# APLIKASI *USER INTERFACE* (UI) *DISCOVERY* UNTUK MENGUKUR AKURASI *QUERY* PADA *INTERFACE REPOSITORY*

**SKRIPSI**

**Oleh :**
**SHOFIYATUN NAJAH**
**NIM. 13650002**

**JURUSAN TEKNIK INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**
**MALANG**
**2018**

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# APLIKASI *USER INTERFACE* (UI) *DISCOVERY* UNTUK MENGUKUR AKURASI *QUERY* PADA *INTERFACE REPOSITORY*

**SKRIPSI**

**Diajukan kepada:**
**Universitas Islam Negeri Maulana Malik Ibrahim Malang**
**Untuk memenuhi Salah Satu Persyaratan dalam**
**Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:**
**SHOFIYATUN NAJAH**
**NIM. 13650002**

**JURUSAN TEKNIK INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**
**MALANG**
**2018**



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# LEMBAR PERSETUJUAN

**LEMBAR PERSETUJUAN**

**APLIKASI *USER INTERFACE* (UI) *DISCOVERY* UNTUK MENGUKUR AKURASI *QUERY* PADA *INTERFACE REPOSITORY***

**SKRIPSI**

**Oleh :**

**Shofiyatun Najah**
**NIM. 13650002**

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: Mei 2018

| Pembimbing I, | Pembimbing II, |
|---|---|
| M. Ainul Yaqin, M.Kom | Linda Salma Angreani, M.T |
| NIP. 19761013 200604 1 004 | NIP. 19770803 200912 2 005 |

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008





CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# LEMBAR PENGESAHAN

**LEMBAR PENGESAHAN**

**APLIKASI *USER INTERFACE* (UI) *DISCOVERY* UNTUK MENGUKUR AKURASI *QUERY* PADA *INTERFACE REPOSITORY***

**SKRIPSI**

**Oleh :**
**Shofiyatun Najah**
**NIM. 13650002**

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : Juni 2018

| Susunan Dewan Penguji | | Tanda Tangan |
|---|---|---|
| Penguji Utama | : Supriyono, M.Kom<br>NIDT. 19841010 20160801 1 078 | ( ) |
| Ketua Penguji | : Syahiduz Zaman, M.Kom<br>NIP. 19700502 200501 1 005 | ( ) |
| Sekretaris Penguji | : M. Ainul Yaqin, M.Kom<br>NIP. 19761013 200604 1 004 | ( ) |
| Anggota Penguji | : Linda Salma Angreani, M.T<br>NIP. 19770803 200912 2 005 | ( ) |

Mengesahkan,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008



iii

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# HALAMAN PERSEMBAHAN

اَلْحَمْدُ لِلّٰهِ رَبِّ الْعٰلَمِيْنَ

Segala puji syukur kehadirat Allah SWT atas karunia kehidupan dan ilmu pengetahuan yang telah dilimpahkan dan ridlo-Nya sehingga dapat terselesaikannya penulisan karya ini. Sholawat serta salam kepada Nabi Muhammad SAW, yang syafaatnya diharapkan di hari akhir.

**Ayahku, H. Ahmad Shohib (Alm)**, terimakasih atas segala do'a dan motivasi untukku. Terimakasih untuk ayah yang selalu memberi semangat ketika pulang ke rumah. Ayah yang tak pernah kenal lelah dalam hal apapun, ayah yang selalu mendukung segala hal demi terwujudnya sebuah harapan. Terimakasih atas segala yang ayah berikan sampai akhir hayatnya. Meskipun ini terlambat, harapanku saat ini ialah melihat ayah tersenyum disana dengan terselesaikan karya ini. Terimakasih untuk ayah yang selalu menguatkanku dengan cara apapun. Hal itulah yang membuatku yakin bahwa semua akan indah pada waktunya dan rencana Allah lebih indah dari segalanya.

**Ibuku, Hj. Nafi'ah**, terimakasih atas segala do'a dan kasih sayang yang tulus untukku. Terimakasih untuk ibu yang selalu memberi motivasi, ibu yang selalu menenangkan hati anaknya ketika ada masalah. Terimakasih atas kesabaran ibu dalam menghadapi segala hal. Ibu yang selalu menujukkan arti sabar, dan memerintahkan sabar dalam menyelesaikan karya ini. Terimakasih untuk ibu yang selalu memberi dukungan apapun.

