

**PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-
KARP PADA SISTEM INFORMASI
PELAPORAN KERUSAKAN FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

Oleh:

AHMAD FARHAN

NIM. 09650098



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-
KARP PADA SISTEM INFORMASI
PELAPORAN KERUSAKAN FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

Oleh:

AHMAD FARHAN

NIM. 09650098



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-
KARP PADA SISTEM INFORMASI PELAPORAN KERUSAKAN
FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh :

**AHMAD FARHAN
NIM. 09650098**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG
2014**

HALAMAN PERSETUJUAN

**PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-KARP PADA SISTEM INFORMASI
PELAPORAN KERUSAKAN FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

Oleh :

**Nama : Ahmad Farhan
NIM : 09650098
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi**

Telah Disetujui, November 2014

Pembimbing I,

Pembimbing II,

A'la Syaqui, M.Kom
NIP.19771201 200801 1 007

Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-KARP PADA SISTEM INFORMASI
PELAPORAN KERUSAKAN FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM
MALANG**

SKRIPSI

**Oleh :
AHMAD FARHAN
NIM. 09650098**

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Informatika (S.Kom)
Tanggal: 21 November 2014

- | | | | |
|-----------------------|-----------------------------------|---|---|
| 1. Penguji Utama | : <u>Totok Chamidy, M.Kom</u> | (|) |
| | NIP. 19691222 200604 1 001 | | |
| 2. Ketua Penguji | : <u>Ririen Kusumawati, M.Kom</u> | (|) |
| | NIP. 19720309 200501 2 002 | | |
| 3. Sekretaris Penguji | : <u>A'la Syauqi, M.Kom</u> | (|) |
| | NIP.19771201 200801 1 007 | | |
| 4. Anggota Penguji | : <u>Dr. Muhammad Faisal, M.T</u> | (|) |
| | NIP.19740510 200501 1 007 | | |

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19749424 200901 1 008

HALAMAN PERSEMBAHAN



“Dia memberikan hikmah (ilmu yang berguna)

kepada siapa yang dikehendaki-Nya.
Barang siapa yang mendapat hikmah itu
Sesungguhnya ia telah mendapat kebajikan yang banyak.
Dan tiadalah yang menerima peringatan
melainkan orang-orang yang berakal”.
(Q.S. Al-Baqarah: 269)

Dengan mengucapkan Syukur Alhamdulillah kepad Allah SWT atas semua karunia Ilmu dan Ridho yang diberikan.

Sholawat serta salam semoga selalu terlimpahkan kepada Nabi Muhammad SAW.

Ungkapan hati sebagai rasa Terima Kasihku

Bapak Syafii dan Ibu Siti Fatimah orang tua terhebat yang selalu mengasihi dan menyanyagiku tanpa batas dalam setiap langkah hidupku.

Adikku tercinta Fitriatus Sholihah dan juga untuk Om Tek dan Bulek Dina yang selalu memotivasiku. serta seluruh keluarga besar tercinta yang tidak dapat disebutkan satu-persatu. Terimakasih atas segalanya.

Teman-teman seperjuangan D'TIC dan semua teman-teman jurusan TI 2009.

Teman-teman Bermain Habib, Dana, Irvan, Ainul dan Agus

Teman-teman Organisasi : HMJ TI 2009/2010/2011, DEMA F Saintek 2012, DEMA U 2013 PMII Galileo Angkatan 2009 “Panah 9”, PMII Komisaria SA Malang.

Teman-teman KK Mas Hamdan, Mas Obet, Mas Naseh, Mbah Jemblunk, pe2nk terimakasih atas semua bantuannya.

Teman-teman Excellent Course Pare Kediri Mr. Andi, Mr. Deddy, Mr. Muji.

Tidak ada kata selain terimakasih yang bisa penulis ucapkan atas semua dukungannya.

MOTTO

Life is not only thought
But also
Done.



PERNYATAAN ORISINALITAS PENELITIAN

Nama : AHMAD FARHAN
NIM : 09650098
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi
Judul Skripsi : **PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA
RABIN-KARP PADA SISTEM INFORMASI
PELAPORAN KERUSAKAN FASILITAS UMUM
DI UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG**

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 November 2014
Yang membuat pernyataan,

Ahmad Farhan
NIM. 09650098

KATA PENGANTAR



Segala puji bagi Allah SWT, karena atas rahmat, hidayah serta karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul “*Penerapan Semantic Search Dan Algoritma Rabin-Karp Pada Sistem Informasi Pelaporan Kerusakan Fasilitas Umum Di Universitas Islam Negeri Maulana Malik Ibrahim Malang*” dengan sebaik-baiknya sebagai salah satu syarat untuk memperoleh gelar sarjana pada program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Shalawat serta salam semoga senantiasa Allah limpahkan kepada Nabi Muhammad SAW, keluarga, sahabat dan ahlinya yang telah membimbing umat menuju kebahagiaan dunia dan akhirat.

Penulis menyadari adanya banyak keterbatasan yang penulis miliki, sehingga ada banyak pihak yang telah memberikan bantuan baik moril maupun materiil dalam menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis menyampaikan doa dan mengucapkan terima kasih kepada:

1. Prof. DR. H. Mudjia Rahardjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. Hj. Bayyinatul Muchtaromah., drh., M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. A’la Syauqi, M.Kom dan Dr.Muhammad Faisal, M.T. selaku dosen pembimbing I dan II, terimakasih atas bimbingan, masukan-masukan dan pemikiran untuk penyelesaian skripsi ini.
5. Segenap civitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas semua kesempatan untuk semua pembelajarannya.
6. Bapak, Ibu dan adikku tercinta serta seluruh keluarga besar yang senantiasa memberikan doa dan dukungannya kepada penulis dalam menuntut ilmu serta dalam menyelesaikan skripsi ini.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna, untuk itu penulis selalu menerima segala kritik dan saran dari pembaca. Harapan penulis, semoga karya ini bermanfaat bagi kita semua.

Malang, 17 November 2014

Penulis



DAFTAR ISI

1. HALAMAN JUDUL	i
2. HALAMAN PENGAJUAN.....	ii
3. HALAMAN PERSETUJUAN.....	iii
4. HALAMAN PENGESAHAN.....	iv
5. HALAMAN PERNYATAAN.....	v
6. HALAMAN MOTTO	vi
7. HALAMAN PERSEMBAHAN.....	vii
8. KATA PENGANTAR.....	viii
9. DAFTAR ISI.....	x
10. DAFTAR TABEL.....	xiii
11. DAFTAR GAMBAR.....	xiv
12. ABSTRAK.....	xvi
 13. BAB I PENDAHULUAN	
13.1 Latar Belakang	1
13.2 Rumusan Masalah	5
13.3 Tujuan Penelitian	5
13.4 Batasan Masalah	5
13.5 Manfaat Penelitian	6
13.6 Metode Penelitian	6
13.7 Sistematika Penulisan	8
 14. BAB II TINJAUAN PUSTAKA	
14.1 Pengertian Data	9
14.2 Pengertian Laporan	9
14.3 Semantic Web	11
14.3.1 Semantic	11
14.3.2 Web	13
14.3.3 Semantic Web	13
14.4 Algoritma Rabin-Karp.....	19
14.5 Protégé	21
14.6 Sesame	22

14.7	D2RQ Server	22
14.7.1	D2RQ Mapping Language	23
14.7.2	DataBase Conection	24
14.7.3	Membuat RDF Resources	24
14.8	JSON	25
14.9	MySQL	25
 15. BAB III ANALISIS DAN PERANCANGAN SISTEM		
15.1	Pemodelan Ontology	27
15.2	Perancangan Managemen Data Laporan.....	29
15.3	Perancangan Sistem	31
15.3.1	Desain Aplikasi	31
15.3.1.1	Context Diagram	31
15.3.1.2	Data Flow Diagram	33
15.3.1.3	DFD Level 2	35
15.3.1.4	Entity Relational Diagram	38
15.3.2	Desain Teknologi Semantic Web	43
15.3.2.1	Proses Pencarian Dengan Teknologi Semantic Web	44
15.3.2.2	Proses Pencocokan String Dengan Algoritma Rabin-Karp	48
15.4	Desain D2R Server	50
15.4.1	Mapping	50
15.4.2	RDF Skema	51
15.5	Desain Interface	58
15.5.1	Desain Interface civitas	58
15.5.2	Desain Interface Teknisi	58
15.5.3	Desain Interface Kepala bagian.....	60
15.5.4	Desain Interface Admin.....	62
15.5.5	Desain Interface User.....	65
15.6	Kebutuhanan Sistem	65
 16. BAB IV MANAJEMEN PELAKSANAAN KERJA		
16.1	Aplikasi managemen data laporan	68
16.1.1	Level Admin	68

16.1.2	Level ketua Bagian	69
16.1.3	Level Teknisi	70
16.1.4	Level Civitas	71
16.2	Aplikasi Pencarian	72
16.2.1	Membuat RDF Map dengan D2R server	73
16.2.2	Menjalankan D2RQ Sever	86
16.3	Proses Pencarian dengan Semantic Web	88
16.3.1	Penerapan Algoritma Rabin Karp	91
16.4	Ujicoba Sistem	93
16.5	Integrasi Dalam Islam	95
17. BAB V PENUTUP		
17.1	Kesimpulan	98
17.2	Saran	98
DAFTAR PUSTAKA		99

DAFTAR TABEL

Tabel 2.1 perbandingan antara semantic web dengan web biasa	15
Tabel 2.2 perbandingan knowledgebase dan relational database	16
Tabel 3.1 Tabel barang pada database kerusakan	40
Tabel 3.2 Tabel gedung pada database kerusakan	41
Tabel 3.3 Tabel kelompok pada database kerusakan	41
Tabel 3.4 Tabel kerusakan pada database kerusakan	41
Tabel 3.5 Tabel perbaikan pada database kerusakan	42
Tabel 3.6 Tabel petugas pada database kerusakan	42
Tabel 3.7 Tabel ruang pada database kerusakan	42
Tabel 3.8 Tabel satuan barang pada database kerusakan	43
Tabel 3.9 Tabel sub kelompok pada database kerusakan	43
Tabel 3.10 Tabel uraian pada database kerusakan	43
Tabel 3.10 Tabel user pada database kerusakan	44
Tabel 4.1 Percocokan nilai hash	94
Tabel 4.2 Hasil Ujicoba	95

DAFTAR GAMBAR

Gambar 2.1 Contoh relasi antara class	16
Gambar 2.2 Arsitektur Semantic Web	17
Gambar 2.3 Desain implementasi D2R Server	23
 Gambar 3.1 Rancangan Aplikasi	 28
Gambar 3.2 Contoh Hirarki is-a	29
Gambar 3.3 Alur proses pelaporan kerusakan fasilitas umum	30
Gambar 3.4 Context Diagram	33
Gambar 3.5 Data Flow Diagram.....	34
Gambar 3.6 DFD Level 2 Pada Admin	36
Gambar 3.7 DFD Level 2 Pada Teknisi	37
Gambar 3.8 DFD Level 2 Pada Ketua Kepala bagian	37
Gambar 3.9 DFD Level 2 Pada Civitas	38
Gambar 3.10 DFD Level 2 DFD Level 2 Pada User	38
Gambar 3.11 ERD Pada aplikasi Pelaporan Kerusakan Fasilitas Umum	38
Gambar 3.12 Desain aplikasi pencarian	44
Gambar 3.13 flowchart pencarian data menggunakan semantic web.....	46
Gambar 3.14 Flowchat pencarian menggunakan semantic web dan algoritma rabin- karp	48
Gambar 3.15 flowchat pencarian menggunakan Algoritma Rabin-Karp..	50
Gambar 3.16 Desain D2R Server	51
Gambar 3.17 RDF Skema pada tabel barang.....	52
Gambar 3.18 RDF Skema pada tabel satuan_barang	53
Gambar 3.19 RDF Skema pada tabel gedung	53
Gambar 3.20 RDF Skema pada tabel ruang	54
Gambar 3.21 RDF Skema pada tabel kelompok	54
Gambar 3.22 RDF Skema pada tabel sub_kelompok	55
Gambar 3.23 RDF Skema pada tabel petugas	55
Gambar 3.24 RDF Skema pada tabel kerusakan	55
Gambar 3.25 RDF Skema pada tabel perbaikan	57
Gambar 3.26 RDF Skema pada tabel uraian	58

Gambar 3.27 RDF Skema pada tabel user	58
Gambar 3.28 desain interface user civitas	59
Gambar 3.29 desain input uraian kerusakan	60
Gambar 3.30 desain perintah pengecekan	61
Gambar 3.31 desain interface untuk memeriksa laporan	62
Gambar 3.32 desain perintah pengecekan kerusakan	62
Gambar 3.33 desain detail pengecekan	63
Gambar 3.34 desain interface untuk manajemen data	64
Gambar 3.35 desain interface menambahkan data	64
Gambar 3.36 desain interface untuk merubah data.....	65
Gambar 3.37 desain interface menghapus data.....	65
Gambar 3.38 desain interface untuk pencarian data.....	66
Gambar 4.1 interface halaman admin.....	70
Gambar 4.2 interface laporan pada ketua bagian.....	71
Gambar 4.3 interface perintah perngecekan pada teknisi.....	72
Gambar 4.4 interface pada civitas akademika.....	73
Gambar 4.5 Proses pembuatan RDF Map dengan command	74
Gambar 4.6 Menjalankan RDF Map dengan command	87
Gambar 4.7 RDF Map ketika berhasil dijalankan.....	88
Gambar 4.8 Tampilan D2R Server pada browser.....	88
Gambar 4.9 Tampilan SPARQL Endpoint.....	89
Gambar 4.10 Tampilan pencarian data.....	90
Gambar 4.11 Tampilan hasil dari pencarian data	90
Gambar 4.12 Pseudocode algoritma rabin-karp.....	90

ABSTRAK

Farhan, Ahmad. 2014. PENERAPAN *SEMANTIC SEARCH* DAN ALGORITMA RABIN-KARP PADA SISTEM INFORMASI PELAPORAN KERUSAKAN FASILITAS UMUM, Jurusan Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) A'la Syauqi, M.Kom, (II) Dr. Muhammad Faisal, M.T

Kata kunci : Semantic Web, Rabin-Karp, D2R Server, Sparql Query, Turtle

Semantic web adalah perkembangan generasi web berikutnya atau yang bisa disebut sebagai evolusi dari WWW (World Wide Web), yang dicetuskan pada tahun 2002. Semantic web merujuk pada kemampuan aplikasi komputer untuk lebih memahami bahasa manusia, bukan hanya bahasa yang baku dari para penggunanya tetapi juga bahasa yang lebih kompleks. Pada penelitian ini penerapan semantic web menggunakan pendekatan *turtle*, dalam menjalankan semantic web pencarian data menggunakan perintah sparql query. Sparql query akan mengambil data melalui D2R server sebagai jembatan untuk bisa mengakses data dari MySql DBMS. Pada pencarian data digunakan juga Algoritma *Rabin-Karp*, pada dasarnya Algoritma *Rabin-Karp* menghitung nilai *hash* untuk *pattern* dan setiap *k-gram* dari teks yang akan dibandingkan. Jika nilai *hash* tidak sama, maka akan dihitung nilai *hash* untuk *k-gram* selanjutnya secara berurutan. Jika nilai *hash* sama, maka dilakukan perbandingan antara *pattern* dan *k-gram*. Kombinasi teknologi semantic web dengan algoritma rabin-karp diharapkan bisa memaksimalkan ketepatan hasil pencarian data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Hal ini bertujuan agar semua civitas akademika bisa mencari data laporan kerusakan fasilitas umum secara tepat dan benar.

ABSTRAK

Farhan. Ahmad. 2014. **APPLICATION OF SEMANTIC SEARCH AND RABIN-KARP ALGORITHM ON INFORMATION SYSTEM OF REPORTING GENERAL FACILITIES IN STATE ISLAMIC UNIVERSITY OF MAULANA MALIK IBRAHIM OF MALANG**, Information Technology. Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim of Malang.

Advisor: (I) A'la Syauqi, M.Kom, (II) Dr. Muhammad Faisal, M.T

Keywords: Semantic Web, Rabin-Karp, D2R Server, SPARQL Query, RDF, Turtle

The Semantic Web is the development of the next generation or the web can be called as the evolution of the WWW (World Wide Web), which was launched in 2002. The Semantic Web refers to the ability of a computer application to better understand human language, not just the standard language of its users but also more complex language. In this study the application of semantic web using turtle approach, in running the semantic web search data using SPARQL query command. SPARQL query will retrieve data through D2R server as a bridge to access the data from the MySQL DBMS. In search of data used Rabin-Karp algorithm also, basically Rabin-Karp algorithm calculates a hash value for the pattern and each k-gram of the text that will be compared. If the hash values are not the same, then it will be calculated hash value for the next k-gram respectively. If the same hash value, then do a comparison between the pattern and the k-gram. The combination of semantic web technologies with Rabin-Karp algorithm is expected to maximize the accuracy of the data reported results of existing damage to public facilities in the State Islamic University of Maulana Malik Ibrahim of Malang. It is intended that all communities can seek damages public facilities report data accurately and correctly.

الملخص

فرحان أحمد. 2014. تطبيق البحث الدلالي و خوارزمية راين كارب في نظم الإبلاغ المعلومات المنشئة العامة. قسم المعلوماتية كلية العلوم والتكنولوجيا. الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: أعلي شوقي الماجستير، الدكتور محمد فيصل الماجستير ،

الكلمات البحث: الويب الدلالي، راين كارب، خادم D2R، SPARQL query، السلاحف
الويب الدلالي هو تطوير الجيل القادم أو يمكن أن يطلق عليه الويب كما تطور WWW (الشبكة العالمية)، التي تم إطلاقها في عام 2002. ويشير الويب الدلالي لقدرة تطبيق الكمبيوتر للآلة لفهم اللغة الإنسان، وليس فقط اللغة القياسية لمستخدميها ولكن أيضا المزيد من اللغات المعقدة. في هذه الدراسة تطبيق الويب الدلالي باستخدام نهج السلحفاة، في إدارة بحث الويب الدلالي البيانات باستخدام القيادة الاستعلام sparql query . سوف الاستعلام SPARQL استرداد البيانات من خلال الخادم D2R كجسر للوصول إلى البيانات من الخلية DBMS . بحثا عن البيانات تستخدم خوارزمية راين كارب أيضا، وأساسا خوارزمية راين كارب بحساب قيمة تجزئة hash لنمط وكل ك-غرام من الكلمة الذي سيتم المقارنة. إذا كانت القيم التجزئة hash ليست هي نفسها، قيمة التجزئة hash ثم سيتم احتسابها التالي من ك-غرام على التوالي. إذا كانت قيمة البعثة نفسها، ثم القيام مقارنة بين نمط وك-غرام. ومن المتوقع تكنولوجيا الويب الدلالي ب خوارزمية راين كارب يحتاج ان نتائج البحث البيانات الإبلاغ المعلومات المنشئة العامة في الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. الغرض منها هي أن كل المجتمع الأكاديمي يمكن أن ننظر البيانات الإبلاغ المعلومات المنشئة العامة بدقة وصحيحة.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Hingga saat ini belum ada sistem yang bisa digunakan oleh civitas akademika untuk bisa mengakses data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Semua laporan kerusakan fasilitas umum yang masuk pada kepala bagian umum berupa kertas laporan, baik berupa *form* pelaporan resmi yang bisa diperoleh dari bagian umum ataupun kertas biasa yang terdapat catatan kerusakan fasilitas umum. Semua administrasi pelaporan kerusakan fasilitas umum masih dilakukan dengan cara manual, yaitu setiap civitas akademika yang ingin melaporkan kerusakan fasilitas umum bisa menggunakan *form* resmi yang dikeluarkan oleh bagian umum ataupun bisa dengan kertas biasa bahkan bisa juga dengan mengirimkan sms kepala pegawai lapangan ataupun langsung kepada kepala bagian umum. Semua data pelaporan tidak disimpan pada sistem tapi masih menggunakan aplikasi microsoft office excel. Sehingga semua data laporan tidak bisa dilihat atau diakses oleh civitas akademika karena belum adanya aplikasi *sharing* data agar semua laporan kerusakan bisa dilihat oleh semua civitas akademika.

Hal inilah yang mendorong peneliti untuk membuat aplikasi pencarian yang memanfaatkan teknologi semantic web sebagai media *sharing* data dan algoritma rabin-karp untuk menunjang ketepatan dalam pencarian data laporan kerusakan

fasilitas umum dengan kata kunci penting yang ada pada data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Semantic web adalah perkembangan generasi web berikutnya atau yang bisa disebut sebagai evolusi dari WWW (World Wide Web), yang dicetuskan pada tahun 2002. Semantic web merujuk pada kemampuan aplikasi komputer untuk lebih memahami bahasa manusia, bukan hanya bahasa yang baku dari para penggunanya tetapi juga bahasa yang lebih kompleks, seperti dalam bahasa percakapan sehingga memudahkan penggunanya untuk berkomunikasi dengan mesin. Semantic web juga dapat mengolah bahasa dan mengenali homonim, sinonim, atau atribut yang berbeda pada suatu database.

Algoritma *Rabin-Karp* merupakan versi awal dari *fingerprinting* dengan menggunakan metode *k-gram* yang diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 (Schleimer, Wilkerson, & Aiken, 2003). Pada dasarnya, Algoritma *Rabin Karp* menghitung nilai *hash* untuk *pattern* dan setiap *k-gram* dari teks yang akan dibandingkan. Jika nilai *hash* tidak sama, maka akan dihitung nilai *hash* untuk *k-gram* selanjutnya secara berurutan. Jika nilai *hash* sama, maka dilakukan perbandingan antara *pattern* dan *k-gram*.

Kombinasi teknologi semantic web dengan algoritma rabin-karp diharapkan bisa memaksimalkan ketepatan hasil pencarian data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Hal ini bertujuan agar semua civitas akademik bisa mencari data laporan kerusakan

fasilitas umum secara tepat dan benar. Allah sudah memerintahkan untuk selektif ketika menerima informasi yang dijelaskna Al-quran surat Al-Hujurat ayat 6:

يٰۤاَيُّهَا الَّذِيْنَ ءَامَنُوْا اِنْ جَاءَكُمْ فَاسِقٌ بِّنَبَاٍ فَتَبَيَّنُوْا اَنْ تُصِيبُوْا
 قَوْمًا بِجَهْلَةٍ فَتُصْبِحُوْا عَلٰٓى مَا فَعَلْتُمْ نٰدِمِيْنَ ﴿٦﴾

Artinya: Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah (kebenarannya) dengan teliti, agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu. (Al-Hujurat: 6)

Ayat ini turun, memberikan penjelasan bagi umat manusia semuanya untuk selalu tabayun dalam segala berita yang disampaikan oleh orang muslim maupun nonmuslim. Kemudian ayat ini menyuruh kita berhati-hati dalam menindakkan sesuatu yang akibatnya tidak dapat diperbaiki (perkataannya banyak menimbulkan kerusakan), supaya tidak ada pihak atau kaum yang dirugikan, dtimpa musibah atau bencana ysng disebabkan berita yang belum pasti kebenarannya, sehingga menyebabkan penyesalan yang terjadi (Ibnu Katsir, 7:316). Ayat ini menolak berita orang-orang fasiq dan mensyaratkan keadilan, baik dia perawi ataupun saksi, dan membolehkan kita menerima khabar seorang yang adil (Tafsir Bayaan, 4:1363). Secara historis, bahwa yang melakukan perbuatan fasiq dalam ayat tersebut adalah orang muslim (M. Quraish Shihab, 13: 237), sehingga tidak ada jaminan bahwa jika seseorang telah memeluk agama islam telah berlaku baik dalam segala aspek.

Memahami tafsir dari surat Al-Hujurat ayat 6, bisa ditarik kesimpulan bahwa kebenaran informasi itu sangat penting. Karena setiap Informasi yang diberikan kepada orang lain harus dipertanggungjawabkan. Kebenaran informasi data kerusakan fasilitas umum Universitas Islam Negeri Maulana Malik Ibrahim Malang yang bisa diakses oleh semua civitas akademika harus benar, karena data laporan kerusakan fasilitas umum mempunyai hubungan dengan bagian keuangan dan bagian pengadaan yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang yang setiap akhir tahun harus dilaporkan kepada bagian keuangan dan bagian pengadaan untuk dipertanggung jawabkan kebenarannya.

