilakukan. Kode 5.1 merupakan contoh pengaturan *federation* yang terdiri dari dataset DBpedia dan Lexvo, dimana DBpedia diambil dari SPARQL endpoint, sementara Lexvo diambil dari Native Store.

Kode 5.1: Contoh Pengaturan Federation

```
@prefix fluid : <http://fluidops.org/config#>.

<http://DBpedia> fluid:store "SPARQLEndpoint";
fluid:SPARQLEndpoint "http://dbpedia.org/sparql"
;
fluid:supportsASKQueries "false" .

<http://Lexvo> fluid:store "NativeStore";
fluid:RepositoryLocation "J:\\Temp\\lexvo".
```

Tabel 5.3: Pengaturan Federation pada Masing-Masing Jurnal

Nomor	Judul Jurnal	Dataset	Pengaturan Federation
1	Countering language attrition with PanLex and the Web of Data	DBpedia	B
		Lexvo	A
2	How Redundant Is It? - An Empirical Analysis on Linked Datasets	DBLP	C
		LinkedMDB	C
		Linked Open Vocabularies	C
3	iRap - An interest-based RDF Update Propagation Framework	DBpedia	B
		LinkedGeoData	B
4	LED: curated and crowdsourced Linked Data on Music Listening Experiences	British National Bibliography	C
		DBpedia	B
5	Lexvo.org: Language Information for the Linguistic Linked Data Cloud	DBpedia	B
		WordNet	A
		Yago	B
6	Property Path over Linked Data: Can It be Done and How to Start?	DBpedia	B
		Wikidata	B
		Yago	B
7	Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	DBpedia	B
		GeoNames	C
		LinkedGeoData	B
8	The Semantic Web Journal as Linked Data	Citeseer	C
		DBLP	C

5.2.2 Eksekusi Query

Setelah pengaturan *federation* pada FedX selesai dilakukan, maka query dapat dijalankan untuk menguji response time dan bandwidth dari dataset-dataset yang terdapat pada suatu jurnal. Proses eksekusi query dilakukan pada dataset pada 8 jurnal yang telah diidentifikasi sebelumnya. Pada masing-masing jurnal, dijalankan 3 query yang telah dibuat, yaitu query pada kode 4.14, kode 4.15, dan kode 4.16. Eksekusi query pada FedX dapat dilakukan dengan menjalankan command berikut pada Command Prompt yang telah diarahkan pada direktori FedX:

Kode 5.2: Eksekusi Query

```
cli -d pengaturanfederation.ttl @q query.txt
```

Pada kode 5.2, "pengaturanfederation.ttl" merupakan lokasi dari file pengaturan *federation* dalam format .ttl. Sementara "query.txt" merupakan lokasi dari file teks yang berisi query yang akan dieksekusi.

Pada saat setiap query dieksekusi, dilakukan penghitungan response time dan bandwidth yang terpakai. Bandwidth yang dimaksud pada pengujian ini yaitu volume data yang masuk dan keluar dari FedX ketika query dieksekusi. Untuk mendapatkan response time dari setiap eksekusi query dapat langsung diketahui pada FedX ketika query telah selesai dieksekusi. Gambar 5.2 menunjukkan tampilan dari FedX ketika eksekusi query telah selesai, dimana pada bagian bawah dapat diketahui durasi waktu yang dibutuhkan hingga FedX menampilkan hasil query. Sementara itu untuk penghitungan bandwidth dilakukan menggunakan bantuan software NetBalancer. NetBalancer merupakan software yang dapat menampilkan volume data yang dipakai oleh suatu aplikasi. Gambar 5.3 menunjukkan tampilan dari NetBalancer ketika eksekusi query pada FedX se-

dang dilakukan. Pada gambar, proses dari FedX tertulis sebagai "java.exe" karena FedX merupakan aplikasi berbasis Java. Dari tampilan NetBalancer tersebut dapat diketahui volume data yang masuk (downloader) maupun keluar (uploaded) pada proses java. Untuk mengetahui volume data yang terpakai, maka tinggal dilakukan penjumlahan volume data yang masuk dan volume data yang keluar.

```

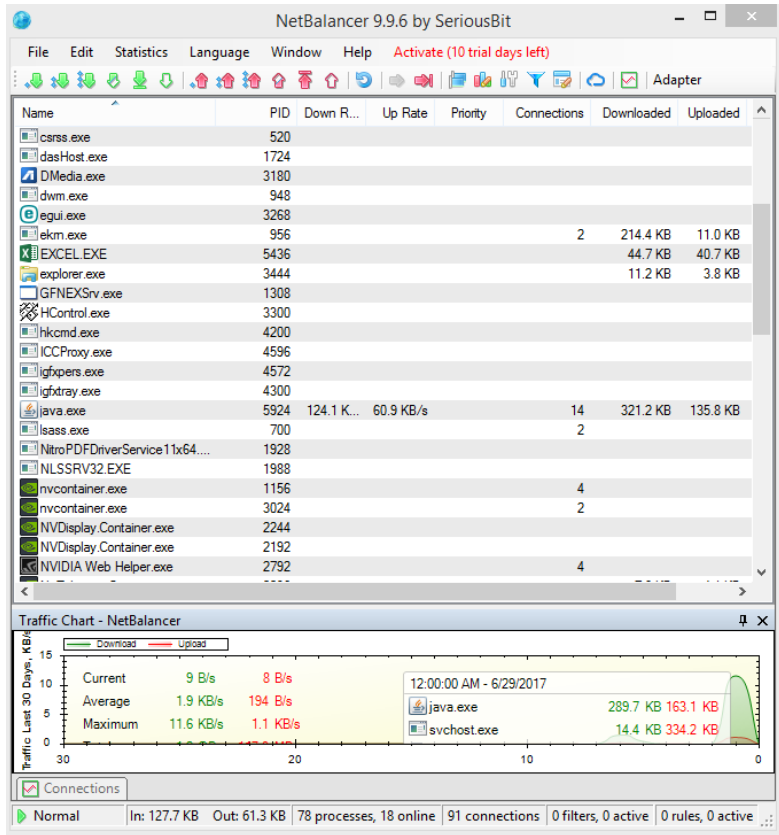
C:\Windows\system32\cmd.exe

C:\Users\Lutfi\Downloads\FedX 3.1>cli -d "F:\Drive E\Kuliah\Semester 8\NTA\Penger
jaan\Doc Per_paper\Property Path over Linked Data Can It be Done and How to Star
t_config.txt" --q1.txt
FedX Cli 3.1 (build 101)
Running Query 1
[comment="Abel Fernando Miglietti (born 4 March 1946 in Maputo) is a former Port
uguese footballer who played as forward.He is the younger brother of Zeca."@en;s
=http://dbpedia.org/resource/Abel_Miglietti;label="Abel Miglietti"@it;sameas=htt
p://purl.org/collections/nl/am/p-256731]
[comment="Abel Fernando Miglietti (born 4 March 1946 in Maputo) is a former Port
uguese footballer who played as forward.He is the younger brother of Zeca."@en;s
=http://dbpedia.org/resource/Abel_Miglietti;label="Abel Miglietti"@it;sameas=htt
p://rdf.freebase.com/ns/m.0cn_96n1]
[comment="Abel Fernando Miglietti (born 4 March 1946 in Maputo) is a former Port
uguese footballer who played as forward.He is the younger brother of Zeca."@en;s
=http://dbpedia.org/resource/Abel_Miglietti;label="Abel Miglietti"@it;sameas=htt
p://www.wikidata.org/entity/Q4666598]
[comment="Abel Fernando Miglietti (born 4 March 1946 in Maputo) is a former Port
uguese footballer who played as forward.He is the younger brother of Zeca."@en;s
=http://dbpedia.org/resource/Abel_Miglietti;label="Abel Miglietti"@it;sameas=htt
p://it.dbpedia.org/resource/Abel_Miglietti]
[comment="Abel Fernando Miglietti (born 4 March 1946 in Maputo) is a former Port
uguese footballer who played as forward.He is the younger brother of Zeca."@en;s
=http://dbpedia.org/resource/Abel_Miglietti;label="Abel Miglietti"@it;sameas=htt
p://wikidata.dbpedia.org/resource/Q4666598]
Done query 1: duration=407733ms, results=5
Done.

C:\Users\Lutfi\Downloads\FedX 3.1>