**Nengku tersayang, Ainur Rohmawati**, terimakasih atas do'a dan motivasi untuk adik, neng yang selalu membangkitkan ketika adik terjatuh, dan neng yang selalu merangkul adiknya dalam hal apapun.

**Mas Hafid, Weni, Shinta, Aniek, dan teman Kos 49 (Dian, Sulva, Dewi, dan Mbak Reni)** terimakasih atas do'a, semangat, dan nasihat yang telah memberikan kekuatan dalam menyelesaikan skripsi.

Terimakasih untuk semua pihak yang telah membantu dan memberi do'a dan semangat dalam menyelesaikan penulisan skripsi.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# PERNYATAAN KEASLIAN

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Shofiyatun Najah

NIM : 13650002

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

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, 27 April 2018

Yang membuat pernyataan


Shofiyatun Najah

NIM. 13650002



v

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# KATA PENGANTAR

*Assalamu'alaikum Wr.Wb.*

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju islam yang *rahmatan lil alamiin.*

Dalam menyelesaikan skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada:

1. Bapak Prof. Dr. Abdul Haris, M.Ag, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Ibu Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak M. Ainul Yaqin, M.Kom, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, dan mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Ibu Linda Salma Angreani, MT, selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

6. Seluruh dosen dan staff jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
7. Ayah, ibu, dan kakak tercinta, serta seluruh keluarga besar yang senantiasa memberikan do'a dan motivasi kapada penulis dalam menuntut ilmu serta do'a yang senantiasa mengiringi setiap langkah penulis.
8. Teman-teman Kos 49 (Mbak Reni, Dian, Sulva dan Dewi) yang senantiasa membantu dan memberi semangat penulis selama di kos.
9. Mas Hafid, Mas Aang, Abdan, Ayom, Shinta, Weni, dan Aniek yang telah membantu dan memberi semangat dalam mengerjakan skripsi.
10. Tim Skripsi Sukses yang telah berjuang bersama dan memberi bantuan dalam menyelesaikan penulisan.
11. Teman-teman seperjuangan Teknik Informatika angkatan 2013.
12. Semua pihak yang ikut dalam berkontribusi dalam membantu menyelesaikan skripsi.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Amin.

*Wassalamualaikum Wr.Wb*

Malang, 27 April 2018

Penulis



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

**MOTTO**

“Berotak London
Berhati Masjidil Haram”



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# DAFTAR ISI





CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# DAFTAR GAMBAR





CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# DAFTAR TABEL





CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# ABSTRAK

Najah, Shofiyatun. 2018. **Aplikasi User Interface (UI) Discovery untuk Mengukur Akurasi Query pada interface Repository**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing (I) M. Ainul Yaqin, M. Kom, (II) Linda Salma Angreani, M.T

**Kata Kunci :** *User Interface* (UI), *Discovery*, akurasi

SaaS (*Software as a Service* atau perangkat lunak berbentuk *service*) adalah suatu model penyampaian aplikasi perangkat lunak oleh suatu vendor perangkat lunak yang mengembangkan aplikasi *web* yang dioperasikan untuk digunakan oleh pengguna melalui internet. Aplikasi SaaS memiliki beberapa lapisan arsitektur yaitu data *layer*, *service layer*, *process layer*, dan UI *layer*.UI *layer* yaitu merupakan lapisan GUI yang berfungsi untuk menyediakan *interface* antara sistem dan *user* dalam menerima *input* dari *user* dan menampilkan hasil kembali kepada *user*.

Penelitian ini menjelaskan tentang bagaimana membangun perangkat lunak berbasis SaaS dan diimplementasikan pada aplikasi *user interface discovery*. Aplikasi tersebut bertujuan untuk memudahkan *user* dalam mengubah dan mengkonfigurasi tampilannya, termasuk menambahkan, mengubah, menghapus ikon, warna, *font*, judul, dan menu yang ada didalam aplikasi yang akan dibuat oleh user. Pencarian UI yang diinginkan oleh user dilakukan dengan memasukkan kata kunci yang mengandung nama form, nama komponen, dan jumlah komponen yang diinginkan. Metode yang digunakan pada aplikasi *user interface discovery* adalah semantik web. Proses yang dilakukan yaitu menggunakan RDF *server* dan *query* SPARQL sebagai alat untuk melakukan *discovery*.