Dari serangkaian permasalahan diatas sehingga peneliti membuat sebuah aplikasi sederhana yang mana dirancang khusus agar penerapan dalam pencarian data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang dapat dilakukan dengan tepat dan spesifik. Adanya teknologi semantic web dapat mendukung proses pencarian tersebut. Dalam penelitian sebelumnya untuk menemukan objek dengan kata kunci tertentu, memerlukan pilihan/kategori terlebih dahulu agar hasil pencarian dapat lebih spesifik. Dengan menggunakan teknologi semantic web dan algoritma rabin-karp untuk pencarian bisa dilakukan dengan menginputkan langsung kata kunci tentang data laporan kerusakan fasilitas umum yang akan dicari tanpa menggunakan kategori/piliha tertentu dan proses pencarian pun akan menjadi lebih mudah. Diharapkan dalam aplikasi ini, nantinya akan membantu civitas akademika dalam mencari data laporan kerusakan fasilitas umum yang ada di Universitas islam negeri

maulana malik ibrahin malang dengan lebih spesifik dan akurat sehingga sesuai dengan yang dikehendaki.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, Rumusan masalah pada penelitian ini adalah Bagaimana merancang dan membangun aplikasi pencarian data laporan kerusakan fasilitas umum di Universitas Islam Negeri Maulana Malik Ibrahim Malang dengan menggunakan teknologi Semantic web dan algoritma *Rabin-Karp* untuk optimasi pencarian?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah membuat aplikasi pencarian data laporan kerusakan fasilitas umum menggunakan teknologi semantic web dan algoritma *Rabin-Karp* di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data Kerusakan Fasilitas Umum sesuai dengan data yang diperoleh dari bagian umum UIN Maliki Malang.
2. Aplikasi berbasis web dan menggunakan teknologi *Semantic Web*.
3. Dan aplikasi ini menggunakan algoritma *Rabin-Karp* untuk pencariannya.

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi ini adalah :

1. Sebagai fasilitas *sharing* data laporan kerusakan fasilitas umum di Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Sebagai media pencarian data laporan kerusakan fasilitas umum di Universitas Islam Negeri Maulana Malik Ibrahim Malang

1.6 Metodologi Penelitian

Dalam melakukan penelitian ini, peneliti mengacu ada konsep penelitian menggunakan metode water fall yang digunakan sebagai acuan dalam penyelesaian penelitian tersebut. Berikut beberapa tahapan water fall dalam melakukan penelitian:

1. Studi literatur

Pada tahapan ini dilakukan pengumpulan informasi terkait beberapa hal berikut:

- a. Pengumpulan informasi tentang Bagian Umum UIN Maliki Malang khususnya pada bagian Kerumahtanggaan.
- b. Pengumpulan data manual terkait proses pelaporan kerusakan fasilitas umum secara manual yang dilakukan oleh pihak bagian umum.

- c. Pengumpulan informasi tentang bagaimana cara membangun sebuah Sistem Informasi Pelaporan Kerusakan Fasilitas Umum menggunakan semantic web.

2. Perancangan dan desain aplikasi

Perancangan aplikasi terdiri atas pembuatan alur sistem, blok diagram, perancangan database, perancangan user dan proses-prosesnya dan desain aplikasi yang terdiri atas desain menu pencarian fasilitas umum dan desain utama dari aplikasi itu sendiri.

3. Pembuatan aplikasi

Setelah perancangan selesai, maka akan diimplementasikan pada pemrograman berbasis web dengan menggunakan tool-tool sebagai pendukung dalam proses pembuatan sistem informasi pelaporan kerusakan fasilitas umum dan semantic web.

4. Uji Coba dan Evaluasi

Uji coba dan evaluasi dilakukan terhadap tahapan aplikasi dan hasil dari pencarian semantik serta akurasi dan perolehan pencarian string yang ditemukan dengan menggunakan algoritma *Rabin-Karp* dengan pencarian yang tidak menggunakan algoritma tersebut.

5. Penyusunan Laporan

Penyusunan laporan akhir merupakan dokumentasi dari keseluruhan pelaksanaan penelitian dan diharapkan bermanfaat bagi penelitian lebih lanjut.

1.7 Sistematika Penulisan

Penulisan laporan tugas akhir ini terdiri dari 5 (lima) bab dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Pada Pendahuluan Membahas tentang latar belakang, rumusan masalah, tujuan masalah, batasan masalah, manfaat masalah, metodologi penelitian dan sistematika penulisan laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Tinjauan pustaka berisi teori-teori yang mendasari pembuatan sistem pada tugas akhir ini. Adapun yang dibahas pada bab ini adalah dasar teori yang berkaitan dengan pembahasan pelaporan kerusakan fasilitas umum, teori pembuatan *Semantic Web* dan teori tentang algoritma *Rabin-Karp*.

BAB III ANALIS DAN PERANCANGAN

Pada bab III membahas tentang Analisis dan perancangan sistem pada pembuatan Sistem Informasi Pelaporan Kerusakan Fasilitas Umum menggunakan teknologi *Semantic Web* dan algoritma *Rabin-Karp* meliputi sistematik perancangan sistem dan pembuatan aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan tentang implementasi, hasil pengujian sistem dan pengujian penggunaan aplikasi.

BAB V PENUTUP

Berisi kesimpulan dan saran

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Data

Data merupakan keterangan-keterangan atau fakta-fakta yang dikumpulkandari suatu populasi atau bagian populasi yang akan digunakan untuk menerangkanciri-ciri populasi yang bersangkutan (Lungan, 2006: 13). Menurut Hasan (2004:16) data merupakan keterangan-keterangan tentang suatu hal, dapat berupasesuatu yang diketahui atau dianggap. Dari pendapat para ahli tersebut, dapatdisimpulkan bahwa data merupakan fakta atau keterangan yang dikumpulkan darisuatu populasi untuk menjelaskan karakteristik populasi tersebut. Agar data dapat menerangkan ciri-ciri populasi dengan benar, maka menurut Lungan (2006: 14) data tersebut harus memenuhi kriteria sebagai berikut.

- a. Objektif. Data yang bersifat objektif ialah data yang benar-benar samadengan keadaan yang sebenarnya (apa adanya)
- b. Mewakili populasi
- c. Galat baku (standard error) kecil
- d. Tepat waktu
- e. Relevan

2.2 Pengertian Laporan

Suatu bentuk penyampaian berita, keterangan, pemberitahuan ataupun pertanggungjawaban baik secara lisan maupun secara tertulis dari bawahan

kepada atasan sesuai dengan hubungan wewenang dan tanggung jawab yang diorganisir dalam sebuah narasi, grafik, atau bentuk tabel, disusun berdasarkan ad hoc, periodik, berulang, teratur, atau sebagai dasar yang diperlukan. Laporan bisa merujuk ke periode tertentu, peristiwa, kejadian, atau subjek, dan dapat dikomunikasikan atau disajikan dalam bentuk lisan atau tertulis.

Laporan mempunyai peranan yang penting pada suatu organisasi karena dalam suatu organisasi dimana hubungan antara atasan dan bawahan merupakan bagian dari keberhasilan organisasi tersebut. Dengan adanya hubungan antara perseorangan dalam suatu organisasi baik yang berupa hubungan antara atasan dan bawahan, ataupun antara sesama karyawan yang terjalin baik maka akan bisa mewujudkan suatu sistem delegation of authority dan pertanggungjawaban akan terlaksana secara efektif dan efisien. Kerja sama diantara atasan bawahan bisa dilakukan, dibina melalui komunikasi baik komunikasi yang berbentuk lisan maupun tulisan (laporan). Agar laporan tersebut bisa efektif mempunyai syarat-syarat yang perlu dipenuhi demi terbentuknya laporan yang baik maka seseorang perlu mengetahui secara baik bagaimana pembuatan format laporan yang sempurna. Sehingga dengan laporan yang terformat bagus akan bisa bermanfaat baik dalam komunikasi maupun dalam mencapai tujuan organisasi.

Dalam Al-quran dijelaskan juga perintah manajemen laporan, hal ini sebagai bentuk tanggung jawab orang yang diberi kepercayaan kepada orang yang memberi kepercayaan.

يَا أَيُّهَا الَّذِينَ ءَامَنُوا لَا تَخُونُوا اللَّهَ وَالرَّسُولَ وَتَخُونُوا أَمَانَاتِكُمْ
وَأَنْتُمْ تَعْلَمُونَ ﴿٢٧﴾

Artinya : Wahai orang-orang yang beriman, janganlah kalian mengkhianati Allah dan Rasul (Muhammad) dan janganlah kalian mengkhianati amanah-amanah yang dipercayakan kepada kalian, sedang kalian dalam keadaan mengetahui. (Al-Anfal: 27)

Arti definisi/pengertian fasilitas umum adalah fasilitas yang diadakan untuk kepentingan umum. Contoh dari fasilitas umum adalah seperti jalan, angkutan umum, saluran air, jembatan, fly over, under pass, halte, alat penerangan umum, jaringan listrik, banjir kanal, trotoar, jalur busway, tempat pembuangan sampah, dan lain sebagainya

2.3 Semantic Web

2.3.1 Semantic

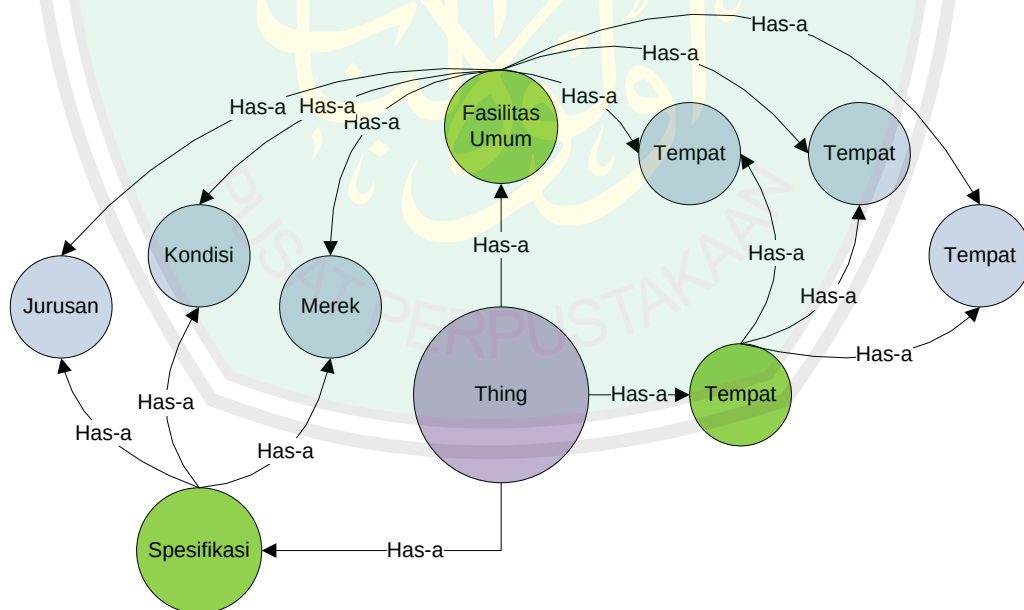
Semantik berasal dari bahasa Yunani semantikos yang artinya adalah cabang linguistik yang mempelajari tentang makna yang terkandung pada suatu bahasa, kode atau jenis representasi lain. Dengan kata lain semantik adalah pembelajaran tentang makna.

Semantik didalam bahasa Indonesia berasal dari bahasa Inggris “semantics” yang istilah tersebut digunakan para pakar untuk menyebut bagian ilmu bahasa yang mempelajari makna. Selanjutnya menurut (Saeed, 2003 :3), “semantics is the study of meaning communicated of through language “, yang berarti semantik adalah ilmu yang mempelajari makna yang

dikomunikasikan melalui bahasa. Jadi teori semantik adalah teori yang mempelajari ilmu tentang makna yang dikomunikasikan melalui bahasa.

Semantik merupakan bagian dari linguistik yang membahas tentang makna kata, frase dan klausa dalam suatu kalimat. Peranan semantik sangat penting dalam kehidupan berkomunikasi, karena bahasa yang digunakan dalam komunikasi tiada lain hanya untuk menyampaikan suatu makna.

Pada dasarnya semantic web merupakan penjelasan mendetail dari kata dan hubungannya dengan kata yang lain sehingga akan membentuk sebuah kosa kata yang difahami. Berikut adalah contoh model hubunga antar kata Fasilitas Umum, Spesifikasi dan Tempat.



Gambar 2.1, contoh relasi antara class satu dengan class yang lain

Pada gambar di atas dijelaskan web data semantik bersumber pada “Thing” thing mempunyai class Fasilitas Umum, spesifikasi dan tempat. Pada class

Fasilitas umum tidak mempunyai *sub class*, pada class spesifikasi mempunyai *sub class* (jurusan, kondisi dan merek), pada class tempat mempunyai *sub class* (gedung a, gedung b, gedung c). pada gambar diatas bisa dilihat relasi antar class dengan class yang lainnya, berarti tempat mengisikan kebutuhan atau penjelasan dari suatu kata satu persatu, kita bisa menggunakan metode ini dengan merelasikan hubungan antara class satu dengan class yang lainnya. Semantic web mempunyai struktur class dan sub class sebagai pembagian dari class. Web data semantik juga mempunyai has-a yang menjelaskan komponen-komponen yang terdapat pada class. Hal ini memungkinkan web data semantik menampilkan data atau pencarian data dengan lengkap dan detail sesuai struktur dan komponen bahasa.

2.3.2 Web

Sedangkan web sendiri yaitu bagian visual dari internet. Web didasarkan pada prinsip-prinsip Web page multimedia yang bentuknya seperti halaman-halaman majalah yang dilengkapi dengan gambar, teks, dan foto. Pembuat web, *European Particle Physics Laboratory* (CERN) mengembangkan bahasa komputer yang disebut dengan Hypertext Markup Language atau HTML, yang mana menjadi dasar dari web (Wayne Ause: 1995).

2.3.3 Semantic web

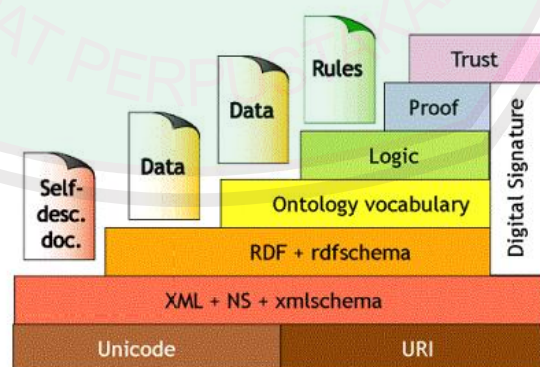
Semantic web adalah sebuah web data yang dijelaskan dan dihubungkan dengan cara-cara tertentu sesuai dengan tata bahasa dan konstruksi bahasa, untuk membangun konteks atau semantik yang memungkinkan computer memahami arti

dari sebuah informasi berdasarkan metadata, yaitu informasi mengenai isi informasi. Dengan adanya metadata, komputer diharapkan mampu mengartikan hasil pemasukan informasi sehingga hasil pencarian menjadi lebih detail dan tepat.

Semantic Web didefinisikan sebagai sekumpulan teknologi dimana memungkinkan komputer memahami arti dari sebuah informasi berdasarkan metadata, yaitu informasi mengenai isi informasi. Dengan adanya metadata, komputer diharapkan mampu mengartikan hasil pemasukan informasi sehingga hasil pencarian menjadi lebih detail dan tepat.

Format metadata tersebut didefinisikan oleh W3C sebagai Resource Description Format (RDF) yang terdiri dari 3 komposisi: subject, predicate, dan object. Hal menarik dari RDF adalah object dapat dijadikan subject yang nantinya diterangkan oleh object lainnya sehingga dapat dipaparkan secara lebih detail.

Arsitektur web semantik terdiri dari lapisan berikut:



Gambar 2.2 Arsitektur Semantic Web (sumber: Tim Berners-Lee)

✚ Lapisan Identifikasi

Lapisan ini terdiri dari dokumen yang ditulis dalam Unicode dan Uniform Resource Identifier (URI) terkait. Bisa menjadi URL atau URN (Uniform Resource Name).

✚ Lapisan Dokumen

Lapisan ini terdiri dari dokumen mark-up dalam XML dengan skema dan name space terkait.

✚ Lapisan Pernyataan

Lapisan ini terdiri dari RDF, juga dikenal sebagai pernyataan.

✚ Skema dan Lapisan Ontologi

Lapisan ini terdiri dari RDF-Schemas (yang mewakili taksonomi) dan ontologi. Mereka dapat direpresentasikan dalam RDF-Schema dan format bahasa OWL.

✚ Logika dan Proofs

Lapisan ini memberikan kemampuan untuk melakukan logika pada pernyataan semantik seperti kesimpulan. Proofs lebih sulit karena harus menarik beberapa pernyataan menjadi kesimpulan.

✚ Trust

Web semantik didasarkan pada internet. Oleh karena itu, tingkat kepercayaan pernyataan dan pengetahuan harus ditentukan melalui

sumber fakta yang dapat dipercaya. Tingkat trust (atau distrust) perlu diperhitungkan dalam agen dan mesin pencari yang menggunakan web semantik. (Berners-Lee, 2000)

Semantic Web merupakan perluasan dari World Wide Web yang memungkinkan orang untuk berbagi konten melampaui batas-batas suatu aplikasi dan website. Semantic web merupakan web data yang memungkinkan mesin untuk memahami suatu informasi pada World Wide Web sesuai maknanya atau secara struktural. Ini dapat memperluas jaringan hyperlink suatu halaman web dengan memasukkan mesin pembaca metadata mengenai halaman yang terdapat dalam web tersebut dan bagaimana mereka saling berhubungan satu sama lain. Hal tersebut memungkinkan suatu agen untuk otomatis mengakses web secara lebih cerdas dan melakukan tugas-tugas atas nama pengguna (Berners-Lee dkk, 2001). Semantic web berbeda apabila digambarkan seperti web pada umumnya. Web data semantik (semantic web) mempunyai kelebihan dibandingkan web data biasa www berikut adalah tabel perbandingan antara semantic web dengan web data biasa :

Tabel 2.1 perbandingan antara semantic web dengan web biasa

FAETURE	WWW	SEMANTIC WEB
<i>Fundamental component</i>	<i>Unstructured content</i>	<i>Formal statement</i>
<i>Primary audience</i>	<i>Humans</i>	<i>Aplication</i>
<i>Link</i>	<i>Indicate location</i>	<i>Indicate location and meaning</i>
<i>Primary vocabulary</i>	<i>Formating instractions</i>	<i>Semantic and logic</i>
<i>Logic</i>	<i>Informal / nonstandart</i>	<i>Description logic</i>

Dari tabel 2.1 perbandingan semantic web lebih kompleks dan mudah difahami dibandingkan dengan web data biasa, karna semantic web mengandung konten data yang saling berhubungan dan keterkaitan data yang sudah disusun dengan konsep semantik yang nantinya tersusun menjadi sebuah statemen sesuai dengan struktur bahasa.

Untuk membentuk statemen semantic web menggunakan kosa kata dari semantic web dan struktur bahasa yang sudah dibuat untuk mengidentifikasi hubungan dan keterkaitan kata kunci semantik web dengan atribut-atribut yang dimiliki oleh kata kunci tersebut. Untuk mendapatkan sebuah statemen, semantic web mengambil dari dua unsur yaitu knowledgebase dan relational database, berikut ini tabel perbandingan knowledgebase dan relational database:

Tabel 2.2 perbandingan knowledgebase dan relational database

FAETURE	Relational Database	Knowledgebase
<i>Structure</i>	<i>Schema</i>	<i>Ontology statements</i>
<i>Data</i>	<i>Rows</i>	<i>Instance Statemen</i>
<i>Administration Language</i>	<i>Data Description Language</i>	<i>Ontology Statemen</i>
<i>Query Language</i>	<i>Foriegn key</i>	<i>Multidimentional</i>
<i>Logic</i>	<i>External of Database</i>	<i>Formal logic statements</i>
<i>Uniqueness</i>	<i>Key for table</i>	<i>URL</i>

Keterangan dari tabel 2.2 Relational Database bergantung pada satu skema struktur hubungan yaitu *foreign key*, dimana *foreign key* digunakan sebagai skema relasi. Hubungan atau skema relasi *Knowledgebase* menawarkan hubungan

multidimensional seperti bagian dari, berhubungan dengan, dan banyak jenis lainnya

Standar yang paling penting dalam membangun *Semantic Web* adalah:

1. XML dan XML Schema

Extensible Markup Language (XML) merupakan bahasa yang didesain untuk mempermudah pengiriman dokumen melalui Web. Berbeda dengan *Hypertext Markup Language* (HTML), XML memungkinkan penggunaanya untuk mendefinisikan *custom tag* sendiri. Sedangkan *XML Schema* merupakan bahasa yang digunakan dalam pendefinisian sekumpulan skema atau aturan yang harus dipatuhi oleh dokumen XML.

2. RDF dan RDF Schema

Resource Description Framework (RDF) merupakan metode umum untuk memodelkan informasi yang dibuat oleh W3C dengan ide dasar bagaimana kita dapat membuat pernyataan mengenai sebuah *resource* Web dalam bentuk ekspresi “Subjet-Predikat-Objek” (yang selanjutnya disebut dengan istilah *N-Triple*). Subjek disini mengacu pada *resource* yang ingin dideksripsikan, sedangkan predikat menggambarkan kelakuan atau karakteristik dari *resource* tersebut dan mengekspresikan hubungan antara subjek dengan objek.

Pendeskripsian *resource-resource* inilah yang dikemukakan W3C sebagai komponen utama untuk perangkat lunak dalam menyimpan, menukar, dan menggunakan informasi sehingga dapat dibaca oleh mesin yang selanjutnya didistribusikan melalui Web.

RDF Schema merupakan kamus data atau *vocabulary* yang digunakan untuk mendeskripsikan *properties* dan *classes* dari *resources* RDF.

3. OWL

Web Ontology Language (OWL) merupakan suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi pada manusia, melainkan juga yang perlu memproses isi informasi. *Ontology* dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berupa *classes*, *properties*, dan *instances*. Dari deskripsi inilah sistem komputer dapat mengelola istilahistilah tersebut dengan cara yang lebih mudah (Lee, 2006).

4. SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) merupakan sebuah *protocol* dan bahasa *query* untuk *Semantic Web's resources*. *Query* SPARQL ini dapat terdiri dari *triple patterns*, konjungsi (*or*), dan disjungsi (*and*). Beberapa *tool* dan API yang dapat digunakan untuk menjalankan SPARQL adalah ARQ, Rasqal, RDF::Query, twingql, Pellet, dan KAON2. (Leigh, 2005)

Menurut Lintang (2012), pengertian lain *sparql* adalah *query* untuk RDF/OWL, *query* ini digunakan untuk mengambil data yang ditulis dengan menggunakan RDF/OWL atau XML.

2.4 Algoritma Rabin-Karp

Algoritma *Rabin-Karp* merupakan versi awal dari *fingerprinting* dengan menggunakan metode *k-gram* yang diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 (Schleimer, Wilkerson, & Aiken, 2003: 3). Pada

dasarnya, Algoritma *Rabin Karp* menghitung nilai *hash* untuk *pattern* dan setiap *k-gram* dari teks yang akan dibandingkan. Jika nilai *hash* tidak sama, maka akan dihitung nilai *hash* untuk *k-gram* selanjutnya secara berurutan. Jika nilai *hash* sama, maka dilakukan perbandingan antara *pattern* dan *k-gram*. Dengan cara ini, hanya ada satu perbandingan dari *subsequence* teks, dan *Brute Force* hanya diperlukan ketika nilai *hash* sama.

Karakteristik Algoritma *Rabin-Karp* : (Fernando, 2009: 2)

- a) Menggunakan sebuah fungsi *hashing*
- b) Fase *preprocessing* menggunakan kompleksitas waktu $O(m)$
- c) Untuk fase pencarian kompleksitasnya : $O(mn)$
- d) Waktu yang diperlukan $O(n+m)$

Fungsi *hashing* mengurangi perbandingan dari setiap *k-gram* yang kuadrat di dalam banyak kasus atau situasi. Daripada melakukan pemeriksaan dari *k-gram* di setiap posisi dari teks ketika terjadi pencocokan *pattern*, akan lebih baik dan efisien untuk melakukan pemeriksaan dari *k-gram* hanya jika teks yang sedang proses memiliki kemiripan nilai *hash* seperti pada *pattern*. Untuk melakukan pengecekan kemiripan antara dua kata ini digunakan fungsi *hash* (Fernando, 2009, hal. 2). Sedangkan perhitungan nilai *hash* selanjutnya dilakukan *Rolling hash*.

Berikut ini merupakan langkah-langkah dalam mencocokkan kata dengan algoritma *Rabin-Karp* (Steven, 2009: 15) :

1. Hilangkan tanda baca dan ubah teks sumber menjadi kata-kata tanpa huruf kapital.
2. Tentukan panjang dari teks sumber yang ingin dicari (n) dan kata yang ingin dicari (m)
3. Mencari nilai *hash* dari teks sumber dan kata yang ingin dicari menggunakan fungsi *hash* yang telah ditentukan
4. Lakukan iterasi dari indeks $i=0$ sampai $i=n-m+1$ untuk membandingkan nilai *hash* dari kata yang ingin dicari dengan nilai *hash* dari teks sumber pada indeks i sampai dengan $i+m-1$. Jika nilai *hash* sama, maka akan diperiksa lebih lanjut apakah kata yang dicari sama dengan bagian teks dari sumber pada indeks i sampai dengan $i+m-1$. Jika sama, maka telah ditemukan kata yang cocok. Jika tidak maka dilanjutkan dengan membandingkan nilai *hash* dari kata yang dicari dengan nilai *hash* teks sumber pada indeks berikutnya.