```

Gambar 5.2: Tampilan dari FedX



Gambar 5.3: Tampilan dari NetBalancer

Halaman ini sengaja dikosongkan

BAB 6

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dan pembahasan dari proses pengujian dataset.

6.1 Hasil Pengujian

Pada bagian ini akan dijelaskan hasil dari pengujian dataset untuk setiap jurnal. Terdapat dua jenis keluaran dari proses pengujian yang telah dilakukan, yaitu response time dan bandwidth. Dari beberapa kali percobaan eksekusi query yang dilakukan pada FedX, diketahui bahwa response time yang dihasilkan cenderung bervariasi antara eksekusi satu dengan yang lain, sedangkan bandwidth yang dihasilkan cenderung sama. Oleh karena itu, untuk mendapatkan hasil pengujian yang lebih akurat, maka proses pengujian dilakukan sebanyak tiga kali. Hasil akhir pengujian didapatkan dengan cara mencari rata-rata dari hasil pengujian 1, pengujian 2, dan pengujian 3. Hasil akhir dari pengujian yang telah dirata-rata terdapat pada tabel 6.1. Pada tabel tersebut, query 1 merujuk pada kode 4.14, query 2 merujuk pada kode 4.15, dan query 3 merujuk pada kode 4.16.

Dari tiga kali pengujian yang telah dilakukan, terjadi beberapa permasalahan yang menyebabkan proses pengujian tidak berjalan sesuai dengan yang diharapkan.

- Permasalahan pertama terdapat pada jurnal yang berjudul "How Redundant Is It? - An Empirical Analysis on Linked Datasets". Permasalahan yang terjadi yaitu munculnya error "Not

a valid URI” ketika query dijalankan pada dataset dari jurnal tersebut. Sebelum query dilakukan, dataset pada jurnal yang meliputi DBLP, LinkedMDB, dan Linked Open Vocabularies di-*load* pada local SPARQL endpoint menggunakan Virtuoso Open Source. Proses *load* pada local SPARQL endpoint berjalan lancar tanpa terjadi error, namun ketika dilakukan query pada endpoint tersebut menggunakan FedX, terjadi error yang menyebabkan tidak munculnya hasil dari eksekusi query. Permasalahan ini kemungkinan disebabkan oleh adanya format penulisan URI yang tidak valid pada file dataset, serta proses pembacaan data yang terlalu ketat (*strict*) yang dilakukan oleh FedX. Hal ini karena kesalahan penulisan tersebut tidak dideteksi oleh Virtuoso Open Source sehingga proses *load* dataset berjalan tanpa terjadi error. Namun kesalahan tersebut ternyata dapat dideteksi oleh FedX sehingga menyebabkan query tidak dapat dilakukan.

- Permasalahan kedua terdapat pada jurnal dengan judul ”The Semantic Web Journal as Linked Data”. Permasalahan yang terjadi yaitu query 1 dan query 2 memunculkan hasil 0 setelah query selesai dieksekusi, sehingga menyebabkan response time dan bandwidth tidak dapat diketahui. Permasalahan ini disebabkan oleh kelas atau properties yang dicari dalam query 1 dan query 2 tidak ditemukan dalam dataset.
- Permasalahan ketiga terdapat pada jurnal berjudul ”Lexvo.org Language Information for the Linguistic Linked Data Cloud”. Pada jurnal ini, query 1 dan query 2 dapat dieksekusi dan hasil dari eksekusi juga muncul. Sementara itu, query 3 dapat dieksekusi, namun hasil eksekusi tidak kunjung muncul, dan pada akhirnya muncul error *out of memory*.

Dengan permasalahan yang terjadi di atas, maka diputuskan bahwa jurnal dengan judul ”How Redundant Is It? - An Empirical Analysis on Linked Datasets” tidak diikuti dalam proses ana-

lisis. Sementara jurnal dengan judul "The Semantic Web Journal as Linked Data", hanya hasil query 3 yang diikuti ke dalam proses analisis. Sedangkan jurnal dengan judul "Lexvo.org Language Information for the Linguistic Linked Data Cloud", hanya hasil query 1 dan query 2 yang dimasukkan dalam proses analisis.

6.2 Pembahasan

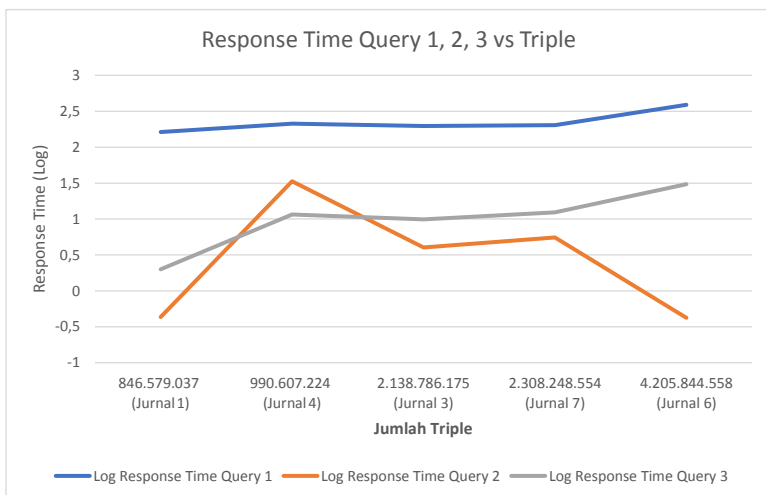
Pada subbab ini akan dilakukan pembahasan dan analisis terkait hasil dari pengujian dataset. Hasil dari analisis yang berupa response time dan bandwidth akan dibandingkan dengan beberapa karakteristik dataset untuk mengetahui keterkaitan antara karakteristik dataset dengan response time dan bandwidth.