Berdasarkan perhitungan uji coba yang telah dilakukan dengan dua model pencarian yakni pencarian data dan pencarian detail memperoleh *recall* 95% dan akurasi 100%, namun presisi pada pencarian data didapatkan nilai 95% dan pada pencarian detail diperoleh nilai 90%. Perbedaan tersebut diperoleh karena pada pencarian detail ada beberapa dokumen tidak relevan yang ditemukan dan terjaring *query*, sehingga pada pencarian data menunjukkan hasil yang lebih relevan dan menunjukkan kemampuan sistem lebih baik.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# ABSTRACT

Najah, Shofiyatun. 2018. ***User Interface* (UI) *Discovery Application To Measure Query Accuracy On Interface Repository***. *Undergraduate Thesis. Informatics Engineering Department. Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang.*

*Adviser* (I) M. Ainul Yaqin, M. Kom, (II) Linda Salma Angreani, M.T

***Keywords**: user interface (UI), discovery, accuracy*

*SaaS (Software as a Service) is a software delivery application model by a software vendor which develops web application and operates using internet by the user. SaaS application has many architecture layers; data layer, service layer, process layer, and UI layer. UI layer is GUI layer which has function for preparing interface between system and user to receive input and user and show the result to the user.*

*This research explains about how to build software SaaS-based and the implementation to user interface discovery application. This application is used to change data easily and display configuration by the user, included adding, changing, deleting the icon, color, font, title, and another menu which will be made by user. The needed UI search is by entering the keyword which consists of form name, component name, and quantity of component by user. The method which is used on user interface discovery application is web semantic. The process is using RDF server and query SPARQL as the tools for discovery.*

*Based on the calculation of trial that has been done with two search model that is searching data and searching detail get 95% recall and 100% accuracy, but precision in data search get value 95% and in detail search obtained value 90%. The difference is obtained because in the detail search there are some irrelevant documents found and netted queries, so the search data shows more relevant results and indicate the ability of the system better.*



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# ملخص

النجة,صفية.2018. تطبيق لقياس دقة الاستعلام في واجهة مستودع التخزين.بحث جامعي. قسم الهندسة المعلوماتية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.

المشرف الأول :محمد عين اليقين الماجستير, المشرف الثاني: ليندا سالما انجرياني الماجستير

---

**الكلمة**: واجهة المستخدم، اكتشاف، دقة

SaaS (برنامج كخدمة أو برنامج في شكل خدمة) هو نموذج لتقديم تطبيقات البرامج بواسطة مورّد برامج يقوم بتطوير تطبيقات الويب التي يتم تشغيلها للاستخدام من قبل المستخدمين عبر الإنترنت. يحتوي تطبيق SaaS على عدة طبقات من البنية ، مثل طبقة البيانات ، وطبقة الخدمة ، وطبقة العملية ، وطبقة واجهة المستخدم. طبقة واجهة المستخدم هي طبقة واجهة المستخدم الرسومية التي تعمل على توفير واجهة بين النظام والمستخدم في تلقي المدخلات من المستخدم وعرض النتائج مرة أخرى إلى المستخدم.

يشرح هذا البحث كيفية بناء برنامج قائم على SaaS وتطبيقه في اكتشاف واجهة المستخدم. يهدف التطبيق إلى تسهيل المستخدم في تغيير وتكوين المظهر ، بما في ذلك إضافة وتغيير وحذف الرموز والألوان والخطوط والعناوين والقوائم الموجودة في التطبيق التي سيتم إنشاؤها بواسطة المستخدم. يتم البحث في واجهة المستخدم التي يريدها المستخدم عن طريق إدخال كلمة رئيسية تقود اسم النموذج ، واسم المكون ، وعدد المكونات المطلوبة. الطريقة المستخدمة في اكتشاف واجهة مستخدم التطبيق هي الويب الدلالي. وتتمثل العملية في استخدام خادم RDF والاستعلام عن SPARQL كأداة لتنفيذ الاكتشاف.