2.5 Protégé

Protege adalah perangkat lunak open source yang digunakan untuk memodelkan *basic knowledge* sebagai representasi sebuah pengetahuan yang dibuat dalam bentuk *ontology*. Pemodelan *ontology* dari sebuah representasi pengetahuan bisa dilakukan dengan menggunakan *protege* untuk mempermudah pembuatan desain pengetahuan yang akan diimplementasikan pada sebuah program. *Protege* menjadi *tool* pembuatan *ontology* yang sangat populer dikalangan developers program, karna *protege* memiliki fitur yang. Protégé

dibangun menggunakan bahasa pemrograman Java dengan fitur-fitur yang dapat digunakan melalui Graphical User Interface (GUI) dalam bentuk Tab. Dalam pendefinisian class dan hirarki class, property dan nilai-nilai dari property tersebut, relasi antar class, properti-properti dari relasi tersebut, penempatan SPARQL *query* (Horridge, 2011) .

2.6 Sesame

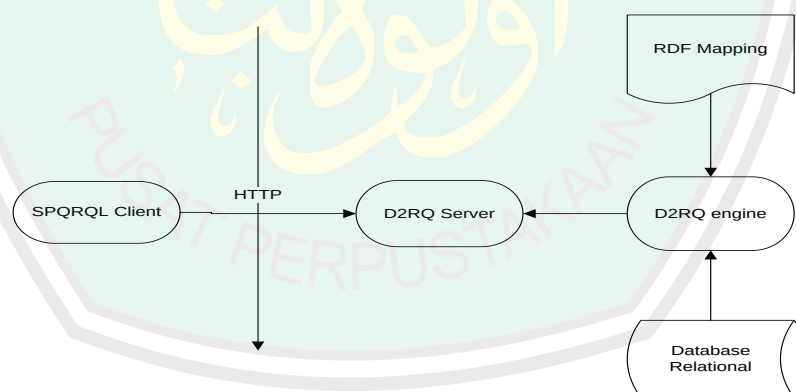
Sesame atau openRDF workbench adalah *framework* standar yang digunakan untuk mengelola data RDF. Sesame dilengkapi dengan fasilitas untuk mem-*parsing* data RDF, menyimpan data RDF dan fasilitas *query* yang digunakan untuk memeriksa *sparql query* untuk data RDF. Sesame dikembangkan oleh Aduna software sebagai proyek *open source* dengan lisensi *BSD-style License*. Sampai saat ini versi sesame terbaru adalah versi 2.7.11 yang dirilis pada tanggal 27 maret 2014. Sesame didesain sangat fleksibel yang mendukung bahasa *query sparql* dan menyediakan dokumentasi yang sangat berlimpah bagi para penggunanya. Dokumentasi yang disediakan meliputi tutorial, manual pengguna, *Frequently Ask Question* (FAQ) dan API. RDF storage program ini memberikan fasilitas *mailing-list* bagi pengguna dan *issue tracker* untuk menerima laporan *bug* dari pengguna. Sehingga sesame bisa digunakan oleh perorangan untuk membangun atau menyimpan data secara kecil dan juga bisa dipakai untuk menyimpan data secara besar.

Sesame menggunakan pustaka pendukung seperti RDF I/O (RIO) RDF *Parser*, *Sprin* dan Log4j. Sesame dikembangkan dengan menggunakan bahasa

pemrograman java. Penggunaan bahasa ini membuat sesame dapat dijalankan diatas semua *System Operasi* (Rakhmad, 2008)

2.7 D2R Server

D2R server adalah *tool* untuk mempublikasika konten dari *relational database* dalam Semantic web. Konten database dipetakan dalam RDF dengan mapping yang dideklarasikan untuk menspesifikasikan bagaimana *resources* diidentifikasi dan bagaimana property value digenerate dari konten database. D2R server mengizinkan RDF dan HTML *browser* untuk mengarahkan konten dari non-RDF *database* dan membolehkan aplikasi untuk melakukan *query* pada database menggunakan *sparql query* dengan SPARQL *protocol*. (Bizer, dkk., 2006)



Gambar 2.3, desain implementasi D2R Server

Data dalam *semantic web* dimodelkan dan digambarkan dalam RDF. D2R Server menggunakan D2RQ *mapping* untuk memetakan konten *database* kedalam format RDF, dan mengijinkan data RDF untuk melakukan *Searching* yang merupakan dua akses utama paradigma menuju semantic web.

2.7.1 D2RQ Mapping Language

D2RQ *language* merupakan bahasa yang dideklarasikan untuk mendeskripsikan relasi antara *relational database schema* dan RDFS *vocabularies* atau OWL *ontologies*. D2RQ merupakan *mapping* merupakan dokumen RDF yang disimpan dalam bentuk *Turtle*. Mapping diekspresikan sesuai dengan term D2RQ *namespace*: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0,1#>. *Mapping* menjelaskan *virtual RDF Graph* yang terdiri atas informasi dari database. (Cyganiak, dkk, 2012).

2.7.2 Database Connection

Mendefinisikan koneksi JDBC ke lokal atau *remote database relational*. D2rq: Database menggunakan propertinya dalam melakukan koneksi ke database. Selain menggunakan property yang dimiliki sendiri, d2rq:Database melakukan koneksi ke *database* juga dapat disukung dengan menggunakan *jdbc:namespace* (*namespace* URI: <http://d2rq.org/terms/jdbc/>) (Cyganiak, dkk, 2012). Contoh penulisan code untuk koneksi *database* MySQL adalah sebagai berikut:

```
map:database a d2rq:Database;
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:jdbcDSN "jdbc:mysql://localhost/kerusakan";
    d2rq:username "root";
    jdbc:autoReconnect "true";
    jdbc:zeroDateTimeBehavior "convertToNull";
```

2.7.3 Membuat RDF Resources

D2rq:ClassMap menyatakan *class* atau kelompok *class* yang serupa dari *OWL ontology* atau *RDFS schema*. Sebuah *ClassMap* mendefinisikan bagaimana instances dari *class* diidentifikasi. D2rq:ClassMap terhubung dengan d2rq:Database dan memiliki kelompok d2rq:PropertyBridges yang melampirkan *property* yang melampirkan *property* ke *instances*. D2rq:ClassMap merupakan *instances* dari RDFS atau OWL *class*. D2RQ secara otomatis membuat *rdf:type* triples bila dibutuhkan (Cyganiak, dkk, 2012).

2.8 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia

sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL).

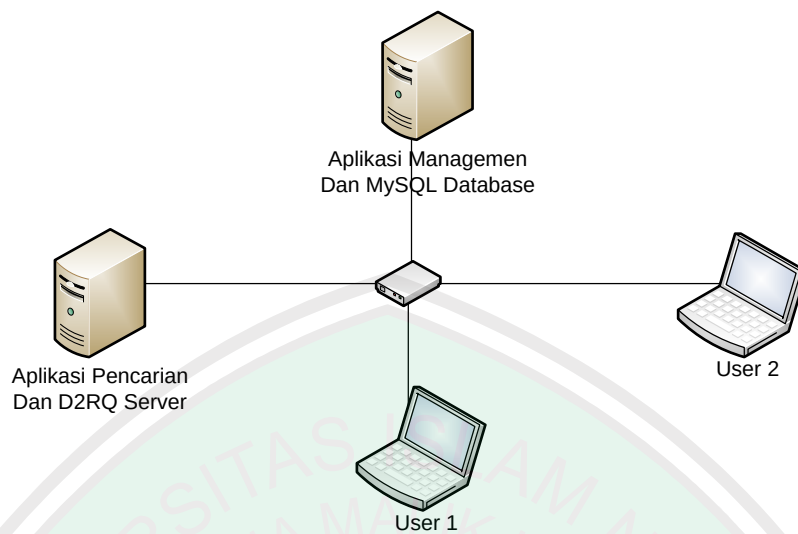
MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan (Edy Winarno, dkk.,2011).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Perancangan sistem utamanya pada penelitian ini terbagi menjadi dua, yaitu perancangan *ontology* dan perancangan aplikasi. Pada perancangan *ontology* membahas tentang bagaimana membuat *ontology*, yaitu dimulai dengan penentuan objek, property dari objek dan relasi antar objek yang nantinya akan diimplementasikan pada pembuatan database. Sedangkan pada perancangan aplikasi membahas mengenai perancangan aplikasi dengan menggunakan teknologi *semantic web*. Yaitu mulai dari gambaran umum *semantic web* hingga desain interface beserta spesifikasi kebutuhan sistem untuk aplikasi. Pada penelitian ini akan membuat dua aplikasi yang saling berhubungan, yaitu aplikasi manajemen data laporan kerusakan fasilitas umum dan aplikasi pencarian menggunakan teknologi *semantic web*. Aplikasi manajemen berfungsi untuk menyimpan data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang yang berhubungan langsung dengan database MySQL. Sedangkan aplikasi pencarian dengan teknologi *semantic web* dalam mengambil data laporan kerusakan fasilitas umum tidak langsung mengambil data dari database MySQL tetapi akan mengakses RDF Map yang sudah di-*generate* sesuai dengan desain database yang sudah dibuat di awal. Untuk bisa mengakses RDF Map, aplikasi pencarian membutuhkan D2R Server sebagai servernya untuk menyimpan data dari database MySQL. Berikut adalah gambar desain aplikasi manajemen dengan aplikasi pencarian.



Gambar 3.1 Rancangan Aplikasi

Pada gambar 3.1 dijelaskan bahwa terdapat dua aplikasi, yaitu aplikasi manajemen dan aplikasi pencarian. Aplikasi manajemen meliputi manajemen data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang yang terdapat proses *insert*, *update* dan *delete*. Data-data yang tersimpan melalui aplikasi manajemen akan diakses oleh aplikasi pencarian melalui RDF Map yang dijalankan bersamaan dengan menjalankan D2R Server.

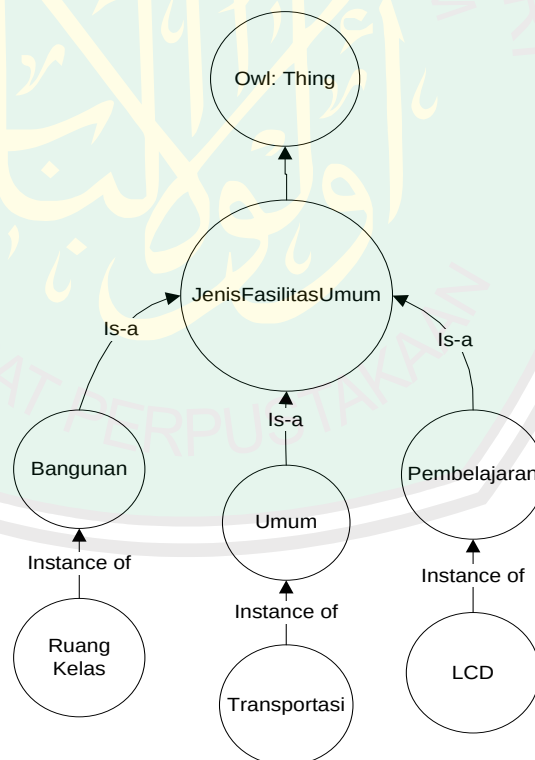
3.1 Pemodelan Ontology

Ontology merupakan bentuk dari suatu pengetahuan tentang dunia nyata yang kemudian direpresentasikan dalam suatu term yang terdiri dari sekumpulan konsep yang saling terhubung. Dari kumpulan konsep tersebut akan membentuk sebuah *tree* yang menunjukkan adanya relasi antar konsep.

Domain dalam pembuatan aplikasi ini adalah data fasilitas umum yang ada di UIN Maliki Malang dengan ruang lingkup ontologi sebagai berikut:

- a. Class Bangunan: Mencakup sub class ruang kuliah, ruang kantor, ruang dosen, ruang seminar, ruang rapat, ruang laboratorium, ruang studio, ruang perpustakaan dll.
- b. Class Sarana Umum: mencakup sub class drainase, listrik, jaringan telekomunikasi, transportasi, parkir, taman dll.
- c. Class Sarana Pembelajaran: mencakup sub class Papan tulis, OHP, LCD, mikrophone, alat peraga dll.

Berikut adalah contoh hirarki is-a dimana class yang lebih umum berada pada class yang lebih spesifik.

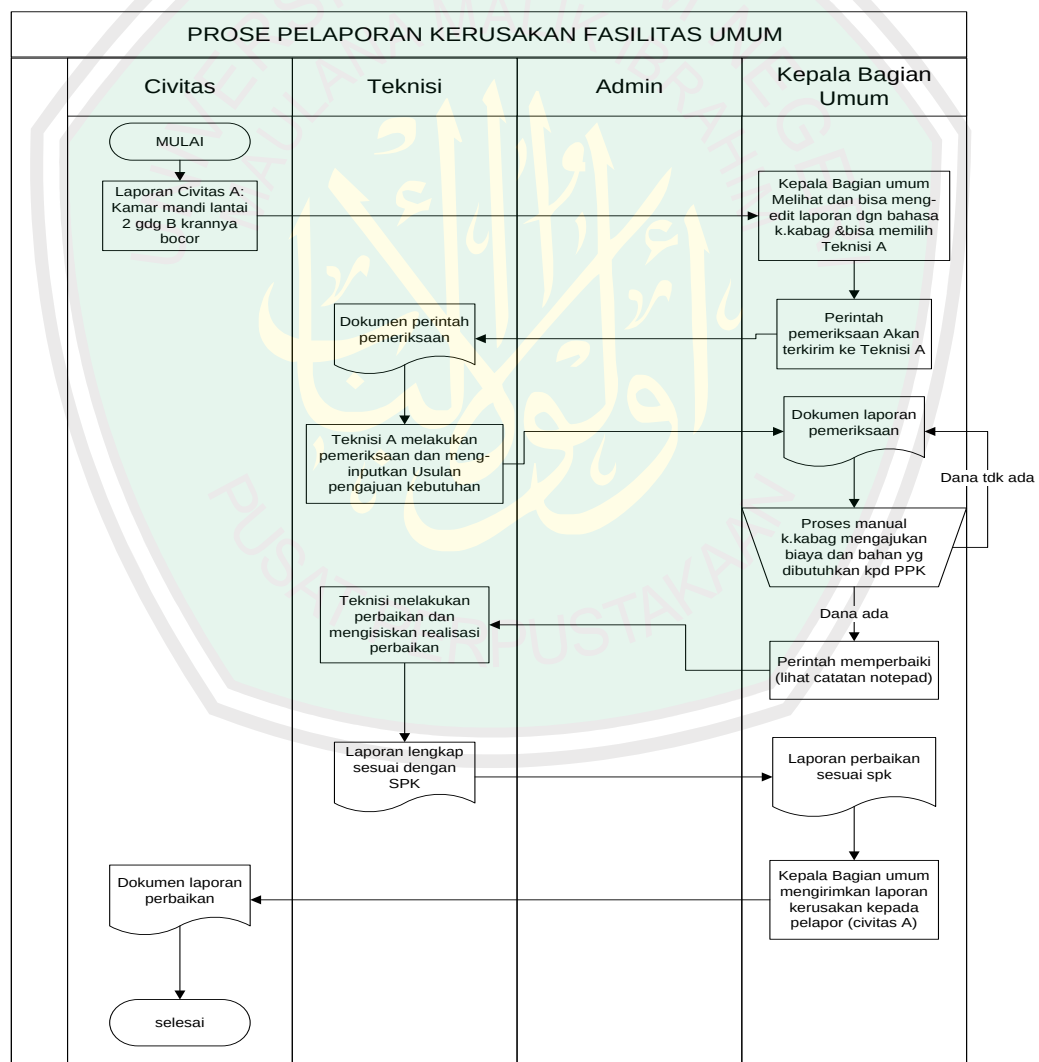


Gambar 3.2: Contoh Hirarki is-a

3.2 Perancangan Managemen Data Laporan

Berdasarkan gambaran umum objek penelitian diatas, pelaporan kerusakan fasilitas umum dimulai dengan adanya laporan civitas akademika melalui surat atau secara lisan kepada kepala bagia umum.

Alur proses pelapopran kerusakan inventaris ke bagian umum sebagaimana gambar 3.3.



Gambar 3.3 Alur proses pelaporan kerusakan fasilitas umum

Berdasarkan alur proses pada gambar 3.3 Sistem yang dibuat mempunyai empat level user yaitu: kepala bagian umum, administrator, teknisi dan civitas akademika. Berikut adalah tugas dari setiap user:

1. Kepala bagian umum mempunyai akses untuk melihat laporan, memberikan instruksi pengecekan dan perbaikan kepada teknisi.
2. Administrator mempunyai akses untuk menginputkan data dan menambahkan user baru.
3. Teknisi mempunyai akses melaporkan hasil pemeriksaan dan perbaikan yang telah dilakukan berdasarkan perintah kepala bagian umum.
4. Civitas akademik mempunyai akses untuk melakukan pelaporan kerusakan dan melihat laporan kerusakan yang telah diinputkan oleh civitas akademika lainnya.

Penjelasan alur sistem pada gambar 3.3 pada proses pelaporan yaitu, civitas atau yang bertindak sebagai pelapor melaporkan kerusakan fasilitas umum melalui aplikasi sesuai level civitas. Tidak semua civitas bisa melaporkan kerusakan fasilitas umum melalui aplikasi ini melainkan hanya civitas-civitas tertentu yang memperoleh izin dari bagian umum. Setelah civitas mengirimkan laporan kerusakan fasilitas umum, data tersebut akan diterima oleh ketua bagian. Ketua bagian akan memerintahkan teknisi untuk memeriksa kerusakan dan mencatat kebutuhan apa saja yang diperlukan untuk memperbaiki kerusakan tersebut. Setelah teknisi menyelesaikan tugas pengecekan data pengecekan akan dikirimkan pada ketua bagian, ketika terdapat dana untuk perbaikan maka ketua bagian akan membuat Surat Perintah Kerja (SPK) kepada teknisi yang

bersangkutan sebagai perintah perbaikan fasilitas umum yang rusak. Ketika tidak ada dana maka proses perbaikan tersebut ditunda sampai terdapat dana.

3.3 Perancangan Sistem

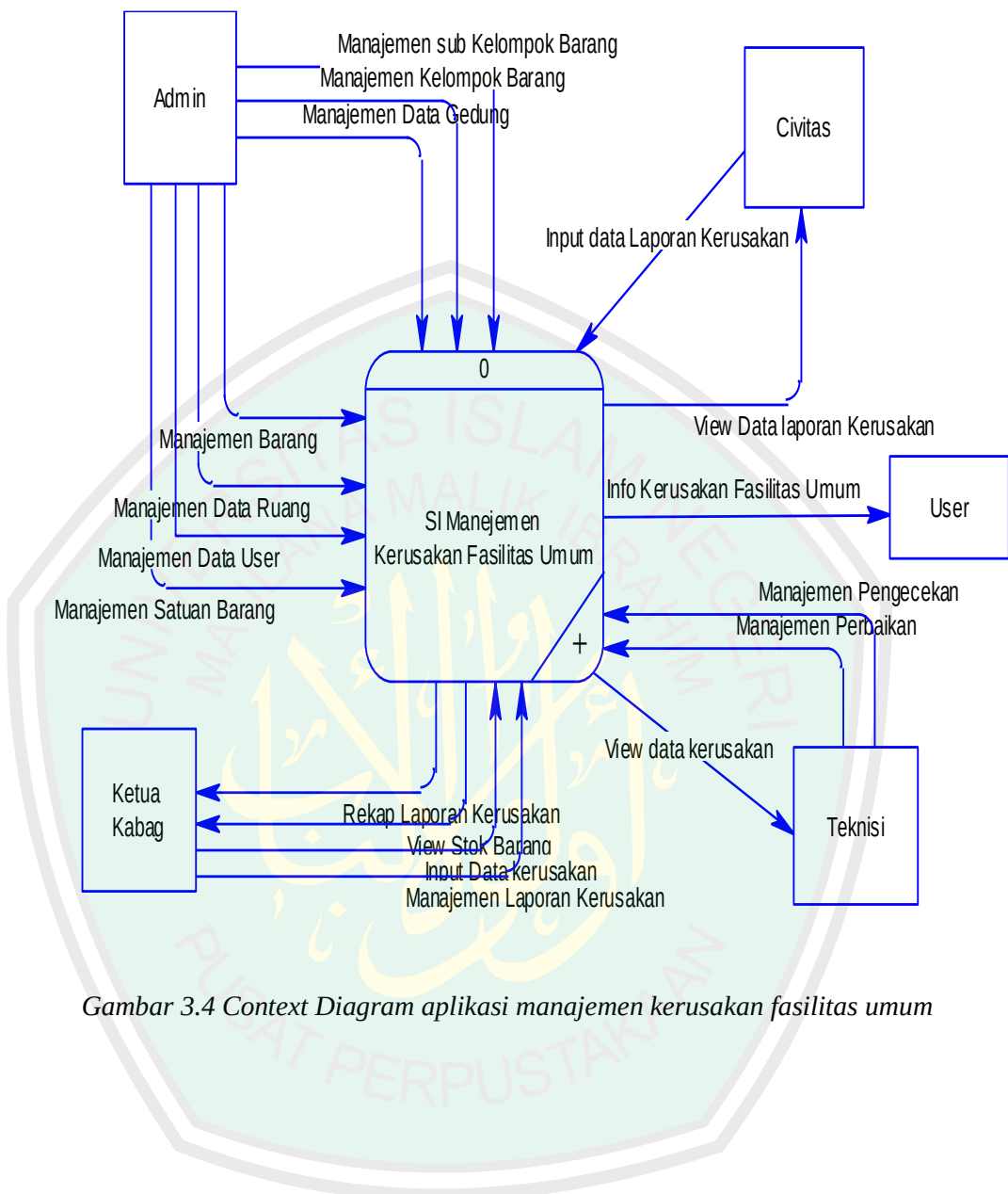
Perancangan sistem akan membahas semua hal yang terkait dengan perancangan aplikasi yang akan dibuat dalam penelitian ini. Rancangan desain aplikasi PHP, desain D2r server dan desain json untuk pencarian menggunakan *Sparql Query*.

3.3.1 Desain Aplikasi

3.3.1.1 Context Diagram

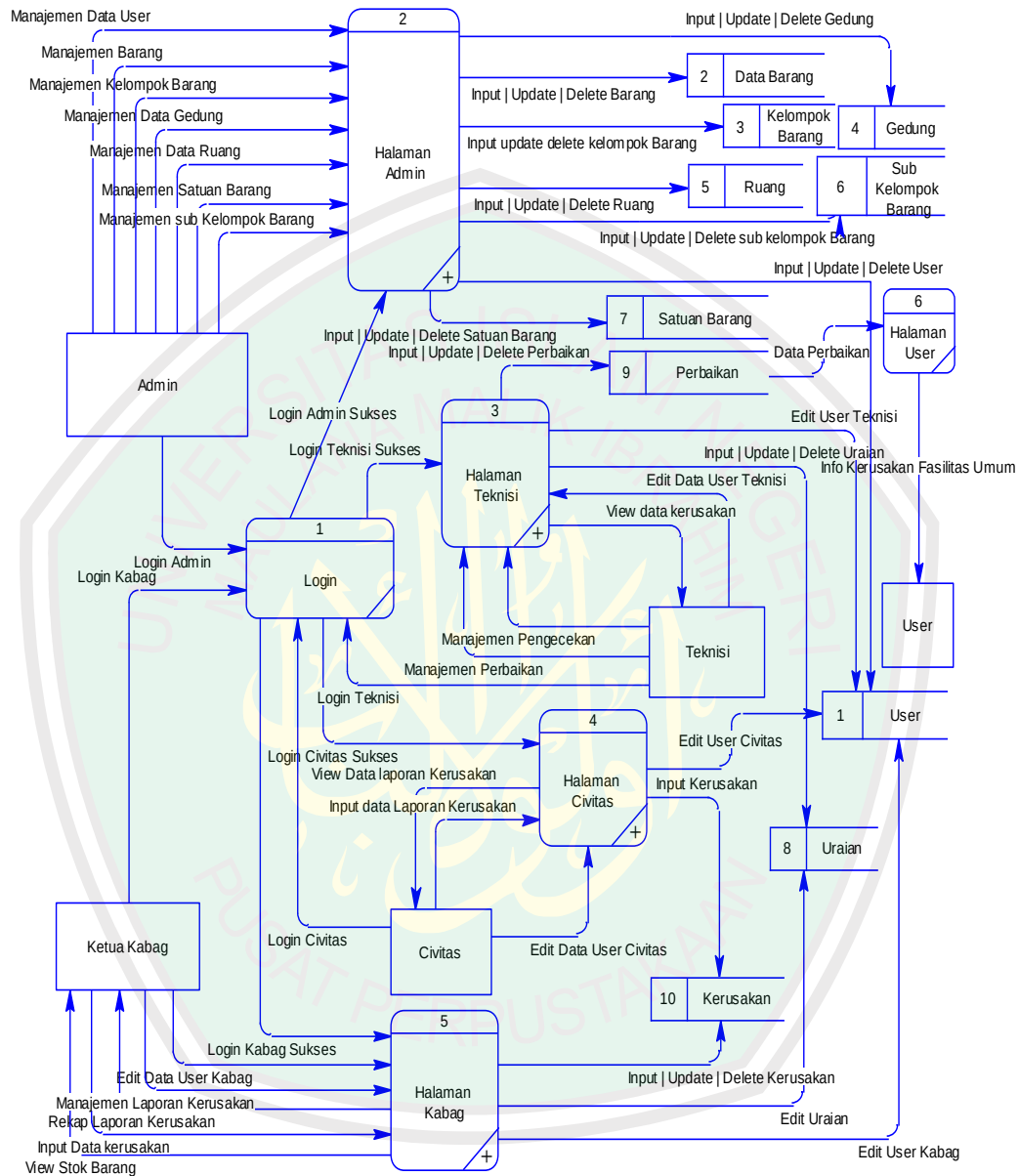
Arus data dalam aplikasi PHP dalam penelitian ini digambarkan dalam sebuah *Context Diagram* dan melibatkan 6 entitas yaitu:

- 1) Admin, memiliki 7 aliran data yaitu input master kelompok barang, master sub kelompok barang, master gedung, master ruang, manajemen data barang, manajemen data satuan barang, data stok barang
- 2) Teknisi, mempunyai 3 aliran data yaitu manajemen pengecekan, manajemen perbaikan dan data kerusakan
- 3) Kepala bagian umum, mempunyai 4 aliran data yaitu input kerusakan barang, manajemen laporan kerusakan, laporan kerusakan dan stok barang
- 4) Civitas, mempunyai 2 aliran yaitu input laporan kerusakan dan data laporan kerusakan.
- 5) User, mempunyai satu aliran data yaitu bisa melihat data kerusakan fasilitas umum.