Untuk mempersingkat penyebutan jurnal, maka jurnal akan ditulis menggunakan nomor jurnal, seperti yang terdapat pada tabel 5.3.

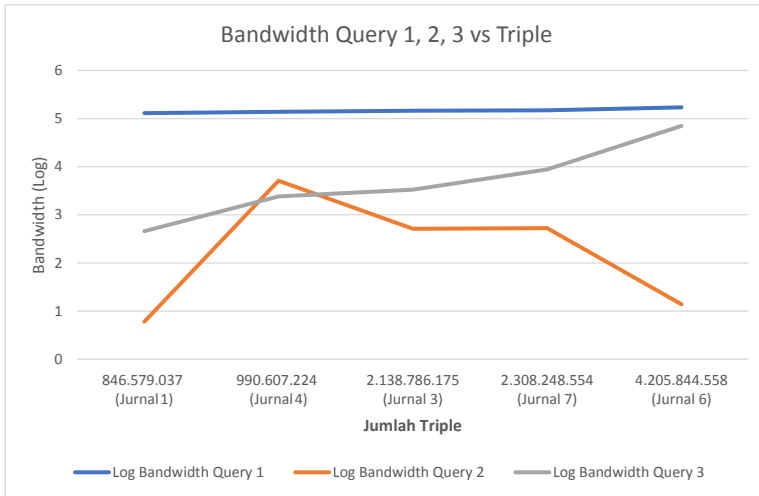
Tabel 6.1: Hasil Pengujian Setelah Dirata-rata

Jurnal	Query 1		Query 2		Query 3	
	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)
Countering language attrition with Panlex and the Web of Data	162,62	129.500	0,4323	6	1,997	457
How Redundant Is It? - An Empirical Analysis on Linked Datasets	-	-	-	-	-	-
IRap - An interest-based RDF Update Propagation Framework	197,366	144.900	4,0313	512	9,925	3.351
LED: curated and crowdsourced Linked Data on Music Listening Experiences	213,282	138.200	33,559	5.100	11,542	2.398,4
Lexvo.org: Language Information for the Linguistic Linked Data Cloud	303,565	150.300	0,427	6,1	-	-
Property Path over Linked Data: Can It be Done and How to Start?	388,841	171.100	0,422	13,8	30,556	70.400
Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	203,239	147.900	5,546	526,6	12,408	8.789,7
The Semantic Web Journal as Linked Data	0	0	0	0	18,53	14,6

6.2.1 Response Time dan Bandwidth Query 1, 2, dan 3 vs Triple



Gambar 6.1: Grafik Response Time Query 1, 2, dan 3 vs Triple



Gambar 6.2: Grafik Bandwidth Query 1, 2, dan 3 vs Triple

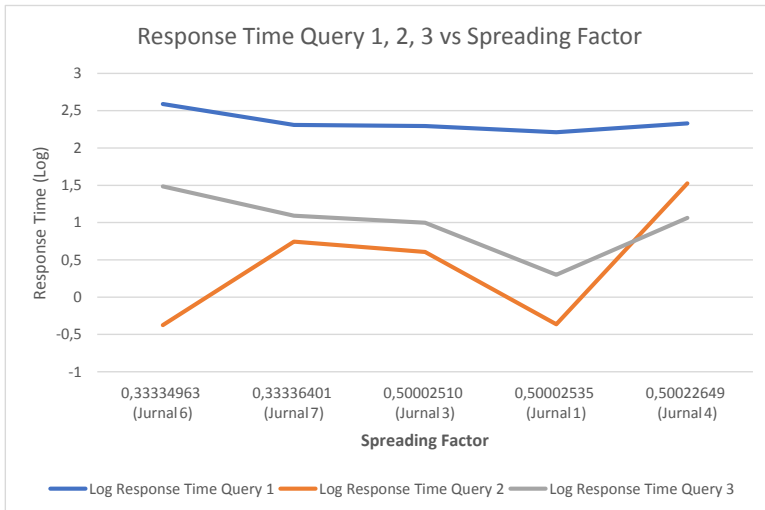
Pada gambar 6.1 dan gambar 6.2, dapat dilihat pada pola grafik bahwa response time dan bandwidth pada query 1 dan 3 mengalami peningkatan, dimana hal tersebut diakibatkan oleh terjadinya peningkatan jumlah triple, yang menunjukkan bahwa jumlah triple berpengaruh terhadap response time dan bandwidth. Pada grafik bandwidth query 1, grafik terlihat mendatar, namun sebenarnya terjadi peningkatan bandwidth meskipun sedikit.

Sementara itu, pada grafik response time dan bandwidth query 2 terjadi sedikit perbedaan, dimana terjadi peningkatan yang signifikan pada jurnal 4. Peningkatan yang signifikan tersebut terjadi karena kelas dalam query 2 yang ditemukan pada dataset jurnal 4 jauh lebih banyak dibanding dataset pada jurnal lain, seperti yang terlihat pada tabel 6.2, meskipun jumlah property jurnal 4 yang terlibat dalam query 2 cenderung lebih rendah dibanding jurnal lain, terutama jurnal 3 dan jurnal 7. Hal itu mengakibatkan response

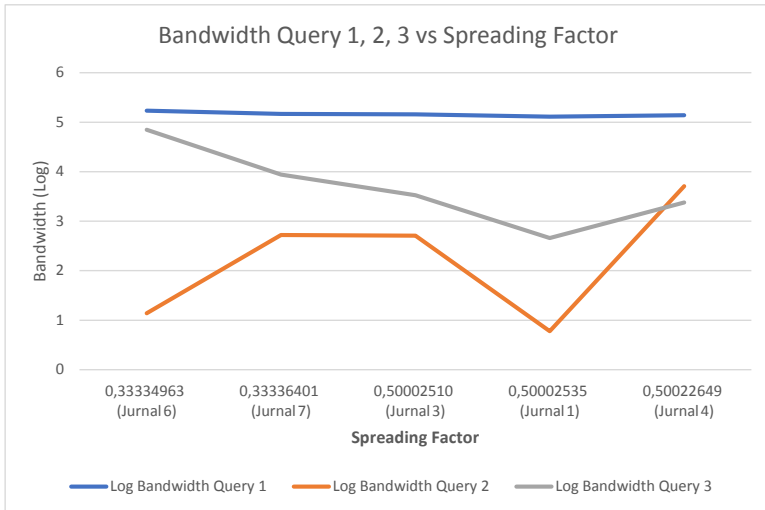
time dan bandwidth yang lebih tinggi pada jurnal 4 ketika query 2 dieksekusi. Dapat dikatakan bahwa jumlah kelas yang terlibat dalam query berpengaruh terhadap response time dan bandwidth, sementara jumlah property yang terlibat dalam query cenderung tidak begitu berpengaruh terhadap response time dan bandwidth. Sementara itu, pada grafik query 2, terjadi penurunan response time dan bandwidth pada jurnal 7 jika dibandingkan dengan jurnal 6, berbeda dengan pada query 1 dan query 3 dimana terjadi kenaikan. Hal tersebut kemungkinan disebabkan oleh kelas dan property yang terdapat pada query 2 tidak banyak ditemukan pada jurnal 6, sehingga menyebabkan response time dan bandwidth menjadi lebih rendah. Namun hal tersebut belum dapat dipastikan, karena salah satu sumber data pada jurnal 6 yaitu Yago SPARQL endpoint tidak dapat diakses pada saat proses analisis ini, sehingga penghitungan jumlah kelas dan property query 2 yang terkandung dalam jurnal 6 tidak dapat dilakukan.