Gambar 3.4 Context Diagram aplikasi manajemen kerusakan fasilitas umum

3.3.1.2 Data Flow Diagram



Gambar 3.5 data flow diagram aplikasi manajemen kerusakan fasilitas umum

Pada data flow diagram level 1 ini menggambarkan sub proses yang terdapat dalam aplikasi manajemen kerusakan fasilitas umum. Terdapat 6 proses yang digambarkan pada gambar 3.5 yaitu:

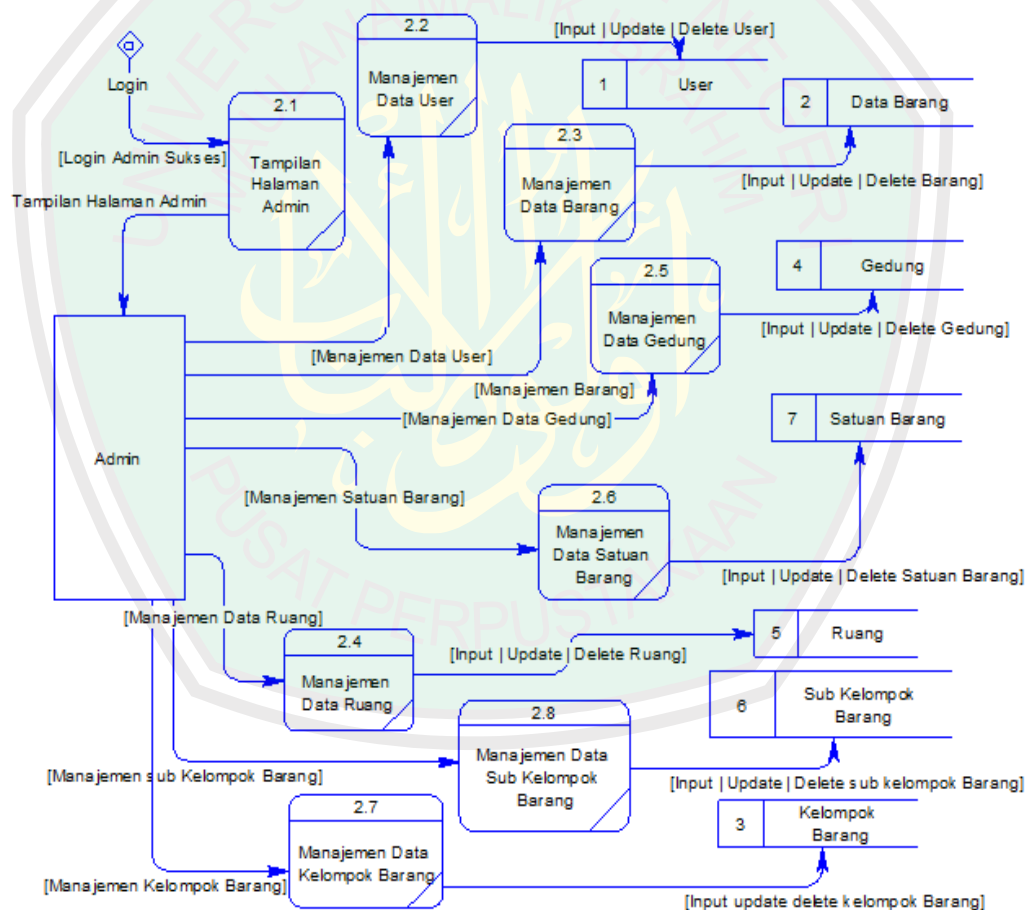
- 1) Proses login, proses ini digunakan untuk membedakan pemberian hak akses kepada setiap user yang akan melakukan akses data kedalam aplikasi.
- 2) Halaman admin, merupakan halaman yang akan dituju oleh admin setelah melakukan login dengan sukses sebagai administrator.
- 3) Halaman teknisi , merupakan halaman yang akan dituju oleh teknis setelah melakukan login dengan sukses sebagai teknisi.
- 4) Halaman Kepala bagian, merupakan halaman yang akan dituju oleh kepala bagian setelah melakukan login dengan sukses sebagai kepala bagian.
- 5) Halaman civitas, merupakan halaman yang akan dituju oleh civitas setelah melakukan login dengan sukses sebagai civitas
- 6) Halaman User halaman yang bisa diakses oleh semua pihak tanpa melakukan login terlebih dahulu.

3.3.1.3 DFD Level 2

Pada dfd level 2 akan menjelaskan setiap proses dari sistem informasi manajemen kerusakan fasilitas umum, yaitu:

1) DFD Level 2 Pada Admin

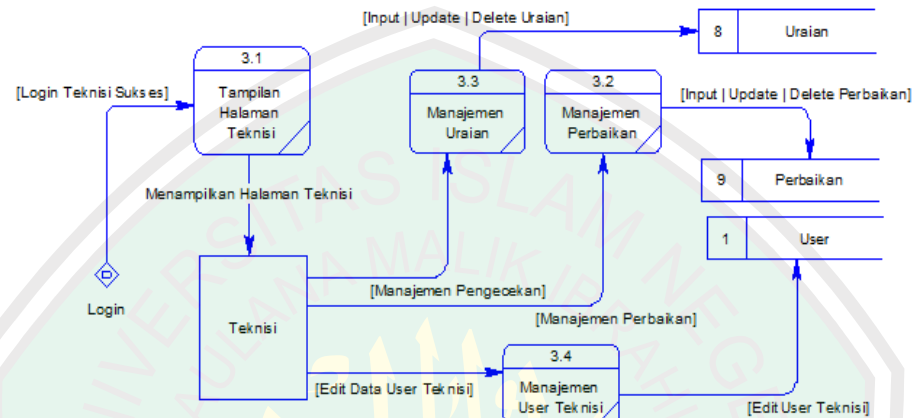
Pada gambar 3.4 merupakan DFD Level 2 Pada Admin yang menjelaskan tentang semua proses yang terjadi pada halaman admin.



Gambar 3.6 DFD Level 2 Pada Admin

2) DFD Level 2 Pada Teknisi

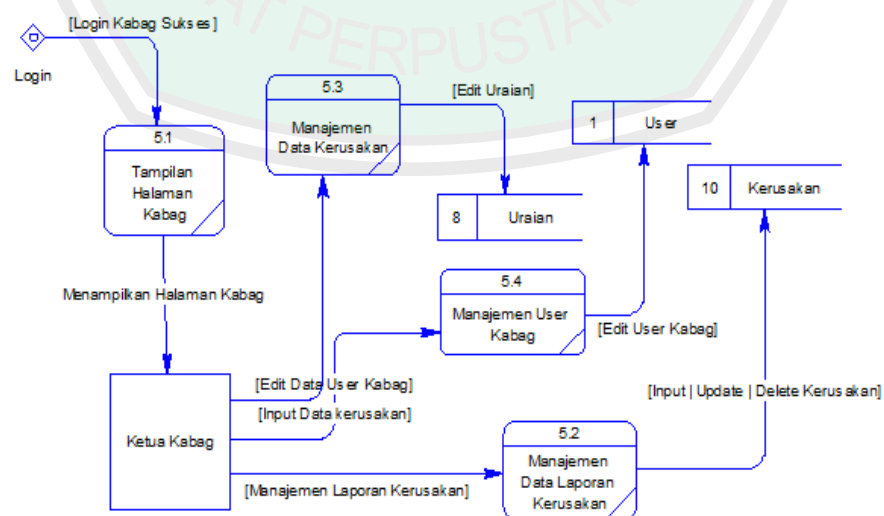
Pada gambar 3.5 merupakan DFD Level 2 Pada Teknisi yang menjelaskan tentang semua proses yang terjadi pada halaman teknisi.



Gambar 3.7 DFD Level 2 Pada Teknisi

3) DFD Level 2 Pada Kepala bagian

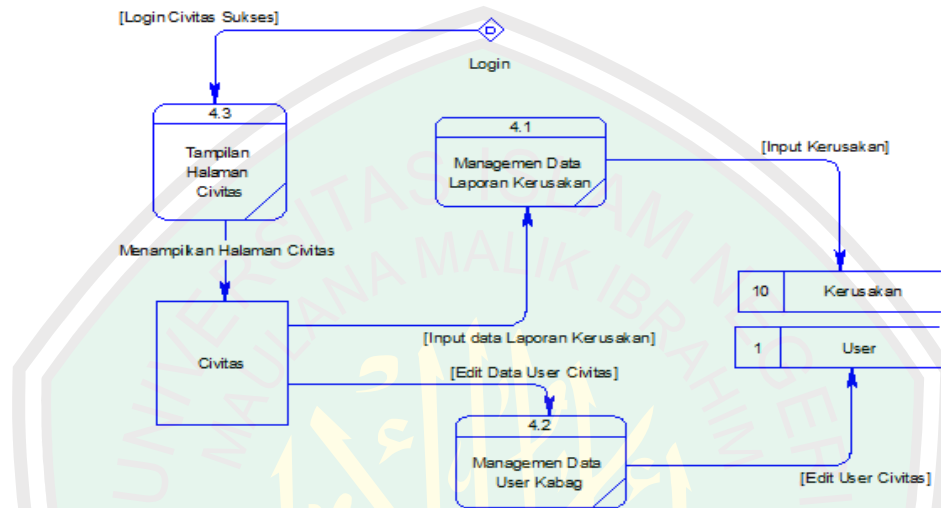
Pada gambar 3.6 merupakan DFD Level 2 Pada Kepala bagian yang menjelaskan tentang semua proses yang terjadi pada halaman kepala bagian.



Gambar 3.8 DFD Level 2 Pada Ketua Kepala bagian

4) DFD Level 2 Pada Civitas

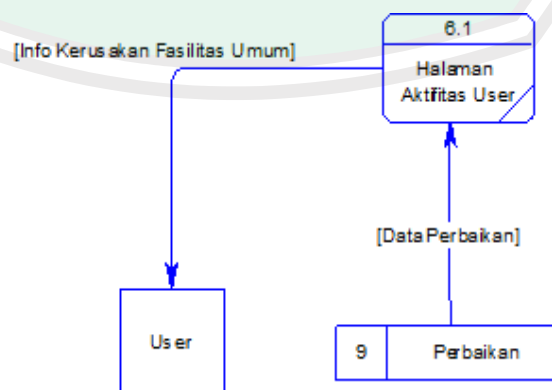
Pada gambar 3.7 merupakan DFD Level 2 Pada Civitas yang menjelaskan tentang semua proses yang terjadi pada halaman civitas.



Gambar 3.9 DFD Level 2 Pada Civitas

5) DFD Level 2 Pada User

Pada gambar 3.8 merupakan DFD Level 2 Pada user yang menjelaskan tentang semua proses yang terjadi pada halaman user.



Gambar 3.10 DFD Level 2 Pada User

3.3.1.4 *Entity Relational Diagram (ERD)*

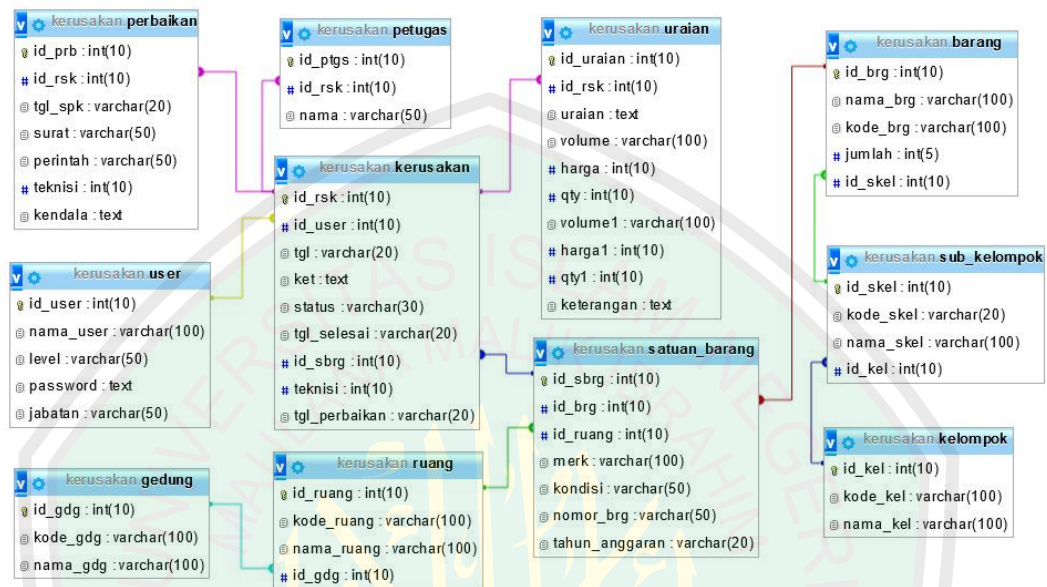
Pernacangan data pada sistem ini digambarkan dengan ERD (Entity Relation Diagram). ERD adalah sebuah diagram yang menggambarkan hubungan/relasi antar entitas (entity), dan setiap entitas terdiri atas satu atau lebih atribut yang mempresentasikan seluruh kondisi atau fakta dari “Dunia Nyata” yang ditinjau (Winarko, Edi, 2006:13).

ERD juga didefinisikan suatu diagram yang digunakan untuk menggambarkan sistem database yang digunakan. ERD terdiri dari tabel, view, dan relasi ERD digunakan untuk menunjukkan hubungan antara entitas dan database dan objek-objek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah basis data dan bagaimana hubungan yang terjadi diantara objek-objek tersebut.

ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Sehingga dapat digambarkan secara lebih sistematis dengan menggunakan ERD. Pembuatan desain sistem program melalui Data Flow Diagram (DFD) dan Entity Relational Database (ERD) dapat dibuat tabel-tabel database yang akan dikelola dan digunakan untuk menjalankan aplikasi.

Entity Relational Database (ERD) dalam aplikasi ini meliputi 10 relasi yang digunakan dalam proses penyimpanan dan menampilkan data. Terdiri dari 11 tabel yang digunakan untuk menyimpan data yang diperlukan dalam aplikasi

Sistem Informasi Manajemen Kerusakan Fasilitas Umum. Berikut ini adalah gambar ERD pada Aplikasi Laporan Kerusakn Fasilitas Umum:



Gambar 3.11 ERD Pada aplikasi Pelaporan Kerusakan Fasilitas Umum

Berikut ini adalah nama-nama tabel yang digunakan beserta *field-field* yang terdapat pada masing-masing tabel.

1. Tabel Barang

Tabel ini digunakan sebagai media penyimpanan data barang

Tabel 3.1: tabel barang pada database kerusakan

Column	Type	Null	Default	Links to
id_brg	int(10)	No		
nama_brg	varchar(100)	No		
kode_brg	varchar(100)	No		
jumlah	int(5)	No		
id_skel	int(10)	No		sub_kelompok -> id_skel

2. Tabel Gedung

Tabel ini digunakan sebagai media penyimpanan data gedung

Tabel 3.2: tabel gedung pada database kerusakan

Column	Type	Null
id_gdg	int(10)	No
kode_gdg	varchar(100)	No
nama_gdg	varchar(100)	No

3. Tabel Kelompok

Tabel ini digunakan sebagai media penyimpanan data kelompok

Tabel 3.3: tabel kelompok pada database kerusakan

Column	Type	Null
id_kel	int(10)	No
kode_kel	varchar(100)	No
nama_kel	varchar(100)	No

4. Tabel Kerusakan

Tabel ini digunakan sebagai media penyimpanan data kerusakan

Tabel 3.4: tabel kerusakan pada database kerusakan

Column	Type	Null	Default	Links to
id_rsk	int(10)	No		
id_user	int(10)	No		user -> id_user
tgl	varchar(20)	No		
ket	text	No		
status	varchar(30)	No		
tgl_selesai	varchar(20)	No		
id_sbrg	int(10)	No		satuan_barang -> id_sbrg
teknisi	int(10)	No		
tgl_perbaikan	varchar(20)	No		

5. Tabel Perbaikan

Tabel ini digunakan sebagai media penyimpanan data Perbaikan

Tabel 3.5: tabel perbaikan pada database kerusakan

Column	Type	Null	Default	Links to
id_prb	int(10)	No		
id_rsk	int(10)	No		kerusakan -> id_rsk
tgl_spk	varchar(20)	No		
surat	varchar(50)	No		
perintah	varchar(50)	No		
teknisi	int(10)	No		
kendala	Text	No		

6. Tabel Petugas

Tabel ini digunakan sebagai media penyimpanan data Petugas

Tabel 3.6: tabel petugas pada database kerusakan

Column	Type	Null	Default	Links to
id_ptgs	int(10)	No		
id_rsk	int(10)	No		kerusakan -> id_rsk
nama	varchar(50)	No		

7. Tabel Ruang

Tabel ini digunakan sebagai media penyimpanan data Ruang

Tabel 3.7: tabel ruang pada database kerusakan

Column	Type	Null	Default	Links to
id_ruang	int(10)	No		
kode_ruang	varchar(100)	No		
nama_ruang	varchar(100)	No		
id_gdg	int(10)	No		gedung -> id_gdg

8. Tabel Satuan Barang

Tabel ini digunakan sebagai media penyimpanan data Satuan Barang

Tabel 3.8: tabel satuan barang pada database kerusakan

Column	Type	Null	Default	Links to
id_sbrg	int(10)	No		
id_brg	int(10)	No		barang -> id_brg
id_ruang	int(10)	No		ruang -> id_ruang
merk	varchar(100)	No		
kondisi	varchar(50)	No		
nomor_brg	varchar(50)	No		
tahun_anggaran	varchar(20)	No		

9. Tabel Sub Kelompok

Tabel ini digunakan sebagai media penyimpanan data Sub Kelompok

Tabel 3.9: tabel sub kelompok pada database kerusakan

Column	Type	Null	Default	Links to
id_skel	int(10)	No		
kode_skel	varchar(20)	No		
nama_skel	varchar(100)	No		
id_kel	int(10)	No		kelompok -> id_kel

10. Tabel Uraian

Tabel ini digunakan sebagai media penyimpanan data uraian

Tabel 3.10: tabel uraian pada database kerusakan

Column	Type	Null	Default	Links to
id_uraian	int(10)	No		
id_rsk	int(10)	No		kerusakan -> id_rsk
uraian	Text	No		
volume	varchar(100)	No		
harga	int(10)	No		
qty	int(10)	No		
volume1	varchar(100)	No		
harga1	int(10)	No		
qty1	int(10)	No		
keterangan	Text	No		

11. Tabel User

Tabel ini digunakan sebagai media penyimpanan data User

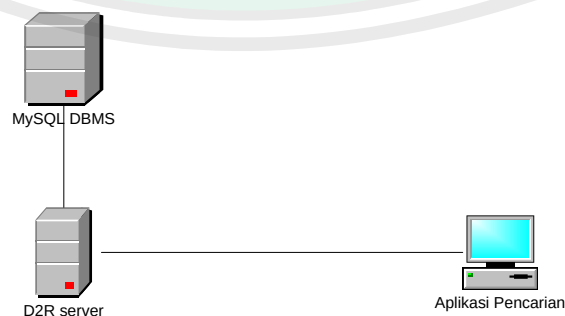
Tabel 3.11: tabel user pada database kerusakan

Column	Type	Null
id_user	int(10)	No
nama_user	varchar(100)	No
level	varchar(50)	No
password	text	No
jabatan	varchar(50)	No

3.4 Desain Teknologi Semantic Web

3.4.1 Aplikasi Pencarian

Aplikasi pencarian digunakan untuk menampilkan data hasil query SPARQL yang mengakse end point dari D2R Server aplikasi manajemen data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Hasil pencarian akan menampilkan data laporan kerusakan fasilitas umum keseluruhan ataupun sesuai dengan status laporan tersebut, baik yang masih dalam proses pengerjaan, selesai ataupun pengecekan. Gambar 3.12 merupakan desain sistem aplikasi pencarian secara umum.



Gambar 3.12 Desain aplikasi pencarian

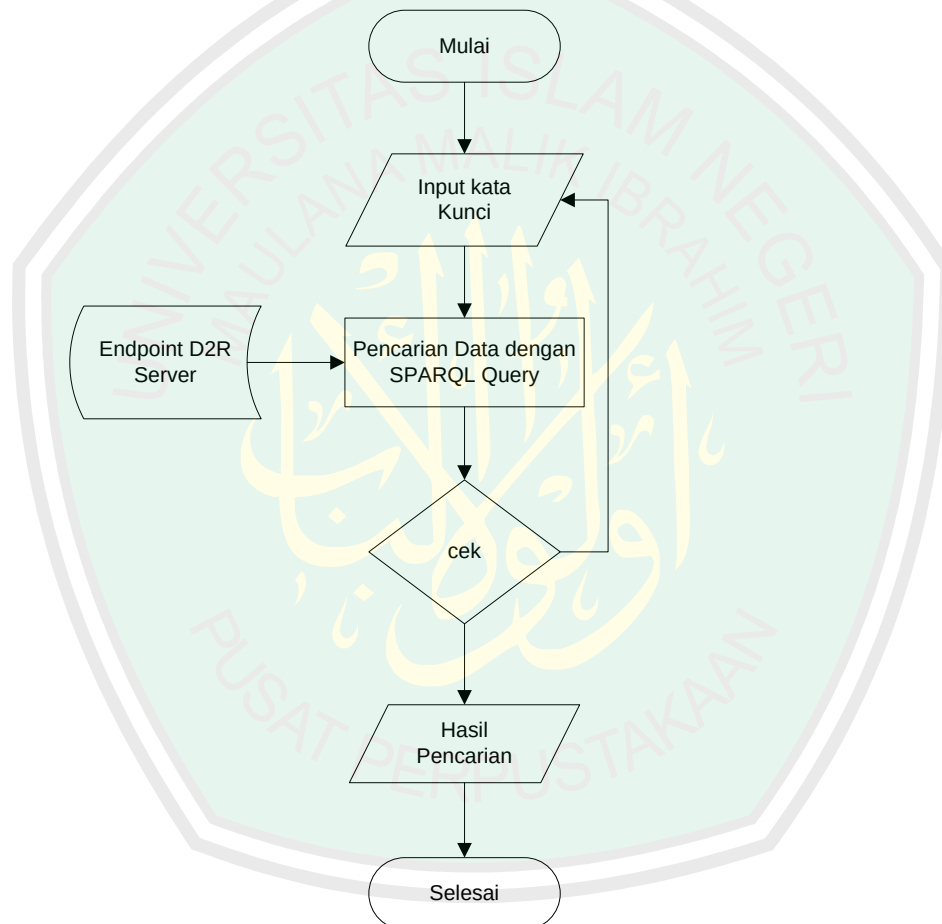
3.4.2 Proses Pencarian Dengan Teknologi Semantic Web

Pencarian pada aplikasi ini akan ditempatkan pada user umum, jadi untuk mengakses aplikasi pencarian ini tidak dibutuhkan *username* dan *password*. Aplikasi pencarian tersebut akan digunakan untuk mencari data kerusakan fasilitas umum yang sudah pernah ditangani oleh bagian umum baik yang sudah selesai pengerjaannya atau masih dalam proses pengerjaan di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pada proses pencarian menggunakan teknologi *semantic web* dan menggunakan algoritma *rabin-karp* untuk mempermudah pencarian *string* pada data kerusakan fasilitas umum

Pencarian dengan jaringan semantik berarti pencarian data berdasarkan kata kunci penelusuran dan makna yang terkait dengan kata kunci tersebut. Pencarian semantik berusaha untuk meningkatkan akurasi pencarian dengan memahami maksud pencari dan makna kontekstual seperti yang ditampilkan dalam data pencarian. Dengan kata lain, pencarian semantik adalah pencarian suatu konten berdasarkan konteks yang tepat. Konten adalah teks tertulis sedangkan konteks adalah kondisi keberadaan teks tersebut. Tujuan pencarian semantik adalah mencari konten yang sesuai dengan konteks yang diinginkan pengguna (Rahutomo, 2009). Ada dua pencarian semantic Jenis yang pertama adalah pencarian semantik dengan memberikan hasil berupa navigasi. Pengguna menggunakan mesin pencari sebagai alat navigasi untuk mengarahkan ke dokumen yang diinginkan. Navigasi ini dapat berupa link. Jenis yang kedua adalah dengan memasukkan frase atau kalimat yang menunjukkan pengguna untuk mendapatkan informasi. Pada jenis yang kedua ini, pengguna akan

mendapatkan keseluruhan dokumen yang akan memberikan informasi secara lengkap. Intinya, pencarian semantic memberikan saran bagi pengguna berdasarkan penarikan kesimpulan yang dilakukan oleh sistem berdasarkan batasan-batasan tertentu (Sarno, Riyanarto dkk: 2012).

Rancangan pencarian data menggunakan semantic web sebagai berikut



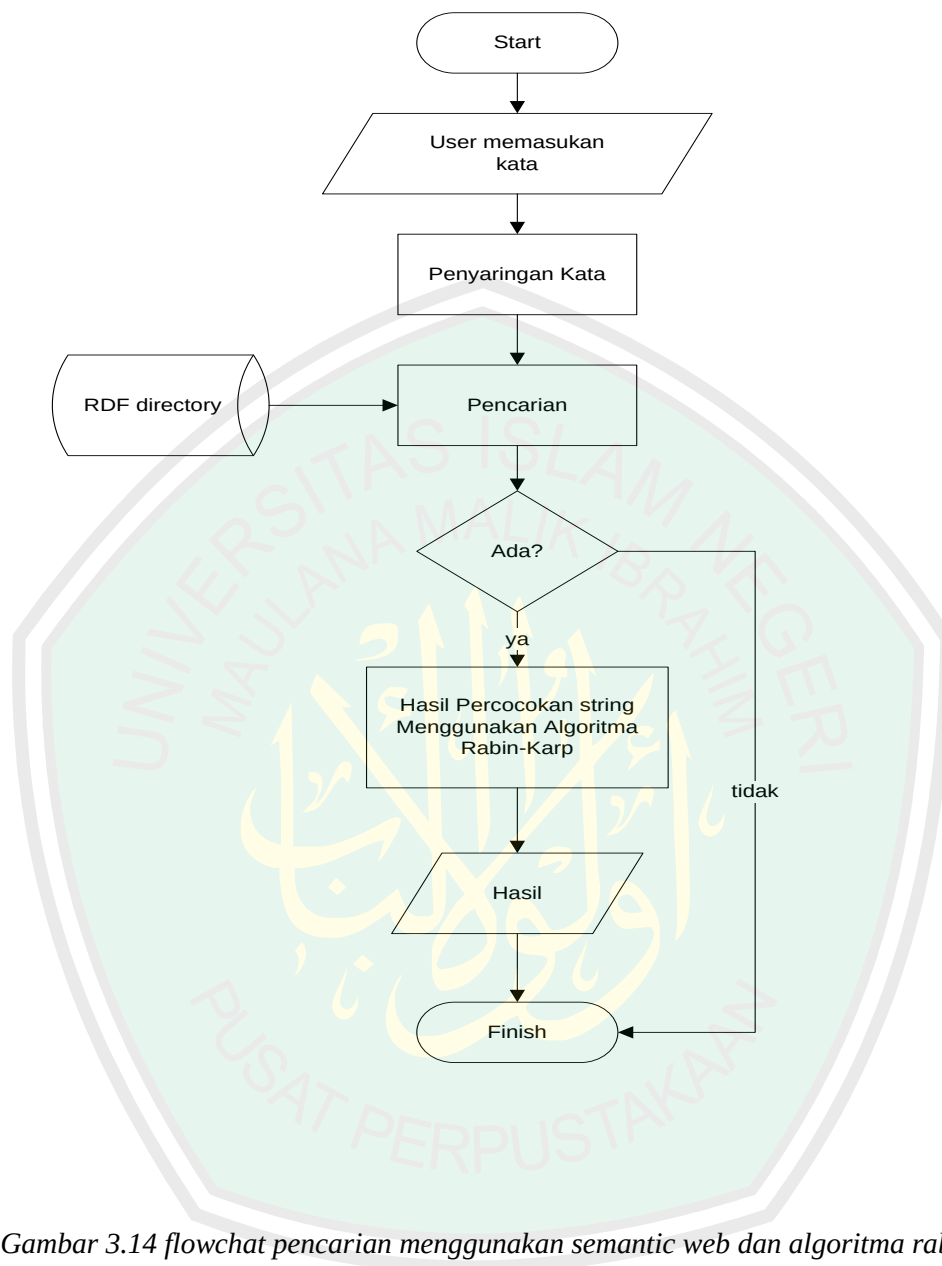
Gambar 3.13 merupakan flowchart pencarian data menggunakan semantic web

Berdasarkan gambar tersebut bisa dilihat bahwa terdapat dua direktori penyimpanan yaitu DBMS menggunakan *MySQL* yang kedua menggunakan RDF *Directory* menggunakan D2R Server. DBMS *MySQL* digunakan untuk menyimpan semua data laporan kerusakan fasilitas umum. Sedangkan D2R Server digunakan

sebagai jembatan aplikasi pencarian agar bisa mengakses data dari DBMS MySQL. mwnggunakan RDF Map bertipe *turtle* (.ttl).

Pada proses pencarian data kerusakan fasilitas umum menggunakan *semantic web technology*. ketika proses pencarian, maka sistem akan menyaring kata sesuai dengan kata kunci yang dimasukkan. Selanjutnya akan menggabungkan antara satu kata dengan kata yang masih ada hubungannya dengan kata kunci sesuai dengan desain ontology yang dibuat diawal. Berikut adalah flowchat pencarian fasilitas umum menggunakan semantic web dan algoritma rabin-karp.





Gambar 3.14 flowchat pencarian menggunakan semantic web dan algoritma rabin-karp

3.4.3 Proses Pencocokan *String* Dengan Algoritma *Rabin-Karp*

Misalkan Pattern adalah “bocor” dan text stringnya adalah “atapbocor”

Misalkan nilai hash adalah 10 dan gramnya 5 karakter. Pada pencocokan nilai hash diatas dengan text string mempunyai 5 kali rolling hash.

b	o	c	o	r
---	---	---	---	---

a	t	a	p	b	o	c	o	r
---	---	---	---	---	---	---	---	---

- Text pertama yang dicocokkan adalah “atapb = 5” ternyata nilainya tidak sama
- Text kedua yang dicocokkan adalah “tapbo = 6” ternyata nilainya tidak sama
- Text ketiga yang dicocokkan adalah “apboc = 7 ” ternyata nilainya tidak sama
- Text keempat yang dicocokkan adalah “pboco = 8 ” ternyata nilainya tidak sama
- Sampai rolling hash yang kelima ternyata nilai hashnya sama, maka proses rolling hash akan berhenti disini.

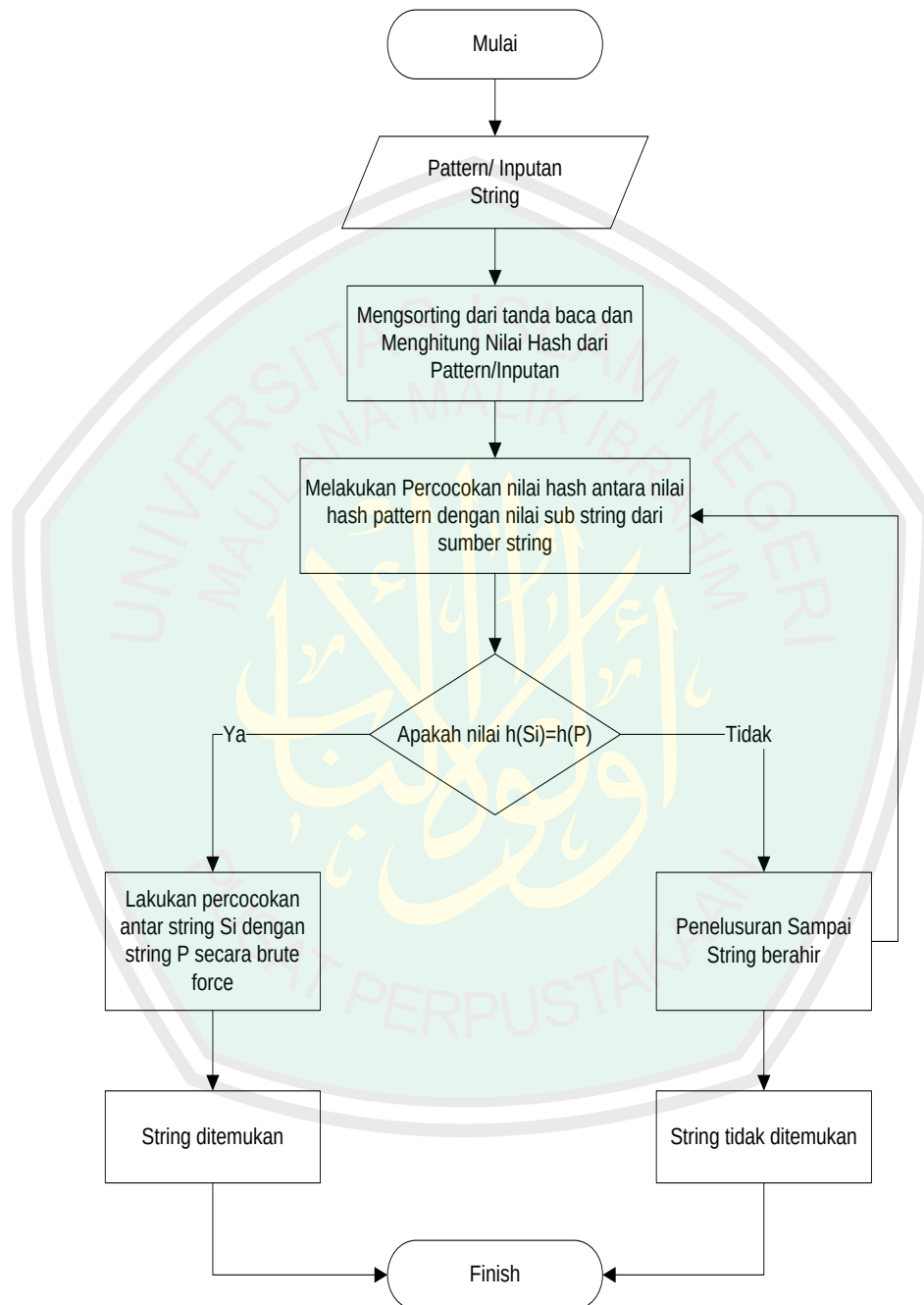
Setiap percocokan yang tidak ditemukan nilai yang sama pada nilai hashnya, maka *substring* pada teks akan bergeser satu karakter ke kanan. Algoritma tidak menghitung kembali nilai *hash substring*. Disinilah dilakukan apa yang disebut *rolling hash* yaitu mengurangi nilai karakter yang keluar dan menambahkan nilai karakter yang masuk sampai menemukan nilai hash yang sama.

b	o	c	o	r
---	---	---	---	---

a	t	a	p	b	o	c	o	r
---	---	---	---	---	---	---	---	---

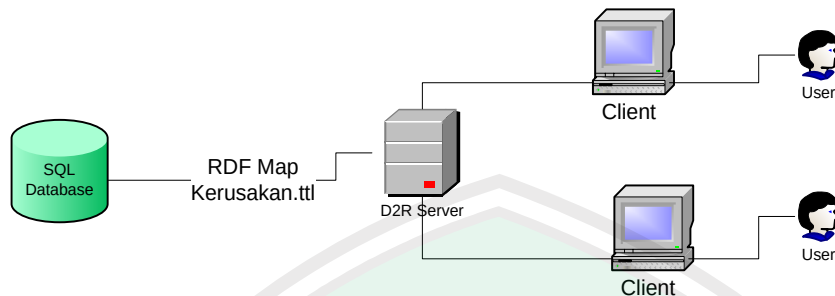
Ketika nilai hash cocok maka proses rolling hash akan berhenti.

Berikut adalah flowchart percocokan menggunakan algoritma Rabin-Karp



Gambar 3.15 flowchat pencarian menggunakan Algoritma Rabin-Karp

3.4.4 Desain D2R Server



Gambar 3.16 desain D2R Server

Data kerusakan fasilitas umum adalah data yang bersal dari kumpulan data yang sudah dijelaskan pada aplikasi PHP yang telah dijelaskan sebelumnya. Pada aplikasi PHP, database tidak diakses secara langsung. Pada saat proses pencarian data kerusakan fasilitas umum, aplikasi tidak mengakses file mapping hasil dari menjeneret database mySql menjadi file berupa *turtle* (.ttl). Proses ini sebagai implementasi teknologi semantic web dengan menggunakan D2R Server. D2R server sebagai sarana yang menjembatani aplikasi PHP supaya bisa mengakses data yang berbentuk RDF dengan menggunakan *Sparql Query*.

Aplikasi php bisa mengakses data kerusakan fasilitas umum jika mapping file sudah dijalankan dengan D2R Server. Apabila mapping file belum dijalankan, maka aplikasi tidak dapat mengakses data apapun. Desain d2r server bisa dilihat pada gambar 3.13

3.4.5 Mapping

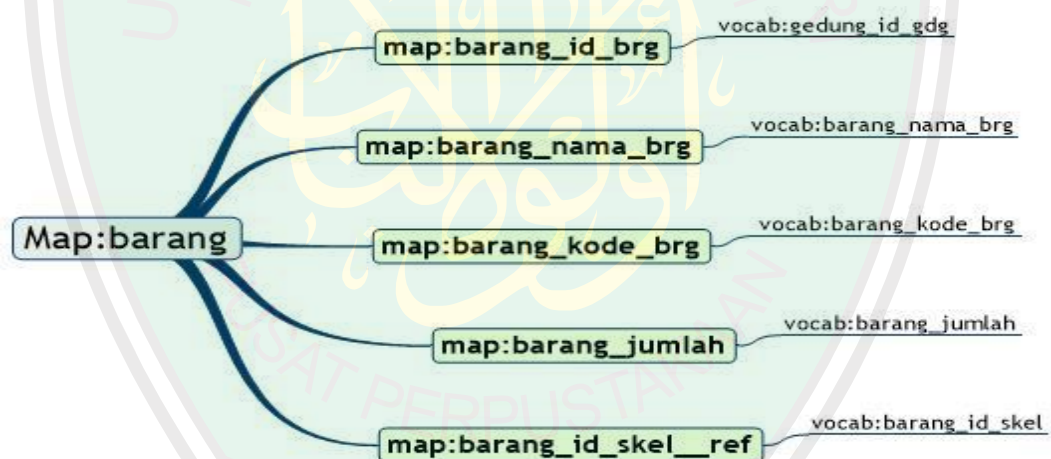
Mapping dituliskan dengan sintaks *turtle*. Dalam penelitian ini, mapping ditulis berdasarkan *template* yang telah ditentukan oleh *D2R Mapping Language*.

Map: database digunakan untuk melakukan koneksi ke *database* kerusakan, dalam *mapping* ini terdapat satu buah map: database. Map: ClassMap berfungsi untuk mendefinisikan kolom yang terdapat dalam database. Setiap classMap akan memiliki *Property Bridge*. *Property Bridge* digunakan untuk menginisialisasikan setiap kolom dalam satu tabel. *Vocab*: merupakan *property bridge* yang nanti akan digunakan sebagai variabel dalam *query* SPARQL.

3.4.6 RDF Skema

1. RDF Skema pada tabel barang

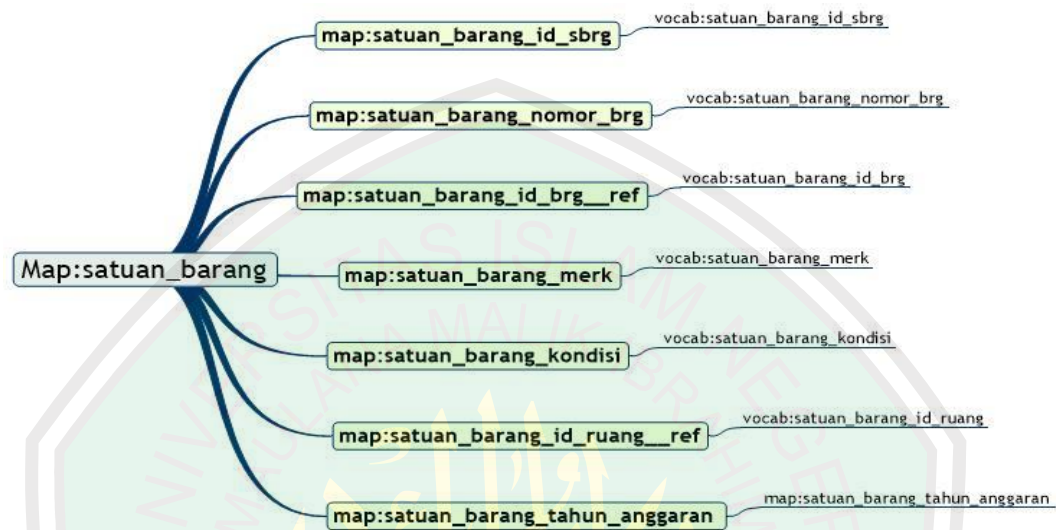
Skema RDF pada tabel barang mempunyai 5 *field*.



Gambar 3.17 RDF Skema pada tabel barang

2. RDF Skema pada tabel satuan_barang

Skema RDF pada tabel satuan_barang mempunyai 7 *field*.



Gambar 3.18 RDF Skema pada tabel satuan_barang

3. RDF Skema pada tabel gedung

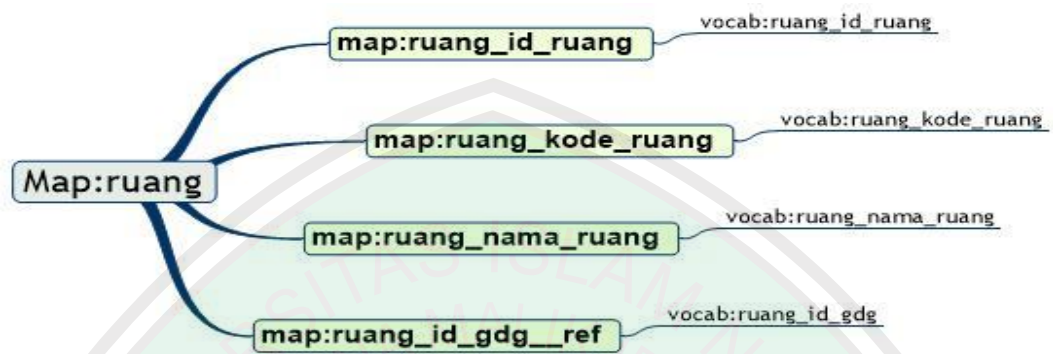
Skema RDF pada tabel gedung mempunyai 3 *field*.



Gambar 3.19 RDF Skema pada tabel gedung

4. RDF Skema pada tabel ruang

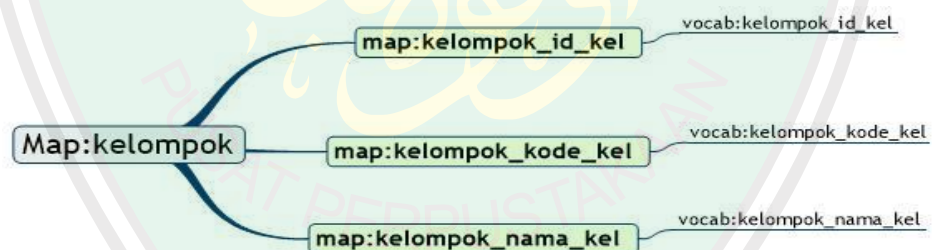
Skema RDF pada tabel ruang mempunyai 4 *field*.



Gambar 3.20 RDF Skema pada tabel ruang

5. RDF Skema pada tabel kelompok

Skema RDF pada tabel kelompok mempunyai 3 *field*.



Gambar 3.21 RDF Skema pada tabel kelompok

6. RDF Skema pada tabel sub_kelompok

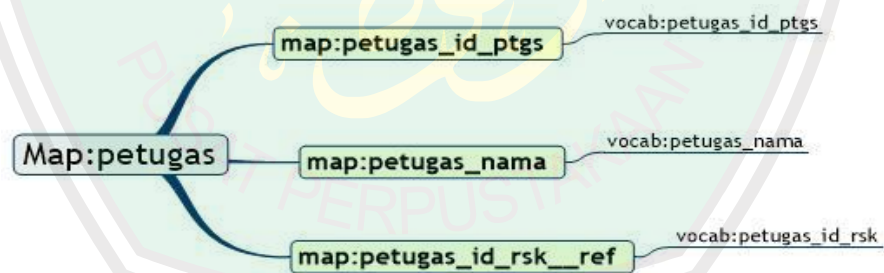
Skema RDF pada tabel sub_kelompok mempunyai 4 *field*.



Gambar 3.22 RDF Skema pada tabel sub_kelompok

7. RDF Skema pada tabel petugas

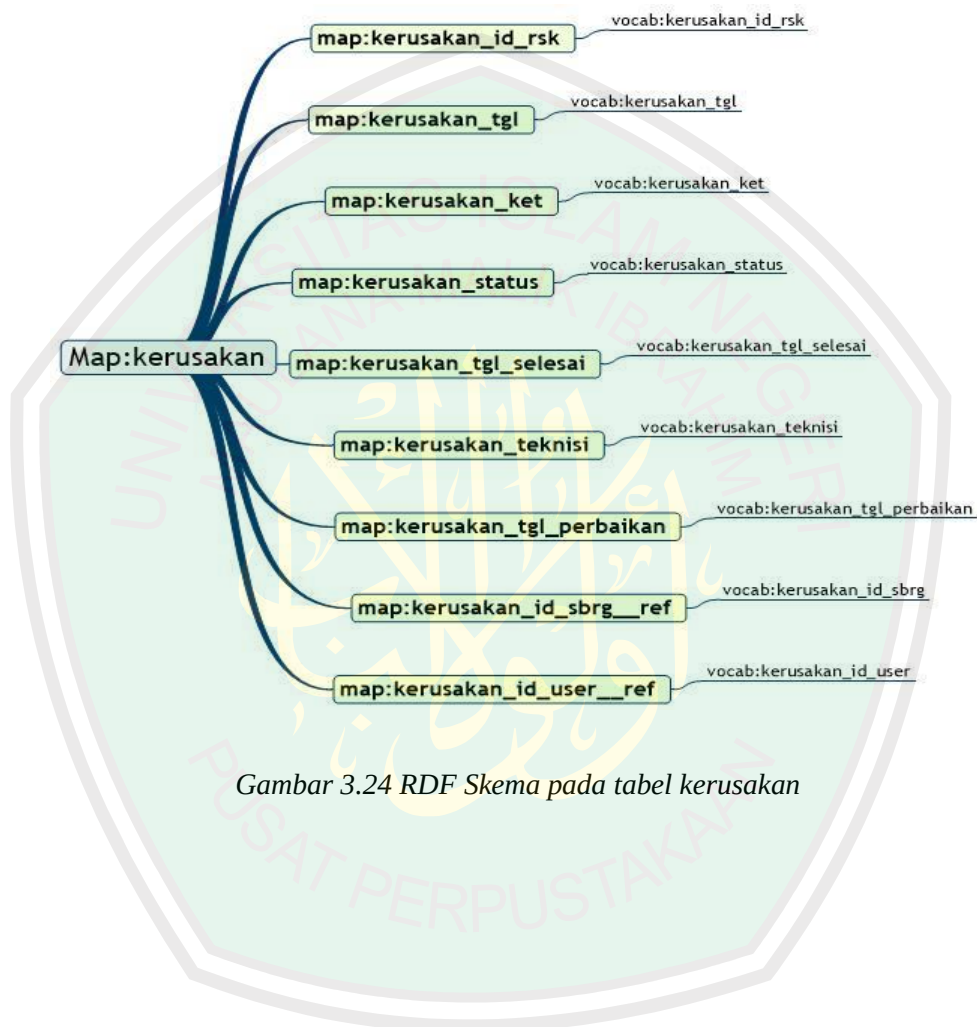
Skema RDF pada tabel petugas mempunyai 3 *field*.