Pada gambar 6.1 dan gambar 6.2 di atas, jenis query yang dieksekusi (star query atau chain query) terlihat tidak berpengaruh terhadap response time dan bandwidth. Dapat dilihat bahwa dua jenis query yang berbeda, yaitu query 2 (star query) dan query 3 (chain query) cenderung memiliki response time dan bandwidth yang berada pada rentang yang sama. Sementara itu, dua jenis query yang sama, yaitu query 1 (star query) dan query2 (star query) memiliki response time dan bandwidth yang berbeda, dimana query 1 (star query) memiliki response time dan bandwidth yang lebih tinggi dibanding query 2 serta query 3. Hal itu terjadi karena baris yang terdapat pada query 1 yang memang lebih banyak dibanding pada query 2 dan query 3. Oleh karena itu, jumlah baris yang terdapat pada query cenderung berpengaruh terhadap bandwidth dan response time yang dibutuhkan ketika eksekusi query dilakukan.

6.2.2 Response Time dan Bandwidth Query 1, 2, dan 3 vs Spreading Factor



Gambar 6.3: Grafik Response Time Query 1, 2, dan 3 vs Spreading Factor



Gambar 6.4: Grafik Bandwidth Query 1, 2, dan 3 vs Spreading Factor

Pada gambar 6.3 dan gambar 6.4 di atas, terdapat kecenderungan pada grafik 1 dan 3 untuk mengalami penurunan dan kenaikan seiring dengan meningkatnya spreading factor. Pada pola grafik bandwidth query 1, grafik cenderung terlihat mendatar, namun sebenarnya pola grafik dari jurnal 6 hingga jurnal 1 mengalami sedikit penurunan, sementara pola grafik dari jurnal 1 ke jurnal 4 mengalami sedikit kenaikan. Namun kecenderungan kenaikan dan penurunan yang tidak konsisten tersebut lebih dipengaruhi oleh jumlah triple yang terdapat pada dataset. Data jumlah triple pada gambar 6.1 terlihat sesuai dengan grafik query 1 dan 2 yang terdapat pada kedua gambar tersebut. Jika dilihat pada spreading factor jurnal 7 dibandingkan dengan spreading factor jurnal 3 dimana terjadi kenaikan spreading factor yang cukup besar, maka pada grafik justru tidak terjadi kenaikan response time dan bandwidth, malah cenderung

menurun. Hal ini disebabkan oleh jumlah triple yang juga menurun. Sementara itu grafik query 2 terjadi sedikit perbedaan, dimana terjadi peningkatan signifikan pada bandwidth jurnal 4. Hal itu disebabkan jumlah kelas dalam query 2 yang ditemukan dalam dataset pada jurnal 4 jauh lebih banyak dibanding dataset pada jurnal lain, seperti yang terlihat pada tabel 6.2. Akibatnya, response time dan bandwidth yang dibutuhkan juga lebih tinggi. Sementara itu, jurnal 6 pada query 2 memiliki response time dan bandwidth yang rendah, berbeda dengan query 1 dan query 3 yang mempunyai response time dan bandwidth lebih tinggi. Seperti yang sudah dijelaskan sebelumnya, hal tersebut kemungkinan disebabkan oleh kelas dan property yang terdapat pada query 2 tidak banyak ditemukan pada jurnal 6, sehingga menyebabkan response time dan bandwidth menjadi lebih rendah. Dapat disimpulkan bahwa pengaruh spreading factor terhadap response time dan bandwidth cenderung kecil hingga tidak ada.

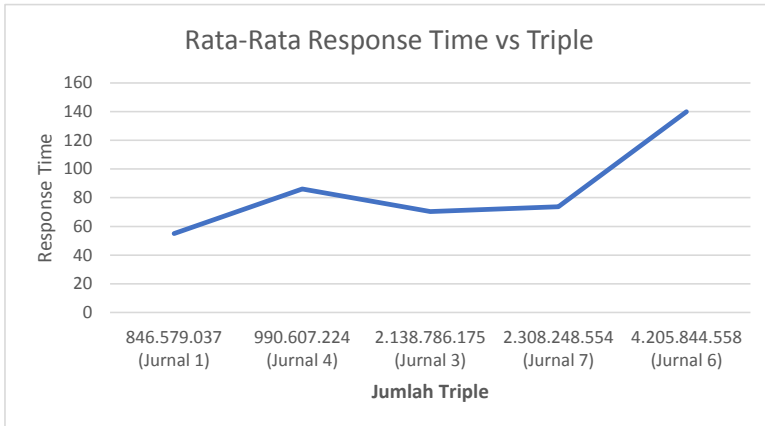
Pada gambar 6.3 dan gambar 6.4 di atas, jenis query yang dieksekusi (star query atau chain query) terlihat juga tidak berpengaruh terhadap response time dan bandwidth. Dapat dilihat bahwa dua jenis query yang berbeda, yaitu query 2 (star query) dan query 3 (chain query) cenderung memiliki response time dan bandwidth yang berada pada rentang yang sama. Sementara itu, dua jenis query yang sama, yaitu query 1 (star query) dan query 2 (star query) memiliki response time dan bandwidth yang berbeda, dimana query 1 (star query) memiliki response time dan bandwidth yang lebih tinggi dibanding query 2 serta query 3. Hal itu terjadi karena baris yang terdapat pada query 1 yang memang lebih banyak dibanding pada query 2 dan query 3. Oleh karena itu, jumlah baris yang terdapat pada query cenderung berpengaruh terhadap bandwidth dan response time yang dibutuhkan ketika eksekusi query dilakukan.