Gambar 3.23 RDF Skema pada tabel petugas

8. RDF Skema pada tabel kerusakan

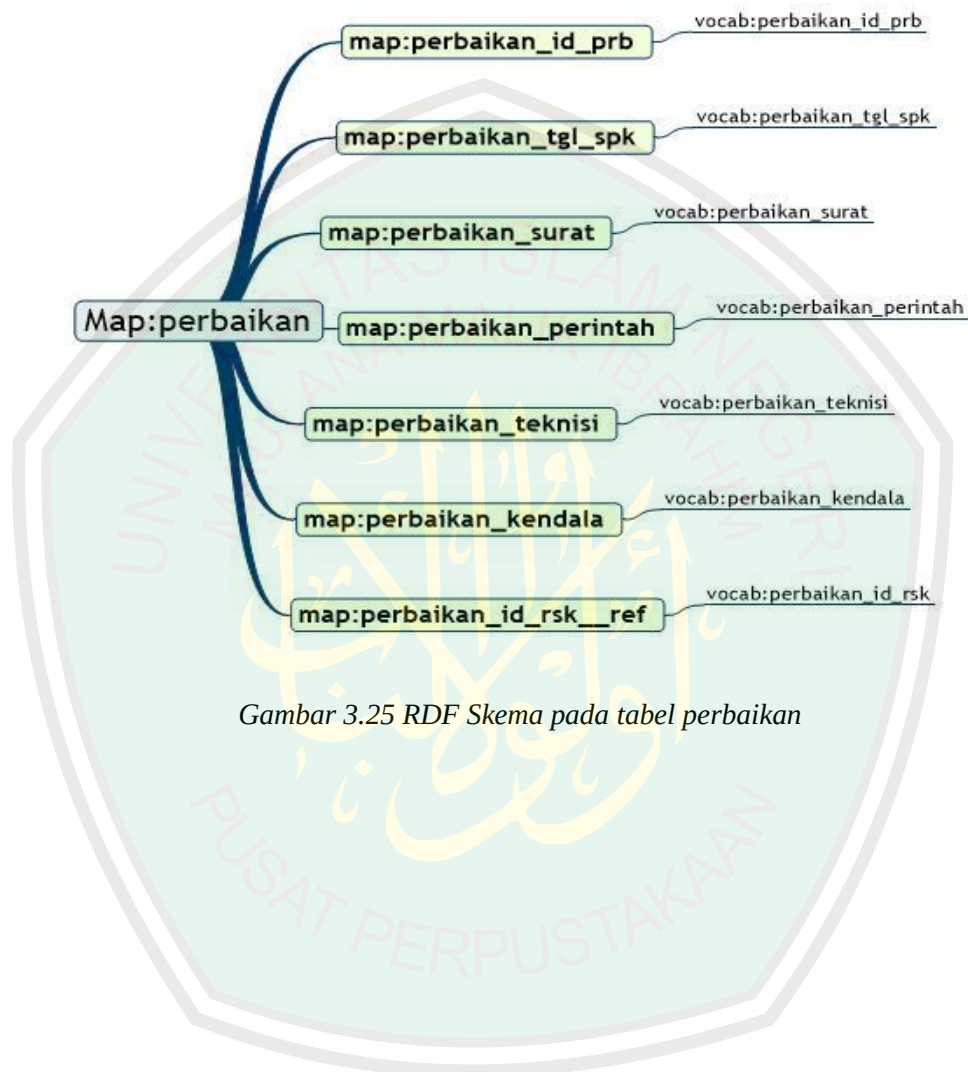
Skema RDF pada tabel kerusakan mempunyai 9 *field*.



Gambar 3.24 RDF Skema pada tabel kerusakan

9. RDF Skema pada tabel perbaikan

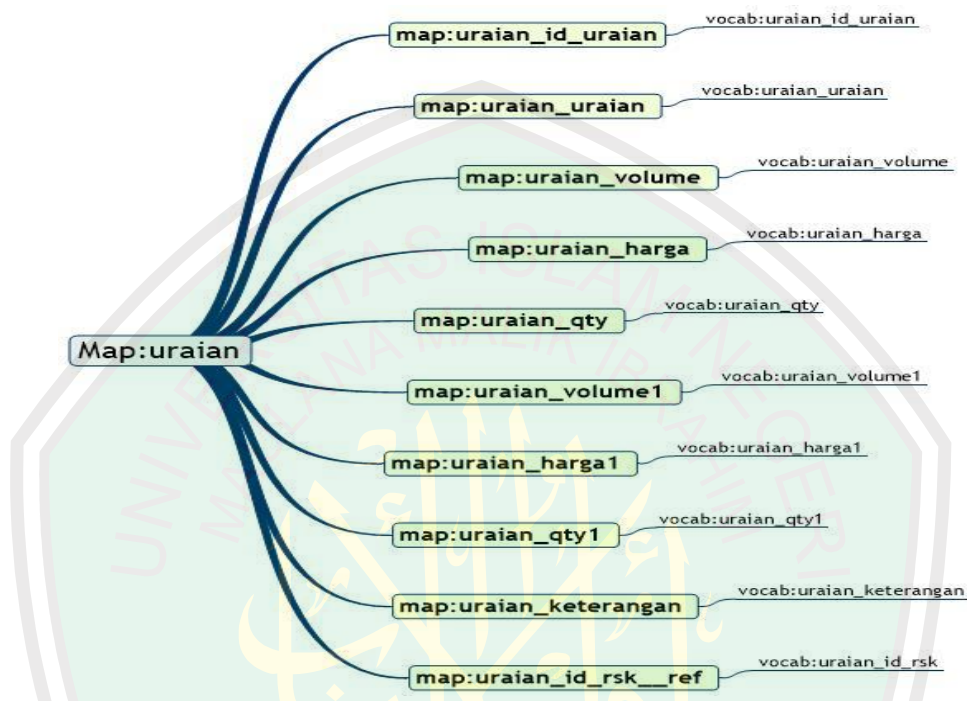
Skema RDF pada tabel perbaikan mempunyai 7 *field*.



Gambar 3.25 RDF Skema pada tabel perbaikan

10. RDF Skema pada tabel uraian

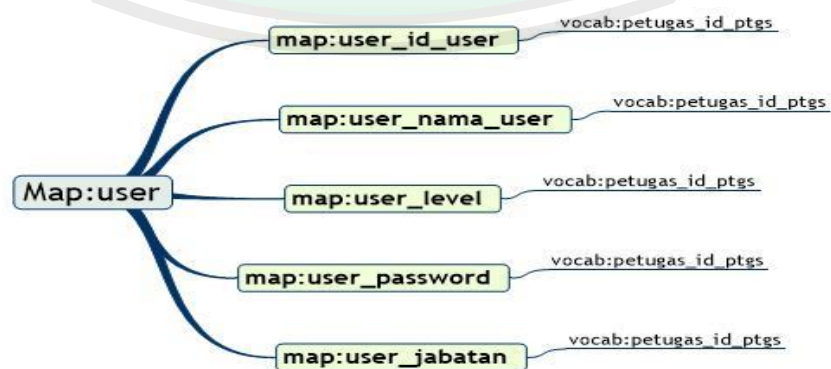
Skema RDF pada tabel uraian mempunyai 10 *field*.



Gambar 3.26 RDF Skema pada tabel uraian

11. RDF Skema pada tabel user

Skema RDF pada tabel user mempunyai 5 *field*.

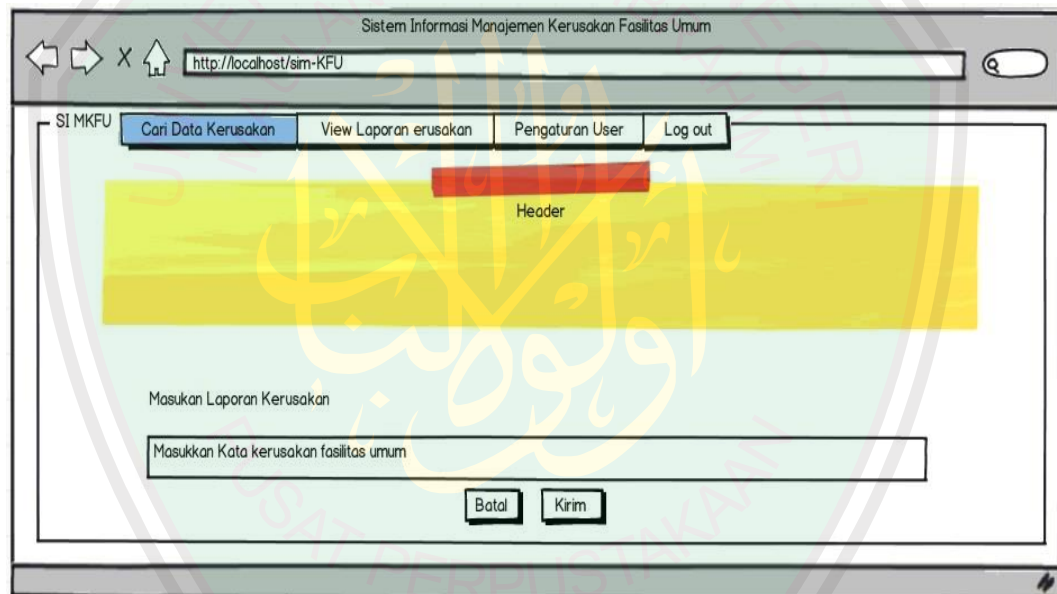


Gambar 3.27 RDF Skema pada tabel user

3.5 Desain *Interface* Aplikasi Managemen Data

3.5.1 Desain interface civitas

Desain interface untuk menginputkan laporan kerusakan fasilitas umum pada civitas. Data yang diinputkan tersebut akan dilihat oleh kepala bagian yang nantinya akan dilanjutkan perintah pemeriksaan oleh kepala bagian kepada teknisi. Berikut ini adalah gambar rancangan interface untuk menginputkan data kerusakan pada user civitas.

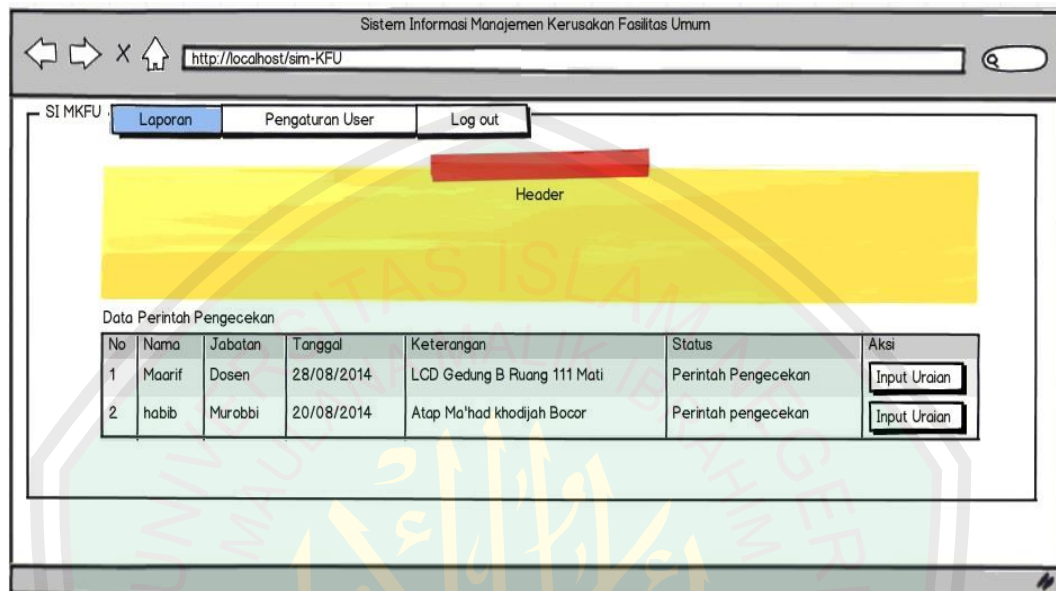


Gambar 3.28 adalah desain interface untuk pelaporan kerusakan fasilitas umum

3.5.2 Desain *Interface* Teknisi

Desain interface untuk manajemen kerusakan fasilitas umum pada teknisi. Data tersebut berasal dari laporan civitas yang sudah dilihat oleh kepala bagian lalu dilanjutkan perintah pengecekan kepada teknisi. Berikut ini adalah gambar

rancangan interface untuk menginputkan uraian kerusakan yang terdapat pada teknisi.



Gambar 3.29 adalah desain input uraian kerusakan fasilitas umum

Gambar berikut adalah desain interface menambahkan uraian kebutuhan untuk memperbaiki kerusakan fasilitas umum. Lebih jelasnya bisa dilihat pada gambar 3.30 berikut.

Sistem Informasi Manajemen Kerusakan Fasilitas Umum

http://localhost/aim-KFU

SI MKFU

Laporan Rekap Laporan Pengaturan User Log out

Header

Data Perintah Pengecekan

Uraian

Usulan Pengajuan Kebutuhan

Save Data

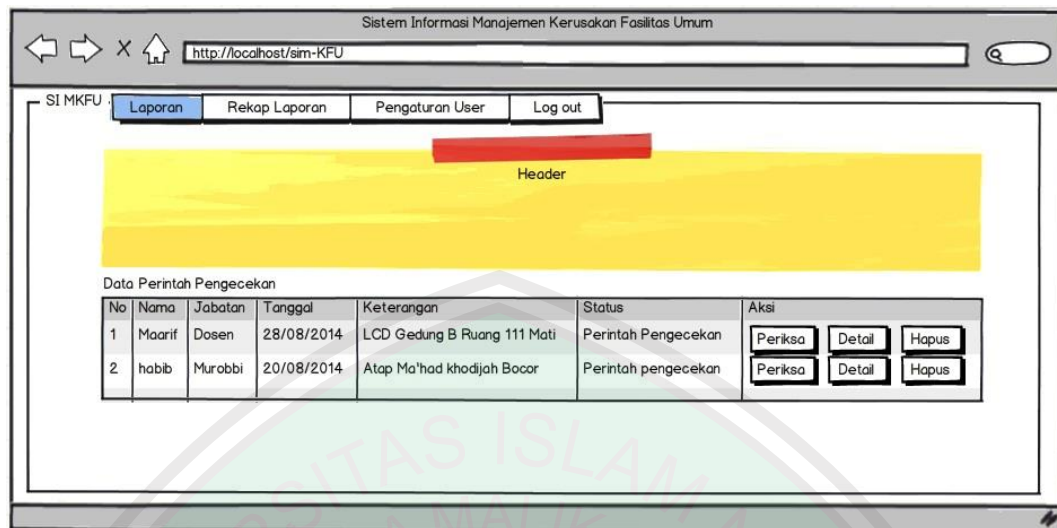
No	Uraian Kerusakan	Satuan	Harga	Jumlah	Satuan	harga	Jumlah	Ket
1	Lensa	3 Biji	50000	150000	-	-	-	

Back Kirim Uraian

Gambar 3.30 adalah desain perintah pengecekan kerusakan fasilitas umum

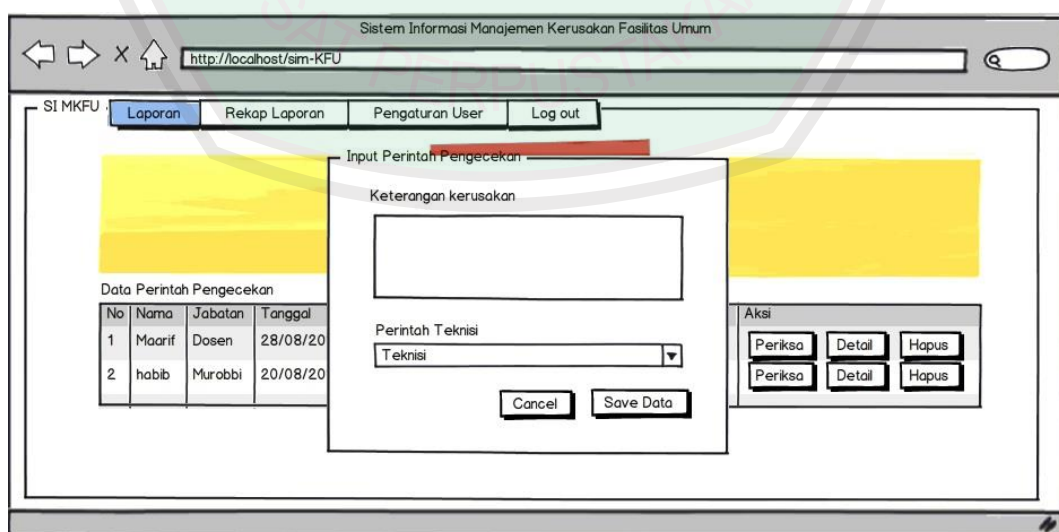
3.5.3 Desain Interface Kepala Bagian

Desain interface untuk manajemen kerusakan fasilitas umum pada kepala bagian. Data tersebut berasal dari laporan civitas yang masuk pada halaman kepala bagian, lalu kepala bagian memerintahkan pengecekan kepada teknisi. Berikut ini adalah gambar rancangan interface untuk memeriksa laporan kerusakan dari civitas yang terdapat pada kepala bagian.



Gambar 3.31 desain interface untuk memeriksa laporan kerusakan yang masuk

Berikut ini adalah desain interface pada proses pengecekan Pelaporan kerusakan umum yang dilaporkan oleh civitas akademika. Kepala bagian bisa melihat, mengubah dan juga melanjutkan atau tidak laporan tersebut pada proses pengecekan dan perbaikan. Lebih jelasnya bisa dilihat pada gambar 3.32 berikut.



Gambar 3.32 adalah desain perintah pengecekan kerusakan fasilitas umum

Berikutnya adalah desain interface untuk melihat detail pengerjaan kerusakan fasilitas umum yang dilakukan oleh teknisi, disini kepala bagian bisa melihat dan menyetujui atau tidak menyetujui pengajuan uraian barang yang dibutuhkan untuk memperbaiki kerusakan fasilitas umum oleh pihak teknisi. Lebih jelasnya bisa dilihat pada gambar 3.33

Sistem Informasi Manajemen Kerusakan Fasilitas Umum

http://localhost/sim-KFU

SI MKFU

Laporan Rekap Laporan Pengaturan User Log out

Header

Detail Perbaikan

Tanggal SPK : 29-08-2014
 Tindak Lanjut Atas Surat : civitas
 Tindak Lanjut Atas Perintah : rektor
 Pegawai Yang Diperintah : Teknisi
 Tanggal Pengerjaan : 29-08-2014
 Kendala :
 Pekerjaan yang dikerjakan : Icd gedung b 106 mati

No	Nama
1	Mohammad Arif
2	Suwamo

Uraian

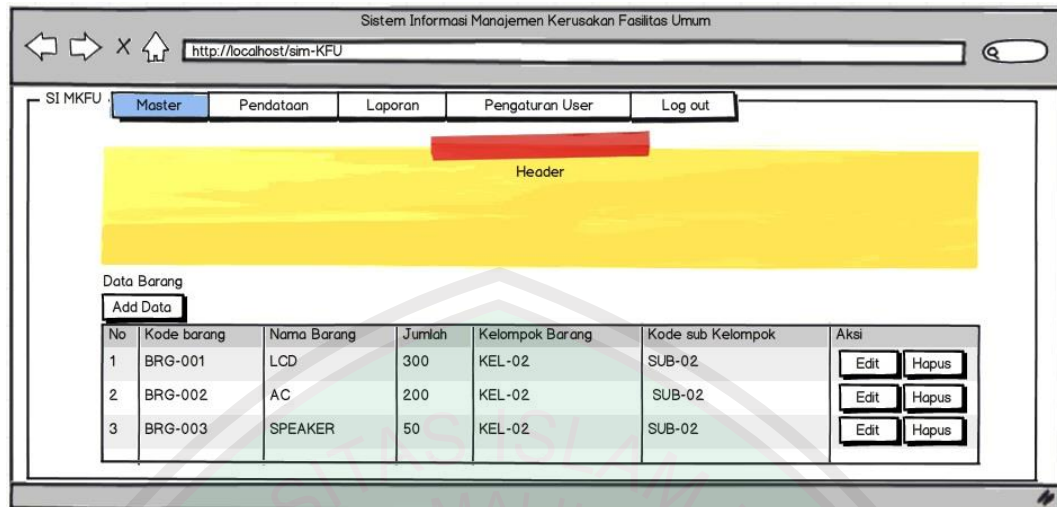
No	Uraian Kerusakan	Satuan	Harga	Jumlah	Satuan	harga	Jumlah	Ket
1	Lensa	3 Biji	50000	150000	-	-	-	

Back Print SPK Print Uraian

Gambar 3.33 adalah desain detail pengecekan kerusakan fasilitas umum

3.5.4 Desain Interface Admin

Desain interface untuk manajemen data. Pada halaman ini admin bisa menginputkan data, merubah data dan menghapus data. Berikut ini adalah gambar rancangan interface pada halaman admin yang berfungsi untuk manajemen data.



Gambar 3.34 desain interface untuk manajemen data pada halaman admin

Selanjutnya adalah interface menambahkan data barang pada admin. Detail desain bisa dilihat pada gambar 3.34 berikut.



Gambar 3.35 desain interface menambahkan data pada halaman admin

Pada level admin terdapat fasilitas merubah data yang sudah tersimpan. Desain interface bisa dilihat pada gambar berikut.

Sistem Informasi Manajemen Kerusakan Fasilitas Umum

http://localhost/sim-KFU

SI MKFU: Master Pendataan Laporan Pengaturan User Log out

Edit Data Barang

Kode Barang: BRG-001

Nama Barang: LCD

Jumlah Barang: 63

Kode Sub Kelompok: SUB-02

Cancel Save Data

No	Kode barang	Nama Ba
1	BRG-001	LCD
2	BRG-002	AC
3	BRG-003	SPEAKER

Aksi: Edit Hapus

Gambar 3.36 desain interface untuk merubah data pada halaman admin

Pada level admin terdapat juga fasilitas untuk menghapus data yang sudah tersimpan. Desain interface bisa dilihat pada gambar berikut

Sistem Informasi Manajemen Kerusakan Fasilitas Umum

http://localhost/sim-KFU

SI MKFU: Master Pendataan Laporan Pengaturan User Log out

Delete Data Barang

Kode Barang: BRG-001

Nama Barang: LCD

Cancel Delete Data

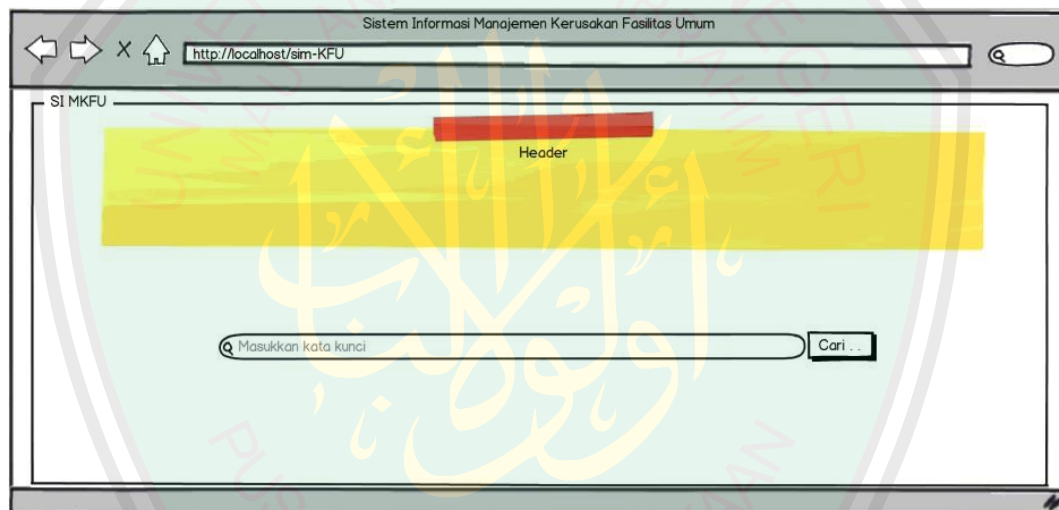
No	Kode barang	Nama Ba
1	BRG-001	LCD
2	BRG-002	AC
3	BRG-003	SPEAKER

Aksi: Edit Hapus

Gambar 3.37 desain interface menghapus data pada halaman admin

3.5.5 Desain *Interface* User

Desain interface untuk user. Pada halaman user nantinya hanya ada tombol pencarian seperti pada gambar 3.38, yang mana pencarian tersebut digunakan untuk mencari data kerusakan fasilitas umum baik yang masih dalam proses perbaikan atau yang sudah selesai. Pada aplikasi pencarian ini menggunakan teknologi semantic web dan algoritma rabin-karp untuk mengoptimalkan pencarian data laporan kerusakan fasilitas umum.



Gambar 3.38 desain interface untuk pencarian data kerusakan fasilitas umum

Setelah memasukkan kata kunci pencarian data laporan kerusakan fasilitas umum akan muncul sesuai dengan kata kunci yang dicari.

3.6 Kebutuhan Sistem

Untuk membangun sebuah aplikasi, tentunya akan mempunyai daftar kebutuhan dalam membangun sistem aplikasi tersebut. Pada tahapan ini akan membahas mengenai kebutuhan apa saja yang dibutuhkan dalam membangun

apilkasi sistem informasi kerusakan fasilitas umum dengan menggunakan teknologi *semantic web*. Kebutuhan tersebut akan diuraikan berdasarkan kebutuhan perangkat lunak (*Software*) serta kebutuhan perangkat keras (*Hardware*).