Tabel 6.2: Jumlah Kelas dan Property Tiap Jurnal yang Terkait dengan Query

Jurnal	Query 1		Query 2		Query 3	
	Kelas	Property	Kelas	Property	Kelas	Property
Jurnal 1	1.243.399	181.264.891	3.688	117.506.495	0	33.825.067
Jurnal 2	-	-	-	-	-	-
Jurnal 3	1.243.399	595.525.863	3.688	517.281.714	0	34.406.578
Jurnal 4	2.734.418	228.213.802	156.119	141.858.268	0	42.990.737
Jurnal 5	-	-	-	-	-	-
Jurnal 6	-	-	-	-	-	-
Jurnal 7	1.243.399	606.867.613	3.688	528.623.423	0	34.932.634
Jurnal 8	0	0	0	0	0	18.390.575

- Jurnal 2 tidak diikuti dalam proses analisis karena mengalami error pada saat proses pengujian. Sehingga jumlah kelas dan property tidak dihitung
- Pada jurnal 5 dan 6, jumlah kelas dan property yang terkait dengan query tidak dapat dihitung karena salah satu endpoint dari dataset pada kedua jurnal tersebut, yaitu Yago tidak dapat diakses
- Jumlah kelas yang terkait dengan query pada query 3 bernilai 0, karena query 3 tidak mengandung kelas.
- Pada jurnal 8, response time dan bandwidth pada query 1 dan 2 tidak diketahui. Sehingga jumlah kelas dan property pada query 1 dan 2 tidak perlu dihitung

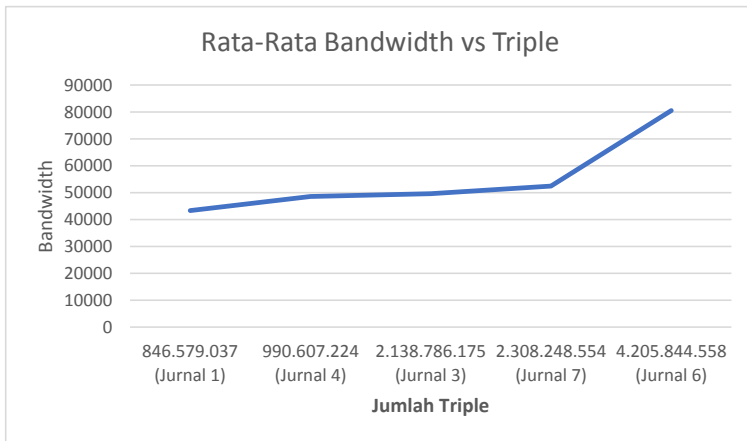
6.2.3 Rata-Rata Response Time Query vs Triple



Gambar 6.5: Grafik Rata-Rata Response Time Query vs Triple

Rata-rata response time pada bagian ini didapatkan dengan cara mencari rata-rata response time dari query 1, query 2, dan query 3, begitu pula untuk rata-rata bandwidth, yang mana akan dijelaskan pada bagian selanjutnya. Pada gambar 6.5, terlihat bahwa response time cenderung meningkat seiring dengan bertambahnya jumlah triple pada jurnal. Pada jurnal 4, seperti yang telah dijelaskan sebelumnya, response time yang cenderung lebih tinggi disebabkan oleh kelas pada query 2 dan query 3 yang terkandung pada jurnal 4 yang memang jauh lebih banyak dibandingkan jurnal-jurnal lain, seperti yang terlihat pada tabel 6.2. Akibatnya, response time yang dibutuhkan juga mengalami peningkatan.

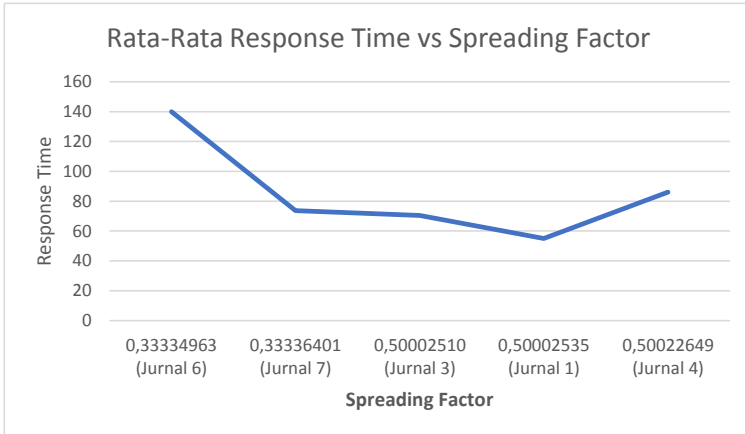
6.2.4 Rata-Rata Bandwidth Query vs Triple



Gambar 6.6: Grafik Rata-Rata Bandwidth Query vs Triple

Pada gambar 6.6, terlihat bahwa rata-rata bandwidth mengalami peningkatan seiring dengan meningkatnya jumlah triple pada dataset. Rata-rata bandwidth jurnal 6 mengalami peningkatan signifikan dibandingkan rata-rata bandwidth jurnal 7 karena jumlah triple pada jurnal 6 juga memiliki perbedaan yang besar dengan jumlah triple pada jurnal 7.

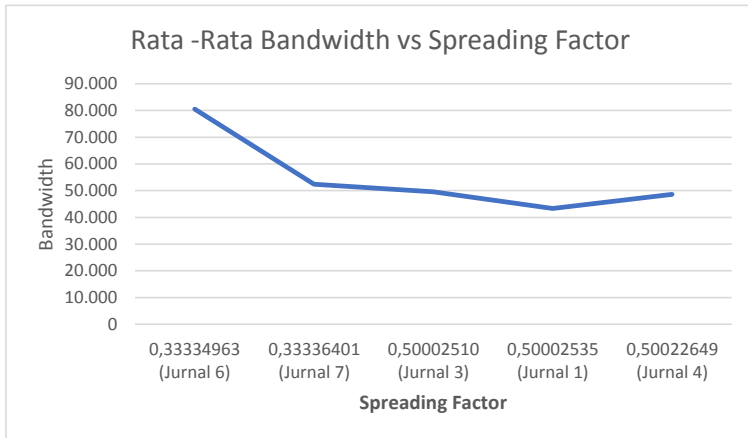
6.2.5 Rata-Rata Response Time Query vs Spreading Factor



Gambar 6.7: Grafik Rata-Rata Response Time Query vs Spreading Factor

Seperti yang terlihat pada gambar 6.7, grafik response time cenderung tidak menentu, dimana terjadi penurunan dan kenaikan seiring dengan meningkatnya spreading factor. Grafik tersebut lebih dipengaruhi oleh jumlah triple pada dataset. Data jumlah triple yang terdapat pada gambar 6.1 sesuai dengan pola grafik pada gambar 6.7 di atas. Jika dilihat pada spreading factor jurnal 7 dibandingkan spreading factor jurnal 3 dimana terjadi kenaikan spreading factor yang cukup besar, maka pada grafik justru terjadi penurunan rata-rata response time, yang disebabkan oleh jumlah triple yang juga menurun. Oleh karena itu, dapat disimpulkan bahwa spreading factor hampir tidak memberikan pengaruh terhadap response time.