3.6.1 *Software*

Perangkat lunak (software) yang digunakan untuk mendukung dalam pembuatan atau pengoperasian semantic web sebagai berikut.

1. *Operating System* menggunakan Windows 8 ultimate 86x
2. JDK (*Java Development Kit*) versi 6.0_05
Software ini sebagai software pendukung untuk menjalankan server D2R.
3. D2R server, aplikasi yang berfungsi untuk menjembatani antara aplikasi PHP agar bisa mengakses data kerusakan fasilitas umum dari MySql DBMS (*Database Management System*) menggunakan *Sparql Query*.
4. Notepad++, sebagai *script editor*
5. MySql DBMS (*Database Management System*), digunakan untuk menyimpan data laporan kerusakan fasilitas umum.
6. Adobe Photoshop CS3, software untuk mengedit gambar
7. Balsamiq Mockups untuk mendesain tampilan website
8. Microsoft office visio 2007, software ini untuk mendesain flowchat
9. Mind Manager Pro 5 , software untuk mendesain alur sistem dan desain database

3.6.2 *Hardware*

Kebutuhan perangkat keras yang digunakan untuk membuat aplikasi semantic web menggunakan laptop dengan spesifikasi sebagai berikut.

- a. Processor AMD Dual-Core c60
- b. Memory RAM 4 GB
- c. Hardisk 500 GB



BAB IV

HASIL DAN PEMBAHASAN

Ujicoba yang dilakukan untuk menampilkan hasil dari penelitian ini adalah dengan menggunakan 2 buah server lokal. Server yang pertama digunakan untuk menjalankan aplikasi manajemen data dan servis D2R Server. Server kedua digunakan untuk menjalankan aplikasi pencarian. Server kedua akan mengakses server pertama untuk mendapatkan data laporan kerusakan fasilitas umum. Server kedua dapat menampilkan data sesuai dengan kata kunci yang diinputkan.

4.1 Aplikasi Manajemen Data Laporan

Sebelum membuat aplikasi untuk proses pencarian menggunakan teknologi *semantic web*, terlebih dahulu perlu membuat aplikasi manajemen data laporan yang akan digunakan sebagai proses penyimpanan, pengeditan dan untuk menghapus data laporan kerusakan fasilitas umum. Aplikasi manajemen data laporan difungsikan untuk menyimpan data ketika terdapat laporan kerusakan fasilitas umum yang masuk, laporan-laporan tersebut yang nantinya akan dicari dengan menggunakan teknologi *semantic web*. Untuk aplikasi manajemen sendiri terdapat 4 level user yaitu admin, ketua bagian, teknisi dan civitas. Berikut adalah penjelasan setiap level user.

4.1.1 Level Admin

Pada level ini, admin bertugas untuk melakukan input dan edit data master seperti data master kelompok barang, master sub kelompok barang, master gedung, master ruang, master barang dan master satuan barang. Admin juga bisa

menambahkan user pada setiap level. Berikut adalah tampilan awal ketika login sebagai admin:



Gambar 4.1 *interface* halaman awal ketika masuk sebagai admin

Admin hanya berfungsi untuk menyimpan, meng-*edit* dan menghapus data fasilitas umum saja bukan data laporan kerusakan fasilitas umum. Karna admin tidak mempunyai relasi / hubungan dengan proses pelaporan kerusakan fasilitas umum.

4.1.2 Level Ketua bagian

Pada level ini, ketua bagian mempunyai beberapa aktifitas. Pertama ketua bagian bisa menulis laporan langsung tanpa melalui laporan dari civitas terlebih dahulu, kedua ketua bagian juga bisa melihat laporan yang masuk dan memverifikasi laporan tersebut selanjutnya ketua bagian bisa memerintahkan teknisi untuk memeriksa dan memperbaiki laporan kerusakan fasilitas umum yang masuk. Berikut ini adalah tampilan halaman ketua bagian yang terdapat laporan masuk, pengecekan selesai dan perbaikan selesai.

SI MKFU Laporan Rekap Laporan Kerusakan Pengaturan Data User Logout

Sistem Informasi Manajemen Kerusakan Fasilitas Umum

Laporan Kerusakan

No	Nama	Jabatan	Tgl	Keterangan	Status	Aksi
1	M. aziz b.	karyawan bagian umum	13-11-2014	Kran rusak di toilet sebelah utara lantai 1 gedung c	pelaporan masuk	Peniksa Hapus
2	tsahyudi w	pegawai	08-11-2014	jaringan listrik gedung b-104 rusak	Proses perbaikan selesai	Detail Hapus
3	Fauzan Almanshur	pegawai	08-11-2014	Pintu toilet gedung b lantai 1 sebelah selatan rusak	Proses pengecekan	Hapus
4	Fauzan Almanshur	pegawai	08-11-2014	Pusat suara (istilamat) tidak dapat digunakan	Proses pengecekan	Hapus
5	Ikhamat basuki	pegawai	08-11-2014	Lampu gedung c lantai 1 dan Lampu kamar mandi putri sebelah selatan mati	Proses perbaikan selesai	Detail Hapus
6	Fauzan Almanshur	pegawai	08-11-2014	Kramik lantai 3 depan ruang biro akademik mumbul	Proses pengecekan	Hapus
7	Aris munandar	pegawai	08-11-2014	Memperbaiki pisiran slimar yang ambrol	Proses pengecekan	Hapus
8	Fauzan Almanshur	pegawai	08-11-2014	Saluran air kamar mandi gedung c lantai 2/3 tidak keluar	Proses perbaikan selesai	Detail Hapus
9	tnasih	dosen	08-11-2014	1 lantai kelas ruang B 301 pecah 2 lantai lorong depan kelas 307 pecah 3 atap ruang lab TU-RATS bocor 4 kamar mandi lantai 3 buntu 5 ruang di depan fakultas ambles	Proses pengecekan	Hapus
10	Fauzan Almanshur	pegawai	08-11-2014	Pembersihan kamar mandi rektorat lantai 1 s/d lantai 5	Proses pengecekan	Hapus
11	kusnadi	pegawai	08-11-2014	lantai 1 dan 2 fakultas ekonomi – gedung sosial rusak	Proses pengecekan	Hapus

Copy Right : Ahmad Farhan || 09600098

Gambar 4.2 interface sebagai ketua bagian

Pada gambar 4.2 bisa dilihat ada status laporan, terdapat empat status laporan yaitu pelaporan masuk, proses pengecekan, pengecekan selesai dan proses perbaikan selesai.

4.1.3 Level Teknisi

Yang dimaksud level teknisi adalah kordinator teknisi, dimana teknisi nantinya akan mendapat perintah pengecekan atau perintah untuk memperbaiki fasilitas umum yang rusak. Setelah tugas selesai teknisi akan memasukkan kebutuhan-kebutuhan yang diperlukan beserta perkiraan anggarannya dan juga

memasukkan teknisi yang bertugas memeriksa dan memperbaiki kerusakan fasilitas umum yang masuk. Berikut adalah tampilan ketika teknisi mendapatkan perintah pengecekan dari ketua bagian



Gambar 4.3 interface pada teknisi ketika mendapatkan perintah pengecekan

4.1.4 Level Civitas

Pada level ini, civitas bisa melaporkan kerusakan fasilitas umum yang terdapat di wilayah kampus UIN. Tidak semua civitas akademika bisa melaporkan kerusakan fasilitas umum, tetapi hanya civitas-civitas tertentu seperti dosen, musrif/ah dan pejabat kampus lainnya yang mendapatkan username dan password dari ketua bagian umum. Hal ini dilakukan untuk bisa mengkonrol laporan yang masuk agar tidak disalahgunakan oleh para mahasiswa yang tidak bertanggung jawab.

Berikut adalah tampilan untuk melaporkan kerusakan fasilitas umum yang ada dilingkungan kampus UIN Maliki Malang.

Gambar 4.4 interface pada civitas akademik

Untuk melaporkan kerusakan fasilitas umum, civitas akademik cukup menuliskan keluhan lalu mengirimkannya dengan menekan tombol kirim. Maka laporan tersebut akan terkirim kepada ketua bagian umum.

4.2 Aplikasi Pencarian

Aplikasi ini yang digunakan untuk melakukan proses pencarian data laporan kerusakan fasilitas umum menggunakan teknologi semantic web. Teknologi semantic web pada penelitian ini menggunakan pendekatan *turtle* (.ttl) sehingga membutuhkan D2R Server untuk *mapping* database dari database aplikasi manajemen data laporan. Aplikasi pencarian bisa mengakses data laporan

kerusakan fasilitas umum melalui D2R Server dengan menggunakan perintah SPARQL *Query* pada proses pencarian. Berikut adalah tahap implementasi semantic web pada aplikasi pencarian data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4.2.1 Membuat RDF Map dengan D2R Server

Pembuatan RDF Map bisa dilakukan dengan menggunakan *command* pada terminal. Adapun langkah-langkah pembuatan RDF Map sebagai berikut:

- a. Masuk pada directori D2R server
- b. Ketikkan “generate-mapping -o kerusakan.ttl -u root jdbc:mysql://localhost/kerusakan”. Lebih jelasnya bisa dilihat pada gambar berikut:



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\mbaos87>cd..
C:\Users>cd..
C:\>cd d2rq
C:\d2rq>generate-mapping -o kerusakan.ttl -u root jdbc:mysql://localhost/kerusak
n_
  
```

Gambar 4.5 Proses pembuatan RDF Map dengan command

Pada gambar 4.5 terdapat perintah “generate-mapping” yaitu perintah mapping data, lalu terdapat “-o kerusakan.ttl” adalah *output* dari mapping menggunakan

sintaks *turtle*. “-root jdbc:mysql://localhost/kerusakan” adalah pengambilan data dari database kerusakan. Berikut ini adalah RDF Map dari database kerusakan yang disimpan dalam format *turtle*.

```
@prefix map: <kerusakan.ttl#> .
@prefix db: <http://localhost:2020/resource/> .
@prefix vocab: <http://localhost:2020/resource/vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwi.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .

map:database a d2rq:Database;
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:jdbcDSN "jdbc:mysql://localhost/kerusakan";
    d2rq:username "root";
    jdbc:autoReconnect "true";
    jdbc:zeroDateTimeBehavior "convertToNull";
    .

# Table barang
map:barang a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "barang/@@barang.id_brg@@";
    d2rq:class vocab:barang;
    d2rq:classDefinitionLabel "barang";
    .

map:barang__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:barang;
    d2rq:property rdfs:label;
    d2rq:pattern "barang #@@barang.id_brg@@";
    .

map:barang_id_brg a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:barang;
    d2rq:property vocab:barang_id_brg;
    d2rq:propertyDefinitionLabel "barang id_brg";
    d2rq:column "barang.id_brg";
    d2rq:datatype xsd:integer;
    .

map:barang_nama_brg a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:barang;
    d2rq:property vocab:barang_nama_brg;
    d2rq:propertyDefinitionLabel "barang nama_brg";
    d2rq:column "barang.nama_brg";
```

```

.
map:barang_kode_brg a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:barang;
  d2rq:property vocab:barang_kode_brg;
  d2rq:propertyDefinitionLabel "barang kode_brg";
  d2rq:column "barang.kode_brg";
.

map:barang_jumlah a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:barang;
  d2rq:property vocab:barang_jumlah;
  d2rq:propertyDefinitionLabel "barang jumlah";
  d2rq:column "barang.jumlah";
  d2rq:datatype xsd:integer;
.

map:barang_id_skel__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:barang;
  d2rq:property vocab:barang_id_skel;
  d2rq:refersToClassMap map:sub_kelompok;
  d2rq:join "barang.id_skel => sub_kelompok.id_skel";
.

# Table gedung
map:gedung a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "gedung/@@gedung.id_gdg@";
  d2rq:class vocab:gedung;
  d2rq:classDefinitionLabel "gedung";
.

map:gedung__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:gedung;
  d2rq:property rdfs:label;
  d2rq:pattern "gedung #@@gedung.id_gdg@";
.

map:gedung_id_gdg a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:gedung;
  d2rq:property vocab:gedung_id_gdg;
  d2rq:propertyDefinitionLabel "gedung id_gdg";
  d2rq:column "gedung.id_gdg";
  d2rq:datatype xsd:integer;
.

map:gedung_kode_gdg a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:gedung;
  d2rq:property vocab:gedung_kode_gdg;
  d2rq:propertyDefinitionLabel "gedung kode_gdg";
  d2rq:column "gedung.kode_gdg";
.

map:gedung_nama_gdg a d2rq:PropertyBridge;

```

```

d2rq:belongsToClassMap map:gedung;
d2rq:property vocab:gedung_nama_gdg;
d2rq:propertyDefinitionLabel "gedung nama_gdg";
d2rq:column "gedung.nama_gdg";
.

```

Table kelompok

```

map:kelompok a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "kelompok/@@kelompok.id_kel@";
  d2rq:class vocab:kelompok;
  d2rq:classDefinitionLabel "kelompok";
.
map:kelompok__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kelompok;
  d2rq:property rdfs:label;
  d2rq:pattern "kelompok #@@kelompok.id_kel@";
.
map:kelompok_id_kel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kelompok;
  d2rq:property vocab:kelompok_id_kel;
  d2rq:propertyDefinitionLabel "kelompok id_kel";
  d2rq:column "kelompok.id_kel";
  d2rq:datatype xsd:integer;
.
map:kelompok_kode_kel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kelompok;
  d2rq:property vocab:kelompok_kode_kel;
  d2rq:propertyDefinitionLabel "kelompok kode_kel";
  d2rq:column "kelompok.kode_kel";
.
map:kelompok_nama_kel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kelompok;
  d2rq:property vocab:kelompok_nama_kel;
  d2rq:propertyDefinitionLabel "kelompok nama_kel";
  d2rq:column "kelompok.nama_kel";
.

```

Table kerusakan

```

map:kerusakan a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "kerusakan/@@kerusakan.id_rsk@";
  d2rq:class vocab:kerusakan;
  d2rq:classDefinitionLabel "kerusakan";
.
map:kerusakan__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kerusakan;

```



```

d2rq:property rdfs:label;
d2rq:pattern "kerusakan #@@kerusakan.id_rsk@@";
.
map:kerusakan_id_rsk a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_id_rsk;
d2rq:propertyDefinitionLabel "kerusakan id_rsk";
d2rq:column "kerusakan.id_rsk";
d2rq:datatype xsd:integer;
.
map:kerusakan_tgl a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_tgl;
d2rq:propertyDefinitionLabel "kerusakan tgl";
d2rq:column "kerusakan.tgl";
.
map:kerusakan_ket a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_ket;
d2rq:propertyDefinitionLabel "kerusakan ket";
d2rq:column "kerusakan.ket";
.
map:kerusakan_status a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_status;
d2rq:propertyDefinitionLabel "kerusakan status";
d2rq:column "kerusakan.status";
.
map:kerusakan_tgl_selesai a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_tgl_selesai;
d2rq:propertyDefinitionLabel "kerusakan tgl_selesai";
d2rq:column "kerusakan.tgl_selesai";
.
map:kerusakan_teknisi a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_teknisi;
d2rq:propertyDefinitionLabel "kerusakan teknisi";
d2rq:column "kerusakan.teknisi";
d2rq:datatype xsd:integer;
.
map:kerusakan_tgl_perbaikan a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:kerusakan;
d2rq:property vocab:kerusakan_tgl_perbaikan;
d2rq:propertyDefinitionLabel "kerusakan tgl_perbaikan";
d2rq:column "kerusakan.tgl_perbaikan";
.

```

```

map:kerusakan_id_sbrg__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kerusakan;
  d2rq:property vocab:kerusakan_id_sbrg;
  d2rq:refersToClassMap map:satuan_barang;
  d2rq:join "kerusakan.id_sbrg => satuan_barang.id_sbrg";
.

```

```

map:kerusakan_id_user__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:kerusakan;
  d2rq:property vocab:kerusakan_id_user;
  d2rq:refersToClassMap map:user;
  d2rq:join "kerusakan.id_user => user.id_user";
.

```

Table perbaikan

```

map:perbaikan a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "perbaikan/@@perbaikan.id_prb@@";
  d2rq:class vocab:perbaikan;
  d2rq:classDefinitionLabel "perbaikan";
.

```

```

map:perbaikan__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:perbaikan;
  d2rq:property rdfs:label;
  d2rq:pattern "perbaikan #@@perbaikan.id_prb@@";
.

```

```

map:perbaikan_id_prb a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:perbaikan;
  d2rq:property vocab:perbaikan_id_prb;
  d2rq:propertyDefinitionLabel "perbaikan id_prb";
  d2rq:column "perbaikan.id_prb";
  d2rq:datatype xsd:integer;
.

```

```

map:perbaikan_tgl_spk a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:perbaikan;
  d2rq:property vocab:perbaikan_tgl_spk;
  d2rq:propertyDefinitionLabel "perbaikan tgl_spk";
  d2rq:column "perbaikan.tgl_spk";
.

```

```

map:perbaikan_surat a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:perbaikan;
  d2rq:property vocab:perbaikan_surat;
  d2rq:propertyDefinitionLabel "perbaikan surat";
  d2rq:column "perbaikan.surat";
.

```

```

map:perbaikan_perintah a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:perbaikan;
  d2rq:property vocab:perbaikan_perintah;
.

```

```

    d2rq:propertyDefinitionLabel "perbaikan perintah";
    d2rq:column "perbaikan.perintah";
    .

map:perbaikan_teknisi a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:perbaikan;
    d2rq:property vocab:perbaikan_teknisi;
    d2rq:propertyDefinitionLabel "perbaikan teknisi";
    d2rq:column "perbaikan.teknisi";
    d2rq:datatype xsd:integer;
    .

map:perbaikan_kendala a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:perbaikan;
    d2rq:property vocab:perbaikan_kendala;
    d2rq:propertyDefinitionLabel "perbaikan kendala";
    d2rq:column "perbaikan.kendala";
    .

map:perbaikan_id_rsk__ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:perbaikan;
    d2rq:property vocab:perbaikan_id_rsk;
    d2rq:refersToClassMap map:kerusakan;
    d2rq:join "perbaikan.id_rsk => kerusakan.id_rsk";
    .

# Table petugas
map:petugas a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "petugas/@@petugas.id_ptgs@@";
    d2rq:class vocab:petugas;
    d2rq:classDefinitionLabel "petugas";
    .

map:petugas__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:petugas;
    d2rq:property rdfs:label;
    d2rq:pattern "petugas #@@petugas.id_ptgs@@";
    .

map:petugas_id_ptgs a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:petugas;
    d2rq:property vocab:petugas_id_ptgs;
    d2rq:propertyDefinitionLabel "petugas id_ptgs";
    d2rq:column "petugas.id_ptgs";
    d2rq:datatype xsd:integer;
    .

map:petugas_nama a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:petugas;
    d2rq:property vocab:petugas_nama;
    d2rq:propertyDefinitionLabel "petugas nama";
    d2rq:column "petugas.nama";

```

```

.
map:petugas_id_rsk__ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:petugas;
    d2rq:property vocab:petugas_id_rsk;
    d2rq:refersToClassMap map:kerusakan;
    d2rq:join "petugas.id_rsk => kerusakan.id_rsk";
.

# Table ruang
map:ruang a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "ruang/@@ruang.id_ruang@@";
    d2rq:class vocab:ruang;
    d2rq:classDefinitionLabel "ruang";
.
map:ruang__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:ruang;
    d2rq:property rdfs:label;
    d2rq:pattern "ruang #@@ruang.id_ruang@@";
.
map:ruang_id_ruang a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:ruang;
    d2rq:property vocab:ruang_id_ruang;
    d2rq:propertyDefinitionLabel "ruang id_ruang";
    d2rq:column "ruang.id_ruang";
    d2rq:datatype xsd:integer;
.
map:ruang_kode_ruang a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:ruang;
    d2rq:property vocab:ruang_kode_ruang;
    d2rq:propertyDefinitionLabel "ruang kode_ruang";
    d2rq:column "ruang.kode_ruang";
.
map:ruang_nama_ruang a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:ruang;
    d2rq:property vocab:ruang_nama_ruang;
    d2rq:propertyDefinitionLabel "ruang nama_ruang";
    d2rq:column "ruang.nama_ruang";
.
map:ruang_id_gdg__ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:ruang;
    d2rq:property vocab:ruang_id_gdg;
    d2rq:refersToClassMap map:gedung;
    d2rq:join "ruang.id_gdg => gedung.id_gdg";
.

# Table satuan_barang

```

```

map:satuan_barang a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "satuan_barang/@@satuan_barang.id_sbrg@@";
  d2rq:class vocab:satuan_barang;
  d2rq:classDefinitionLabel "satuan_barang";
.
map:satuan_barang__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property rdfs:label;
  d2rq:pattern "satuan_barang #@@satuan_barang.id_sbrg@@";
.
map:satuan_barang_id_sbrg a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_id_sbrg;
  d2rq:propertyDefinitionLabel "satuan_barang id_sbrg";
  d2rq:column "satuan_barang.id_sbrg";
  d2rq:datatype xsd:integer;
.
map:satuan_barang_merk a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_merk;
  d2rq:propertyDefinitionLabel "satuan_barang merk";
  d2rq:column "satuan_barang.merk";
.
map:satuan_barang_kondisi a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_kondisi;
  d2rq:propertyDefinitionLabel "satuan_barang kondisi";
  d2rq:column "satuan_barang.kondisi";
.
map:satuan_barang_nomor_brg a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_nomor_brg;
  d2rq:propertyDefinitionLabel "satuan_barang nomor_brg";
  d2rq:column "satuan_barang.nomor_brg";
.
map:satuan_barang_tahun_anggaran a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_tahun_anggaran;
  d2rq:propertyDefinitionLabel "satuan_barang tahun_anggaran";
  d2rq:column "satuan_barang.tahun_anggaran";
.
map:satuan_barang_id_ruang__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_id_ruang;
  d2rq:refersToClassMap map:ruang;
  d2rq:join "satuan_barang.id_ruang => ruang.id_ruang";

```

```

.
map:satuan_barang_id_brg__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:satuan_barang;
  d2rq:property vocab:satuan_barang_id_brg;
  d2rq:refersToClassMap map:barang;
  d2rq:join "satuan_barang.id_brg => barang.id_brg";
.

```

Table sub_kelompok

```

map:sub_kelompok a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "sub_kelompok/@@sub_kelompok.id_skel@@";
  d2rq:class vocab:sub_kelompok;
  d2rq:classDefinitionLabel "sub_kelompok";
.

```

```

map:sub_kelompok__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:sub_kelompok;
  d2rq:property rdfs:label;
  d2rq:pattern "sub_kelompok #@@sub_kelompok.id_skel@@";
.

```

```

map:sub_kelompok_id_skel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:sub_kelompok;
  d2rq:property vocab:sub_kelompok_id_skel;
  d2rq:propertyDefinitionLabel "sub_kelompok id_skel";
  d2rq:column "sub_kelompok.id_skel";
  d2rq:datatype xsd:integer;
.

```

```

map:sub_kelompok_kode_skel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:sub_kelompok;
  d2rq:property vocab:sub_kelompok_kode_skel;
  d2rq:propertyDefinitionLabel "sub_kelompok kode_skel";
  d2rq:column "sub_kelompok.kode_skel";
.

```

```

map:sub_kelompok_nama_skel a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:sub_kelompok;
  d2rq:property vocab:sub_kelompok_nama_skel;
  d2rq:propertyDefinitionLabel "sub_kelompok nama_skel";
  d2rq:column "sub_kelompok.nama_skel";
.

```

```

map:sub_kelompok_id_kel__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:sub_kelompok;
  d2rq:property vocab:sub_kelompok_id_kel;
  d2rq:refersToClassMap map:kelompok;
  d2rq:join "sub_kelompok.id_kel => kelompok.id_kel";
.

```

Table uraian

```

map:uraian a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "uraian/@@uraian.id_uraian@@";
    d2rq:class vocab:uraian;
    d2rq:classDefinitionLabel "uraian";
.
map:uraian__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property rdfs:label;
    d2rq:pattern "uraian #@@uraian.id_uraian@@";
.
map:uraian_id_uraian a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_id_uraian;
    d2rq:propertyDefinitionLabel "uraian id_uraian";
    d2rq:column "uraian.id_uraian";
    d2rq:datatype xsd:integer;
.
map:uraian_uraian a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_uraian;
    d2rq:propertyDefinitionLabel "uraian uraian";
    d2rq:column "uraian.uraian";
.
map:uraian_volume a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_volume;
    d2rq:propertyDefinitionLabel "uraian volume";
    d2rq:column "uraian.volume";
.
map:uraian_harga a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_harga;
    d2rq:propertyDefinitionLabel "uraian harga";
    d2rq:column "uraian.harga";
    d2rq:datatype xsd:integer;
.
map:uraian_qty a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_qty;
    d2rq:propertyDefinitionLabel "uraian qty";
    d2rq:column "uraian.qty";
    d2rq:datatype xsd:integer;
.
map:uraian_volume1 a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:uraian;
    d2rq:property vocab:uraian_volume1;

```

```

d2rq:propertyDefinitionLabel "uraian volume1";
d2rq:column "uraian.volume1";
.
map:uraian_harga1 a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:uraian;
d2rq:property vocab:uraian_harga1;
d2rq:propertyDefinitionLabel "uraian harga1";
d2rq:column "uraian.harga1";
d2rq:datatype xsd:integer;
.
map:uraian_qty1 a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:uraian;
d2rq:property vocab:uraian_qty1;
d2rq:propertyDefinitionLabel "uraian qty1";
d2rq:column "uraian.qty1";
d2rq:datatype xsd:integer;
.
map:uraian_keterangan a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:uraian;
d2rq:property vocab:uraian_keterangan;
d2rq:propertyDefinitionLabel "uraian keterangan";
d2rq:column "uraian.keterangan";
.
map:uraian_id_rsk_ref a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:uraian;
d2rq:property vocab:uraian_id_rsk;
d2rq:refersToClassMap map:kerusakan;
d2rq:join "uraian.id_rsk => kerusakan.id_rsk";
.