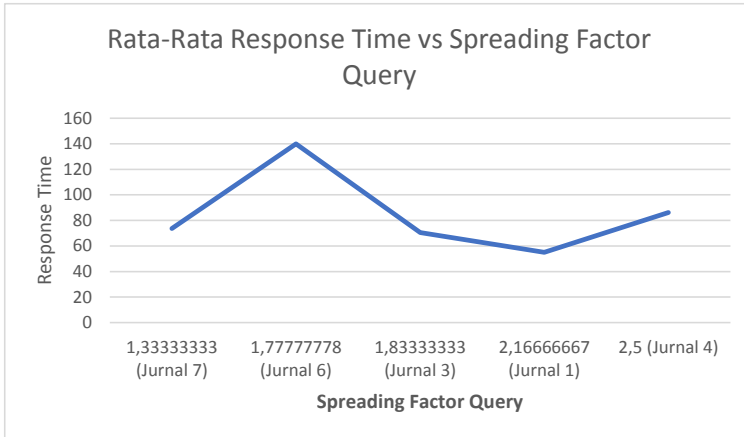
6.2.6 Rata-Rata Bandwidth Query vs Spreading Factor



Gambar 6.8: Grafik Rata-Rata Bandwidth Query vs Spreading Factor

Pada gambar 6.8, grafik bandwidth cenderung dapat mengalami penurunan maupun kenaikan seiring dengan meningkatnya spreading factor. Grafik tersebut lebih dipengaruhi oleh jumlah triple pada dataset. Data jumlah triple yang terdapat pada gambar 6.1 sesuai dengan pola grafik pada gambar 6.8 di atas. Jika dilihat pada spreading factor jurnal 7 dibandingkan dengan spreading factor jurnal 3, dimana terjadi kenaikan spreading factor yang cukup besar, maka pada grafik justru terjadi penurunan rata-rata bandwidth, yang disebabkan oleh jumlah triple yang juga menurun. Oleh karena itu, dapat disimpulkan bahwa spreading factor hampir tidak memberikan pengaruh terhadap bandwidth.

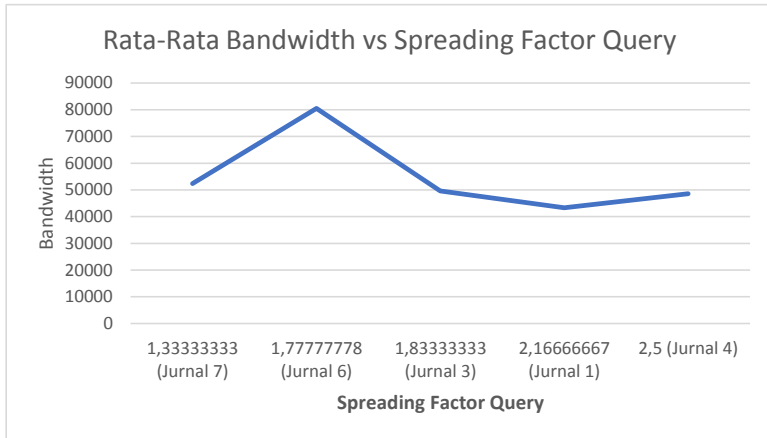
6.2.7 Rata-Rata Response Time Query vs Spreading Factor Query



Gambar 6.9: Grafik Rata-Rata Response Time Query vs Spreading Factor Query

Pada gambar 6.9 diatas, pola grafik rata-rata response time cenderung tidak menentu, dimana seiring dengan meningkatnya spreading factor query, rata-rata response time dapat mengalami peningkatan maupun penurunan. Grafik pada gambar tersebut cenderung lebih dipengaruhi oleh jumlah triple, dimana semakin banyak jumlah triple, maka semakin tinggi juga rata-rata response time. Data jumlah triple yang terdapat pada gambar 6.1 cenderung sesuai dengan pola grafik pada gambar 6.9 di atas. Dapat disimpulkan bahwa spreading factor query mempunyai pengaruh yang sedikit hingga tidak berpengaruh terhadap rata-rata response time.

6.2.8 Rata-Rata Bandwidth Query vs Spreading Factor Query



Gambar 6.10: Grafik Rata-Rata Bandwidth Query vs Spreading Factor Query

Pada gambar 6.10 diatas, pola grafik rata-rata bandwidth juga cenderung tidak menentu, dimana seiring dengan bertambahnya spreading factor query, rata-rata response time dapat mengalami peningkatan maupun penurunan. Grafik pada gambar tersebut cenderung lebih dipengaruhi oleh jumlah triple, dimana semakin banyak jumlah triple, maka semakin tinggi juga rata-rata bandwidthnya. Data jumlah triple yang terdapat pada gambar 6.1 cenderung sesuai dengan pola grafik pada gambar 6.10 di atas. Dapat disimpulkan bahwa spreading factor query mempunyai pengaruh yang sedikit hingga tidak berpengaruh terhadap rata-rata bandwidth.

Halaman ini sengaja dikosongkan

BAB 7

KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan kesimpulan dan saran dalam pengerjaan tugas akhir.

7.1 Kesimpulan

Berdasarkan pengerjaan tugas akhir yang telah dilakukan dengan judul "Identifikasi Karakteristik Dataset untuk Federated SPARQL Query", maka dapat disimpulkan beberapa hal sebagai berikut:

1. Suatu dataset RDF memiliki beberapa karakteristik yang membedakannya dengan dataset RDF lain, yang meliputi jumlah triple, jumlah kelas, jumlah property, jumlah distinct subjek, jumlah distinct entity, jumlah distinct objek, serta spreading factor dataset. Proses identifikasi karakteristik dataset RDF dapat dilakukan dengan mengeksekusi query yang berfungsi mendapatkan karakteristik-karakteristik tersebut, dimana dataset yang akan diidentifikasi sudah di-*load* terlebih dahulu pada aplikasi pemroses data RDF yang mendukung query. Sedangkan proses identifikasi spreading factor dilakukan dengan penghitungan menggunakan rumus.
2. Jumlah triple dan jumlah kelas yang terlibat dalam query pada suatu dataset memiliki kecenderungan untuk berpengaruh terhadap bandwidth maupun response time yang dibutuhkan oleh federated SPARQL query engine untuk melakukan query pada dataset tersebut. Apabila jumlah triple dan jumlah kelas yang terlibat dalam query mengalami peningkatan, bandwidth dan response time yang dibutuhkan ketika query

- dieksekusi juga akan mengalami peningkatan.
3. Jumlah property yang terlibat pada query, spreading factor, dan spreading factor berdasarkan query dari suatu dataset cenderung memiliki pengaruh yang kecil hingga tidak berpengaruh terhadap response time dan bandwidth yang dibutuhkan oleh federated SPARQL query engine untuk melakukan query pada dataset tersebut.
 4. Bandwidth dan response time yang dibutuhkan ketika suatu query dieksekusi cenderung tidak dipengaruhi oleh jenis query (star query atau chain query), namun dipengaruhi oleh jumlah *tuple* (baris) yang terdapat pada query tersebut.