# Table user
map:user a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern "user/@@user.id_user@@";
d2rq:class vocab:user;
d2rq:classDefinitionLabel "user";
.
map:user__label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:user;
d2rq:property rdfs:label;
d2rq:pattern "user #@@user.id_user@@";
.
map:user_id_user a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:user;
d2rq:property vocab:user_id_user;
d2rq:propertyDefinitionLabel "user id_user";
d2rq:column "user.id_user";

```



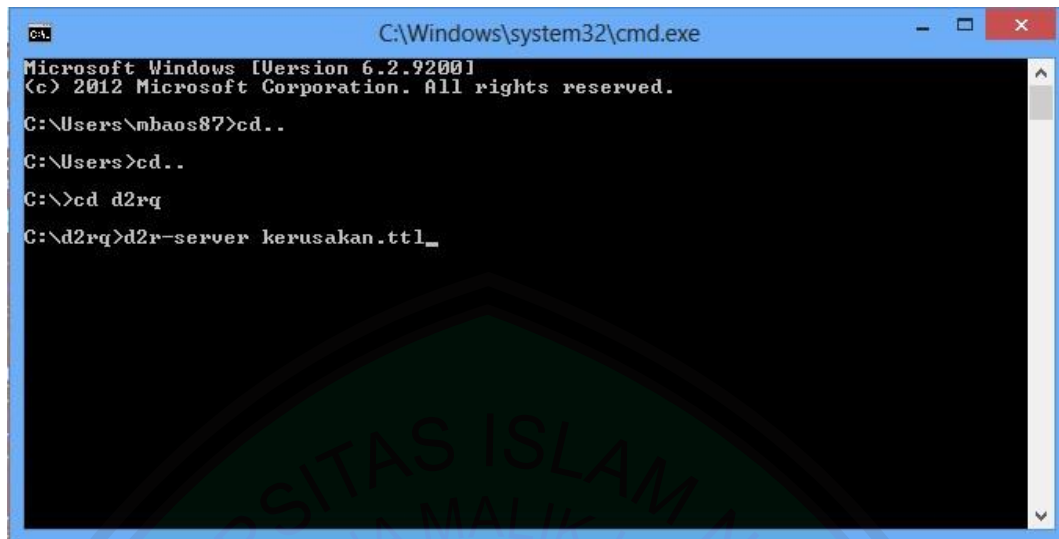
```

d2rq:datatype xsd:integer;
.
map:user_nama_user a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:user;
  d2rq:property vocab:user_nama_user;
  d2rq:propertyDefinitionLabel "user nama_user";
  d2rq:column "user.nama_user";
.
map:user_level a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:user;
  d2rq:property vocab:user_level;
  d2rq:propertyDefinitionLabel "user level";
  d2rq:column "user.level";
.
map:user_password a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:user;
  d2rq:property vocab:user_password;
  d2rq:propertyDefinitionLabel "user password";
  d2rq:column "user.password";
.
map:user_jabatan a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:user;
  d2rq:property vocab:user_jabatan;
  d2rq:propertyDefinitionLabel "user jabatan";
  d2rq:column "user.jabatan";
.

```

4.2.2 Menjalankan D2R Server

Servis D2R server akan dijalankan dengan menggunakan RDF Map, file RDF map yang digunakan bernama kerusakan.ttl dengan port standar 2020. Gambar 4.6 adalah proses start servis D2R Server yang dijalankan menggunakan *command* pada terminal



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\mbaos87>cd..
C:\Users>cd..
C:\>cd d2rq
C:\d2rq>d2r-server kerusakan.ttl_

```

Gambar 4.6 cara menjalankan RDF Map dengan command

Sampai muncul tulisan server started, berarti server D2R sudah berjalan. Berikut adalah tampilan ketika RDF Map berhasil dijalankan.



```

C:\Windows\system32\cmd.exe - d2r-server kerusakan.ttl
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\mbaos87>cd..
C:\Users>cd..
C:\>cd d2rq
C:\d2rq>d2r-server kerusakan.ttl
11:32:31 INFO JettyLauncher :: [!! Server started at http://localhost:20
20/ 111

```

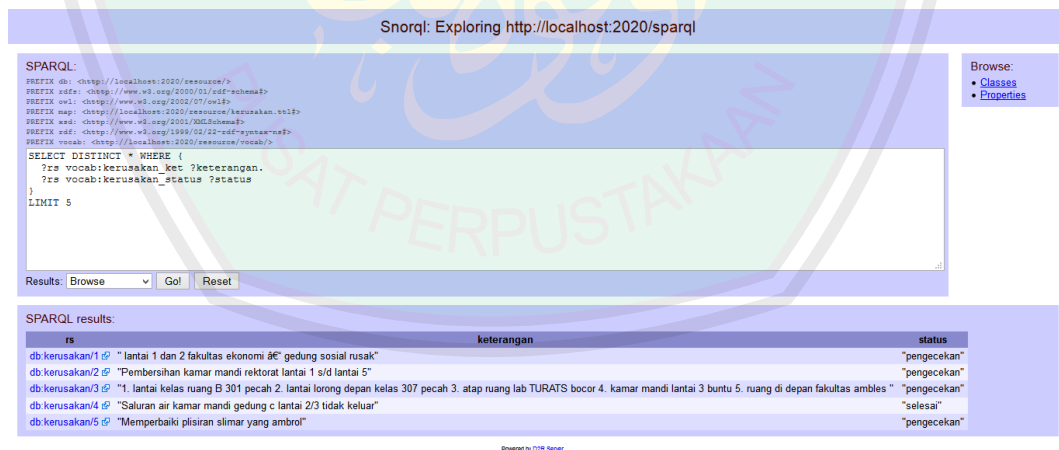
Gambar 4.7 RDF Map ketika berhasil dijalankan

Pada gambar 4.7 penjelasan bahwa RDF Map berjalan pada alamat <http://localhost:2020>. Berikut adalah tampilan D2R Server yang sudah berhasil dengan menggunakan RDF Map yaitu `kersakan.ttl`



Gambar 4.8 tampilan D2R Server pada browser

Gambar 4.8 adalah tampilan D2R server, pada D2R Server tabel-tabel database yang degenerate menjadi RDF Map akan berubah menjadi menu di D2R server. Menu-menu tersebut berisikan data-data yang ada pada tiap tabel dari database. Berikut adalah tampilan SPARQL Endpoint Pada Gambar 4.9



Gambar 4.9 tampilan SPARQL Endpoint

Pada gambar 4.9 adalah tampilan SPARQL Endpoint, pada halaman ini digunakan untuk mengecek query sparql apakah query tersebut benar atau salah dan bisa mengakses data atau tidak.

4.3 Proses Pencarian dengan Semantic Web

Tahapan melakukan pencarian yaitu dengan cara mengetikkan kata yang akan dicari pada *form* pencarian. Desain halaman pencarian seperti pada gambar 4.10



Gambar 4.10 tampilan pencarian data

Teknologi semantic web akan dijalankan pada aplikasi pencarian. Pada aplikasi ini data yang dicari tidak langsung diambil dari database MySQL, melainkan mengambil data yang sudah tersimpan pada web semantic itu sendiri, yaitu disimpan pada D2R server. Jadi ketika mengakses data tersebut tidak dibutuhkan user atau password database MySQL. Berikut adalah potongan *sourcecode* untuk mengakses data laporan kerusakan fasilitas umum melalui D2R server.

```
PREFIX db: <http://localhost:2020/resource/>
PREFIX map: <http://localhost:2020/resource/kerusakan.ttl#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>

SELECT * where {
    ?rs vocab:kerusakan_ket ?keterangan.
    ?rs vocab:kerusakan_status ?status

    filter ($pencarian && regex(?status,'$kunci1','i'))
}
```

Sourcode tersebut untuk mengakses D2R server dan mengambil data dari tabel kerusakan pada field keterangan laporan dan status laporan yang terdapat pada database melalui RDF Map.

Ketika proses pencarian selesai maka akan dihasilkan data sesuai dengan kata kunci yang diinputkan. Tampilan hasil dari pencarian menggunakan semantic search bisa dilihat pada gambar 4.11



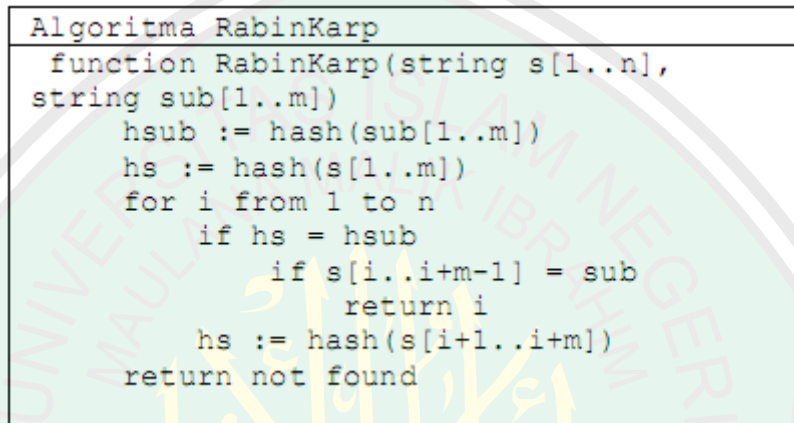
Gambar 4.11 tampilan hasil dari pencarian data

Pada gambar 4.11 adalah hasil dari pencarian menggunakan semantic search, dimana kata yang berwarna merah adalah kata kunci yang dicari. Berikut adalah sourcode untuk menampilkan hasil data yang dicari.

```
echo"
<table class='table table-bordered table-striped' align = 'center' border='5'>
    <tr>
        <th>No</th>
        <th>Keterangan Laporan</th>
        <th>Status</th>
    <tr>
        ";
```

4.3.1 Penerapan Algoritma Rabin-Karp

Pada aplikasi pencarian memanfaatkan Algoritma rabin-karp untuk membantu memaksimalkan pencarian data laporan kerusakan fasilitas umum. Berikut adalah Pseudocode untuk proses pencocokan string dengan Rabin-karp function.



```

Algoritma RabinKarp
function RabinKarp(string s[1..n],
string sub[1..m])
    hsub := hash(sub[1..m])
    hs := hash(s[1..m])
    for i from 1 to n
        if hs = hsub
            if s[i..i+m-1] = sub
                return i
            hs := hash(s[i+1..i+m])
    return not found
  
```

Gambar 4.12 Pseudocode algoritma rabin-karp

Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n). Apabila *hash value* keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila hasil keduanya tidak sama, maka substring akan bergeser ke kanan. Pergeseran dilakukan sebanyak (n-m) kali.

Berikut ini adalah alur pencarian menggunakan algoritma rabin-karp secara manual. Pada pencarian ini menggunakan *text* inputan “Bocor” dan *text String*nya adalah “Atap Bocor”

1. Menghilangkan tanda baca dan spasi mengubah kata yang ingin dicari menjadi kata-kata tanpa huruf kapital.
 - “Atap Bocor” menjadi "atapbocor"
2. Membagi teks ke dalam gram-gram yang ditentukan nilai k-gram nya.
 - “atapbocor” dibagi menjadi gram-gram yang sudah ditentukan, karna text inputan adalah “bocor” yang mempunyai gram 5 maka ditentukan gramnya adalah 5, maka dihasilkan
 - “atapb” , ”tapbo” , ”apboc” , “pboco” , “bocor”
3. Mencari nilai hash dengan fungsi rolling hash dari tiap gram yang terbentuk. Untuk mencari nilai hash menggunakan American Standard Code for Information Interchange (ASCII) sehingga nilai dari karakter sudah ditentukan. Berikut adalah nilai hash dari tiap gram.
 - atapb = 97, 116, 97, 112, 98 menghasilkan nilai hash 12296
 - tapbo = 116, 97, 112, 98, 111 menghasilkan nilai hash 13428
 - apboc = 97, 112, 89, 111, 99 menghasilkan nilai hash 12195
 - pboco = 112, 98, 111, 99, 111 menghasilkan nilai hash 13449
 - bocor = 89, 111, 99, 111, 114 menghasilkan nilai hash 12273

untuk mendapatkan nilai hash menggunakan rumus

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b + c_k$$

- atapb = $97 * 3^4 + 116 * 3^3 + 97 * 3^2 + 112 * 3^1 + 98 * 3^0 = 12296$
- tapbo = $116 * 3^4 + 97 * 3^3 + 112 * 3^2 + 98 * 3^1 + 111 * 3^0 = 13428$
- apboc = $110 * 3^4 + 97 * 3^3 + 109 * 3^2 + 97 * 3^1 + 115 * 3^0 = 12916$
- pboco = $110 * 3^4 + 97 * 3^3 + 109 * 3^2 + 97 * 3^1 + 115 * 3^0 = 12916$
- bocor = $110 * 3^4 + 97 * 3^3 + 109 * 3^2 + 97 * 3^1 + 115 * 3^0 = 12916$

4. Mencari nilai hash yang sama antara 2 teks.

Text pertama adalah bocor mempunyai nilai hash 12273. Text kedua adalah atapbocor dibagi menjadi 5 gram yaitu “atapb”, ”tapbo”, ”apboc”, “pboco”, “bocor” masing-masing mempunyai nilai hash “12296”, “13428”, “12195”, “13449”, “12273” sehingga proses pencocokan antara text pertama dan text kedua adalah.

1	12273				
	12296	13428	12195	13449	12273
2		12273			
	12296	13428	12195	13449	12273
3			12273		
	12296	13428	12195	13449	12273
4				12273	
	12296	13428	12195	13449	12273
5					12273
	12296	13428	12195	13449	12273

Tabel 4.1 percocokan nilai hash

Percocokan nilai hash ditemukan kesamaan pada rolling yang kelima.

4.4 Uji Coba Sistem

Ujicoba dalam penelitian ini dilakukan untuk membuktikan bahwa dengan menerapkan teknologi semantic web, dokumentasi laporan kerusakan fasilitas umum bisa diakses dengan menggunakan kata kunci yang diambil dari kata yang berhubungan dengan tempat kerusakan, jenis kerusakan, barang yang rusak dan status pengerjaan baik yang sudah selesai dikerjakan ataupun yang masih dalam proses pengerjaan. Hasil ujicoba diambil dari hasil pencarian dengan menggunakan beberapa kata kunci hasil pencarian akan dicek satu persatu untuk

mengetahui atas dasar apa data tersebut muncul dalam pencarian. Tabel 4.2 merupakan data hasil ujicoba dengan menggunakan 10 kata kunci yang berbeda.

Tabel 4.2 Hasil Ujicoba

No	Kata kunci	Kategori					Total
		Selesai	Pengecekan	Uraian	Perbaikan	Laporan	
1	lantai	4	4	0	1	13	22
2	gedung	4	1	1	1	3	10
3	kelas	0	1	0	0	0	1
4	toilet	2	0	0	0	1	3
5	kamar	2	2	0	0	14	18
6	ruang	0	2	0	0	2	4
7	lampu	1	0	0	0	1	2
8	listrik	0	0	1	0	1	2
9	keramik	1	0	0	0	10	11
10	air	1	0	0	0	2	3

*ujicoba ini dilakukan pada tanggal 14 november 2014

Tabel 4.2 adalah tabel dari hasil pencaian data sesuai dengan kata kunci yang dimasukkan. Data yang dihasilkan dari pencarian sesuai dengan kata kunci yang diinputkan yang datanya terdapat kata yang sama dengan kata kunci yang diinputkan.

4.5 Integrasi Dalam Islam

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهْلَةٍ فَتُصْحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ ﴿٦﴾

“Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah (kebenarannya) dengan teliti, agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu”. (Al-Hujurat: 6)

Turunnya ayat ini untuk mengajarkan kepada kaum muslimin agar berhati-hati dalam menerima berita dan informasi. Keputusan yang salah akan menyebabkan semua pihak merasa menyesal. Pihak pembuat keputusan merasa menyesal karena keputusannya itu menyebabkan dirinya mendhalimi orang lain. Pihak yang menjadi korban pun tak kalah sengsaranya mendapatkan perlakuan yang dhalim. Maka jika ada informasi yang berasal dari seseorang yang integritas kepribadiannya diragukan harus diperiksa terlebih dahulu.

Intergrasi surat Al-Hujurat ayat 6 dengan penelitian yang dilakukan yaitu membuat aplikasi pencarian menggunakan teknologi semantic web dan algoritma rabin-karp adalah: aplikasi pencarian digunakan untuk mencari data laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Ketepatan dalam pencarian dan kebenaran data sangat penting untuk mendapatkan data laporan yang diinginkan tentunya data tersebut harus benar adanya. Ketika data dalam laporan benar tetapi dalam proses pencariannya tidak maksimal, maka akan mendapatkan data yang tidak maksimal juga. Sebaliknya, ketika dalam proses pencariannya maksimal tetapi data yang ada tidak benar maka akan mendapatkan data yang tidak benar juga. Untuk mendapatkan data pencarian laporan kerusakan fasilitas umum yang ada di Universitas Islam Negeri Maulana Malik Ibrahim Malang yang benar, maka harus didukung dengan data yang benar dan proses pencarian yang maksimal. Pada pembuatan pencarian data laporan kerusakan fasilitas umum menggunakan teknologi semantic web dan algoritma rabin-karp untuk mengoptimalkan pencarian data diharapkan data yang tampil dari hasil pencarian adalah data yang

benar. Penjelasan tentang meneliti kebenaran suatu berita juga dijelaskan oleh Ibnu Katsir. beliau menafsirkan surat At-Taubah ayat 96.

يَحْلِفُونَ لَكُمْ لِتَرْضَوْا عَنْهُمْ فَإِنْ تَرْضَوْا عَنْهُمْ فَإِنَّ اللَّهَ لَا يَرْضَىٰ عَنِ
الْقَوْمِ الْفَاسِقِينَ ﴿٩٦﴾

Artinya: Mereka akan bersumpah kepadamu, agar kamu ridha kepada mereka. Tetapi jika sekiranya kamu ridha kepada mereka, sesungguhnya Allah tidak ridha kepada orang-orang yang fasik itu. Mereka akan bersumpah kepadamu, agar kamu ridha kepada mereka. Tetapi jika sekiranya kamu ridha kepada mereka, sesungguhnya Allah tidak ridha kepada orang-orang yang fasik itu.

Ibnu katsir menafsirkan surat At-Taubah ayat 96 sebagai berikut “maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu” : bahwasanya adab dalam menerima berita adalah dengan tabayyun yaitu klarifikasi atau cek and recek atas berita tersebut agar adanya kejelasan berita dan keakuratan kebenarannya, sebab warta dan fakta terkadang berbeda.

Syaikh Muhammad Ali Ash-Shabuni menjelaskan bahwa ayat ini adalah suatu keharusan akan pengecekan suatu berita, dan juga keharaman akan berpegang kepada berita orang-orang yang fasik yang banyak menimbulkan bahaya. Ayat ini mengajarkan bahwa mencari kebenaran berita serta tidak mempercayai berita yang dibawa oleh orang yang fasik yang menentang Allah adalah suatu keharusan.

Sebab berpegang kepada berita yang belum jelas kebenarannya, terlebih berita yang disebarkan oleh orang fasik ini membahayakan dari dua sisi. Yaitu, Sisi dari sumber berita dan jenis berita, berita yang dibawa oleh orang fasik berkemungkinan adalah berita yang munkar oleh sebab kedengkian dan kejelekan sikap yang ada pada dirinya. Dan juga jenis berita yang dibawa oleh orang fasik biasanya juga berjenis berita yang munkar.



BAB V

PENUTUP

5.1 Kesimpulan

Terdapat beberapa kesimpulan yang dapat diambil dari penelitian ini diantaranya:

1. Penggunaan semantic web pada aplikasi pencarian data laporan kerusakan fasilitas umum dilakukan dengan menggunakan RDF Map dan D2RQ Server
2. Untuk mengakses data dari D2R Server menggunakan *Sparql Query* yang langsung mengakses data Endpoint D2R Server.
3. Untuk pengoptimalan pencarian aplikasi ini memanfaatkan algoritma rabin-karp untuk percocokan string

5.2 Saran

Penelitian ini merupakan penelitian yang cukup berpotensi untuk terus dikembangkan. Beberapa saran dari peneliti untuk pengembangan selanjutnya yaitu;

1. Dapat dibuat konsep dan relasi yang lebih komplek, tidak hanya data laporan kerusakan fasilitas umum tetapi bisa dikembangkan dengan data-data dari bagian keuangan dan bagian pengadaan.
2. Aplikasi ini dapat dikembangkan versi mobile dengan tetap memanfaatkan teknologi semantic web.
3. Untuk tampilan pencarian data dapat ditambahkan detail setiap data dan mengganti tampilan dengan *responsive template*.

DAFTAR PUSTAKA

- Andres, Nicolas dkk. *Penelaahan Algoritma Rabin-Karp dan Perbandingan Prosesnya dengan Algoritma Knut-Morris-Pratt*. Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, Institut Teknologi Bandung Jl. Ganesha 10, Bandung
- Abul Fida' 'Imaduddin Ismail. 2003. *Tafsir Ibnu Katsir*. Jakarta: Pustaka Imam Syafi'i.
- Ause, Wayne. 1995. *Instant HTML Web Pages*. Ziff-Dafis Press
- Berners-Lee, T., Hendler, J., Lasilla, O., 2001, *The Semantic Web*, American Scientific
- Bizer, c., & Cyganiak, R., 2006. *D2rq Server-Publishing Relational Databases on Semantic Web*. ISWC.
- Cyganiak, R., bizer, C., Garber, J., Maresch, O., & Christian, B. 2012. *The D2RQ Mapping Language*. <http://d2rq.org> diakses tanggal 22 juni 2014
- Dodds, Leigh. 2005. *Introducing SPARQL: Querying the Semantic Web*. (<http://www.xml.com/lpt/a/2005/11/16/introducing-sparql-querying-semantic-web-tutorial.html>) Akses website tanggal 21 agustus 2014
- Winarno, Edy, Zaki, Ali, Smit. 2011. *Mudah Membuat Website dan Ecommerce Dengan PHP Framework*, PT Elex Media Komputindo. jakarta
- Fadillah, Nava'atul dkk. 2010. *Penerapan Teknologi Semantic Web pada Aplikasi Pencarian Koleksi Perpustakaan (Studi Kasus: Perpustakaan FTI UPN "Veteran" Yogyakarta*. Yogyakarta

- Jeffrey L. Whitten., Lonnie D. Bentley., Kevin C. Dittman. 2004. Metode Desain dan Analisis Sistem Edisi 6. Yogyakarta: Andi.
- Hebeler John., Fisher Matthew., Blace Ryan., Perez-Lopez Andrew. 2009. Semantic Web Programming. Canada. Wiley Publishing.
- Hendler Jim., Allemang Dean. 2011. Semantic Web for the Working Ontologist. 255 Wyman Street. Waltham. USA.
- Keputusan Inspektur Jenderal Departemen Agama Nomor: ij/84/2000. Tentang Tata Cara Pendataan Dan Pengelolaan Inventaris Barang Milik Negara
- Peraturan Pemerintah Nomor 6 Tahun 2006 Jo. Pp Nomor 38 Tahun 2008 Tentang Pengelolaan Barang Milik Negara/Daerah
- Nugroho, Andi. 2013. *Ensiklopedia Objek Wisata Jawa Timur Menggunakan Semantic Web*. Thesis Jurusan Teknik Informatika, UIN Maliki Malang.
- Nugroho, Bunafit. 2004. Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL. Yogyakarta : Gava Media.
- Rahutomo, Faisal. 2009. *Penerapan Agoritma Weighted Tree Similary untuk Pencarian Semantik Wikipedia*. Master Thesis Jurusan Teknik Informatika, Surabaya.
- Sarno, Riyanarto dkk. 2012. *Semantic Search-Pencarian Berdasarkan Konten*. Yogyakarta: ANDI
- <https://girlycious09.wordpress.com/tag/definisi-laporan/> diakses pada tanggal 25 juni 2014