7.2 Saran

Dalam pengerjaan tugas akhir ini, masih terdapat beberapa kekurangan yang perlu untuk diperbaiki demi mewujudkan penelitian selanjutnya yang lebih baik lagi. Saran penulis untuk penelitian selanjutnya sebagai berikut:

1. Akibat keterbatasan spesifikasi komputer yang digunakan, sumber data pada proses pengujian menggunakan FedX berasal dari kombinasi antara SPARQL endpoint online serta Native Store RDF4J yang berasal dari dataset yang telah didownload (offline). Sementara proses identifikasi karakteristik dataset sendiri hanya dilakukan secara offline dengan cara mendownload dataset. Hal ini menimbulkan sedikit permasalahan karena ada kemungkinan terdapat perbedaan data antara SPARQL endpoint dengan dataset yang tersedia untuk didownload, biasanya karena dataset yang tersedia untuk didownload tidak diperbarui sesuai dengan SPARQL endpointnya atau sebaliknya. Sehingga ketika dilakukan perbandingan antara karakteristik dataset (dilakukan secara offline)

dengan hasil dari pengujian (sebagian dilakukan secara online), ada kemungkinan kedua data yang dibandingkan tersebut berasal dari sumber data yang tidak benar-benar sama. Oleh karena itu, pada penelitian selanjutnya diharapkan proses identifikasi karakteristik dataset dan proses pengujian dapat dilakukan menggunakan sumber data yang sama secara keseluruhan (online saja atau offline saja) untuk memastikan kesamaan data ketika dilakukan perbandingan pada saat proses analisis.

2. Pada penelitian ini, proses pengujian hanya diulang sebanyak tiga kali dan hanya dilakukan dengan tiga jenis query. Untuk lebih meningkatkan keakuratan hasil pengujian, pada penelitian selanjutnya jumlah proses pengujian dapat ditambah. Selain itu, jenis atau variasi dari query juga dapat ditambah.
3. Pada penelitian ini, penulis mengalami beberapa permasalahan terkait dataset yang mengakibatkan proses identifikasi dataset menjadi sedikit terhambat. Diantaranya yaitu ukuran dataset yang terlalu besar sehingga menyulitkan proses identifikasi data. Selain itu, terdapat juga beberapa dataset yang mempunyai kesalahan pada penulisan formatnya yang mengakibatkan dataset tidak dapat di-load pada software pengolah data RDF dengan baik. Untuk penelitian selanjutnya, pemilihan dataset perlu dilakukan dengan lebih baik lagi, seperti dengan memperhitungkan ukuran dataset sesuai dengan spesifikasi komputer yang mengolahnya, serta dengan mendownload dataset dari situs asli penyedia dataset.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Convert GeoNames RDF dump format into ntriples. Available at: <https://gist.github.com/baskaufs/54207ab81eee4f9aa468137df5967d30>.
- [2] RDF - semantic web standards. Available at: <https://www.w3.org/RDF/>.
- [3] Richard Cyganiak. An RDF schema and associated documentation for expressing metadata about RDF datasets. Available at: <https://github.com/cygri/void/blob/master/archive/google-code-wiki/SPARQLQueriesForStatistics.md>.
- [4] Vianney le Clément de Saint-Marcq, Yves Deville, Christine Solnon, and Pierre-Antoine Champin. Castor: a constraint-based sparql engine with active filter processing. In *Extended Semantic Web Conference*, pages 391–405. Springer, 2012.
- [5] Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, and Octavian Udrea. Apples and oranges: a comparison of rdf benchmarks and real rdf datasets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 145–156. ACM, 2011.
- [6] Thomas Holst. *Structural analysis of unknown RDF datasets via SPARQL endpoints*. PhD thesis, Master thesis defense 11, 2013.
- [7] Prasad Kulkarni. Distributed sparql query engine using map-reduce. *Master of Science, Computer Science, School of Informatics, University of Edinburgh*, 2010.
- [8] Mulugeta Mammo and Srividya K Bansal. Distributed sparql over big rdf data: A comparative analysis using presto and

- mapreduce. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 33–40. IEEE, 2015.
- [9] Gabriela Montoya, Maria-Esther Vidal, Oscar Corcho, Edna Ruckhaus, and Carlos Buil-Aranda. Benchmarking federated sparql query engines: Are existing testbeds enough? In *International Semantic Web Conference*, pages 313–324. Springer, 2012.
- [10] Nur Aini Rakhmawati, Marcel Karnstedt, Michael Hausenblas, and Stefan Decker. On metrics for measuring fragmentation of federation over sparql endpoints. In *WEBIST (1)*, pages 119–126, 2014.
- [11] Nur Aini Rakhmawati, Jürgen Umbrich, Marcel Karnstedt, Ali Hasnain, and Michael Hausenblas. Querying over federated sparql endpoints—a state of the art survey. *arXiv preprint arXiv:1306.1723*, 2013.
- [12] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. Fedx: a federation layer for distributed query processing on linked open data. In *Extended Semantic Web Conference*, pages 481–486. Springer, 2011.
- [13] Liyang Yu. *A developer’s guide to the semantic Web*. Springer Science & Business Media, 2011.

LAMPIRAN A

KODE PROGRAM

Kode A.1: *Script* untuk Konversi Dataset GeoNames ke dalam Format N-Triples

```
import rdflib
fo = open("geonames.nt", "wb")
totalStmt = 0
with open("all-geonames-rdf.txt", encoding="utf8
") as fileobject:
    count = 0
    for line in fileobject:
        if count/10000 == int(count/10000):
            print(count)

        if count%2 != 0:
            g = rdflib.Graph()
            result = g.parse(data=line, format=
                'xml')
            totalStmt += len(g)
            s = g.serialize(format='nt')
            fo.write(s)

        count = count + 1

print ("Total statements: ", totalStmt)
fo.close()
```

Halaman ini sengaja dikosongkan

LAMPIRAN B

DAFTAR KELAS DAN PROPERTY TERBANYAK PADA SEMUA DATASET

Tabel B.2: Daftar Property Terbanyak

Property	Jumlah Kemunculan pada Semua Dataset
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	16
http://www.w3.org/2000/01/rdf-schema#label	13
http://www.w3.org/2002/07/owl#sameAs	10
http://xmlns.com/foaf/0.1/page	5
http://www.w3.org/2004/02/skos/core#prefLabel	5
http://www.w3.org/2000/01/rdf-schema#comment	5

Tabel B.1: Daftar Kelas Terbanyak

Kelas	Jumlah Kemunculan pada Semua Dataset
http://xmlns.com/foaf/0.1/Person	4
http://xmlns.com/foaf/0.1/Agent	3
http://xmlns.com/foaf/0.1/Organization	2
http://xmlns.com/foaf/0.1/Document	2
http://purl.org/ontology/bibo/Book	2

Tabel B.2: Daftar Property Terbanyak

Property	Jumlah Kemunculan pada Semua Dataset
http://purl.org/dc/terms/title	5
http://purl.org/dc/terms/description	5
http://xmlns.com/foaf/0.1/name	4
http://xmlns.com/foaf/0.1/homepage	4
http://www.w3.org/2000/01/rdf-schema#subPropertyOf	4
http://www.w3.org/2000/01/rdf-schema#subClassOf	4
http://www.w3.org/2000/01/rdf-schema#seeAlso	4
http://www.w3.org/2000/01/rdf-schema#range	4
http://www.w3.org/2000/01/rdf-schema#domain	4
http://purl.org/dc/terms/contributor	4
http://www.w3.org/2004/02/skos/core#altLabel	3
http://www.w3.org/2003/01/geo/wgs84_pos#long	3
http://www.w3.org/2003/01/geo/wgs84_pos#lat	3
http://www.w3.org/2002/07/owl#imports	3
http://www.w3.org/2000/01/rdf-schema#isDefinedBy	3
http://purl.org/dc/terms/language	3
http://purl.org/dc/terms/creator	3
http://purl.org/dc/elements/1.1/title	3

Tabel B.2: Daftar Property Terbanyak

Property	Jumlah Kemunculan pada Semua Dataset
http://xmlns.com/foaf/0.1/primaryTopic	2
http://xmlns.com/foaf/0.1/givenName	2
http://xmlns.com/foaf/0.1/focus	2
http://xmlns.com/foaf/0.1/familyName	2
http://xmlns.com/foaf/0.1/depiction	2
http://www.w3.org/ns/prov#wasDerivedFrom	2
http://www.w3.org/2004/02/skos/core#broader	2
http://www.w3.org/2002/07/owl#unionOf	2
http://www.w3.org/2002/07/owl#someValuesFrom	2
http://www.w3.org/2002/07/owl#onProperty	2
http://www.w3.org/2002/07/owl#disjointWith	2
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest	2
http://www.w3.org/1999/02/22-rdf-syntax-ns#first	2
http://purl.org/dc/terms/tableOfContents	2
http://purl.org/dc/terms/subject	2
http://purl.org/dc/terms/modified	2
http://purl.org/dc/terms/license	2
http://purl.org/dc/terms/issued	2
http://purl.org/dc/terms/isPartOf	2

Tabel B.2: Daftar Property Terbanyak

Property	Jumlah Kemunculan pada Semua Dataset
http://purl.org/dc/terms/hasPart	2
http://purl.org/dc/elements/1.1/type	2
http://purl.org/dc/elements/1.1/source	2
http://purl.org/dc/elements/1.1/description	2
http://purl.org/dc/elements/1.1/creator	2
http://lexvo.org/ontology#iso639P3PCode	2
http://lexvo.org/ontology#iso639P1Code	2

LAMPIRAN C

HASIL PENGUJIAN

Tabel C.1: Hasil Pengujian Pertama

Journal	Query 1		Query 2		Query 3	
	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)
Countering language attrition with PanLex and the Web of Data	184.888	129500	0.375	6	1.616	457
How Redundant Is It? - An Empirical Analysis on Linked Datasets	-	-	-	-	-	-
iRap - An interest-based RDF Update Propagation Framework	164.762	144900	4.11	512	14.868	3351
LED: curated and crowdsourced Linked Data on Music Listening Experiences	161.509	138200	29.771	5100	7.36	2398.4
Lexvo.org: Language Information for the Linguistic Linked Data Cloud	238.47	150300	0.328	6.1	-	-
Property Path over Linked Data: Can It be Done and How to Start?	345.314	171100	0.328	13.8	29.111	70400
Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	173.832	147900	5.438	526.6	18.234	8789.7
The Semantic Web Journal as Linked Data	0	0	0	0	18.083	14.6

Tabel C.2: Hasil Pengujian Kedua

Jurnal	Query 1		Query 2		Query 3	
	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)
Countering language attrition with Panlex and the Web of Data	142.79	129500	0.469	6	2.532	457
How Redundant Is It? - An Empirical Analysis on Linked Datasets	-	-	-	-	-	-
iRrap - An interest-based RDF Update Propagation Framework	214.575	144900	4.109	512	7.219	3351
LED: curated and crowdsourced Linked Data on Music Listening Experiences	233.928	138200	32.557	5100	9.141	2398.4
Lexvo.org: Language Information for the Linguistic Linked Data Cloud	344.398	150300	0.484	6.1	-	-
Property Path over Linked Data: Can It be Done and How to Start?	407.733	171100	0.453	13.8	31.835	70400
Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	218.435	147900	5.746	526.6	11.864	8789.7
The Semantic Web Journal as Linked Data	0	0	0	0	17.99	14.6

Tabel C.3: Hasil Pengujian Ketiga

Journal	Query 1		Query 2		Query 3	
	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)	Response Time (s)	Bandwidth (kilobyte)
Countering language attrition with PanLex and the Web of Data	160.183	129500	0.453	6	1.843	457
How Redundant Is It? - An Empirical Analysis on Linked Datasets	-	-	-	-	-	-
iRap - An interest-based RDF Update Propagation Framework	212.762	144900	3.875	512	7.688	3351
LED: curated and crowdsourced Linked Data on Music Listening Experiences	244.409	138200	38.349	5100	18.126	2398.4
Lexvo.org: Language Information for the Linguistic Linked Data Cloud	327.826	150300	0.469	6.1	-	-
Property Path over Linked Data: Can It be Done and How to Start?	413.476	171100	0.485	13.8	30.721	70400
Semantic-enabled Framework for Spatial Image Information Mining of Linked Earth Observation Data	217.45	147900	5.454	526.6	7.126	8789.7
The Semantic Web Journal as Linked Data	0	0	0	0	19.517	14.6

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis lahir di Jombang pada tanggal 16 Februari 1995. Merupakan anak kedua dari 2 bersaudara dan telah menempuh pendidikan formal yaitu; SD Negeri Bendo 02 Blitar, SMP Negeri 1 Blitar, dan SMA Negeri 1 Blitar.

Pada tahun 2013 melanjutkan pendidikan di Jurusan Sistem Informasi FTIF - Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 5213100059. Selama menjadi mahasiswa penulis mengikuti kegiatan kemahasiswaan seperti kegiatan UKM, kepanitiaan di ITS serta aktif sebagai staff Kajian Islam Sistem Informasi (KISI) periode 2015/2016. Pada tahun keempat karena penulis tertarik dengan bidang desiminasi informasi dan linked data, maka penulis mengambil bidang minat Laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui email lutfinf@gmail.com.