الحساب علي أساس التجارب التي أجريت مع اثنين من نماذج البحث ، اي البحث التفاصيل البيانات البحث والحصول علي استدعاء 95 ٪ و 100 ٪ من الدقة ، ولكن دقه البحث الحصول علي البيانات قيمه 95 ٪ وبحثا عن التفاصيل المكتسبة من القيم 90 %. يتم الحصول علي الفرق لان هناك بعض التفاصيل حول البحث المستندات ذات الصلة تم العثور عليها والاستعلام عنها ، وذلك علي بيانات البحث تظهر نتائج أكثر اهميه وإظهار قدرات النظام علي نحو أفضل.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# BAB 1

# PENDAHULUAN

## 1.1 Latar Belakang

SaaS (*Software as a Service*) adalah suatu model penyampaian aplikasi perangkat lunak oleh suatu vendor perangkat lunak yang mengembangkan aplikasi *web* yang dioperasikan untuk digunakan oleh pelanggannya melalui internet [1]. Jadi SaaS (*Software as a Service*) merupakan model dimana aplikasi ditawarkan kepada klien sebagai sebuah *service*. Jika sebuah *service* disajikan kepada klien, klien tidak perlu merawat dan melakukan *update* pada aplikasi tersebut. Namun sebaliknya, apabila *provider* akan melakukan *update* pada aplikasi tersebut, maka klien hanya bisa mengikuti apa yang dilakukan oleh *provider*.

SaaS dengan berbagai macam lapisan arsitektur bertujuan untuk memenuhi berbagai kebutuhan fungsional dan berkualitas. Penelitian ini menyajikan *framework* pada lapisan GUI untuk memudahkan *user* dalam mengubah dan mengkonfigurasi tampilannya, termasuk menambahkan/ menghapus/ mengubah ikon, warna, *font*, judul, dan menu yang ada didalamnya. Alternatif yang digunakan untuk mengatasi permasalahan tersebut yaitu menggunakan pendekatan ontologi.

Ontologi adalah deskripsi tentang suatu konsep dalam sebuah domain dan properti dari setiap konsep beserta dengan batasannya. Ontologi digunakan untuk memperoleh kostumisasi dan penyebaran informasi untuk klien. Pada lapisan GUI, ontologi UI dibangun untuk memberikan makna yang lebih luas dan membantu sistem komputer dengan cara yang lebih mudah.

Aplikasi SaaS memiliki beberapa lapisan arsitektur yaitu data *layer*, *service layer*, *process layer*, dan UI *layer*. Data *layer* dan *service layer* berfungsi sebagai



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

lapisan yang menetapkan struktur data dan operasi untuk aplikasi. *Procces layer* yaitu lapisan yang berfungsi mengelola mekanisme *service*. UI *layer* yaitu merupakan lapisan GUI yang berfungsi untuk menyediakan *interface* antara sistem dan *user* dalam menerima *input* dari *user* dan menampilkan hasil kembali kepada *user*. UI yang tersedia memudahkan *user* untuk mengubah dan mengkonfigurasi tampilannya, termasuk menambahkan/mengubah/menghapus ikon, warna, font, judul, dan menu yang ada didalamnya[2].

Pada penelitian ini mengajukan pendekatan anotasi *interface repository* menggunakan ontologi dengan UI *discovery*. Agar UI *discovery* menjadi cerdas dan terotomasi maka dicoba menggunakan *query* SPARQL.

Memanfaatkan waktu ialah sangat penting dalam kehidupan sebagai orang yang beriman, sehingga manusia harus mampu memanfaatkan waktu sebaik-baiknya. Waktu memegang peranan yang penting dalam kehidupan sehingga Allah SWT akan meminta pertanggungjawaban dari setiap manusia untuk waktu yang telah diberikan Allah SWT kepada setiap manusia untuk waktu yang telah diberikan Allah SWT kepada setiap hamba-Nya. Dalam suatu hadits telah disebutkan :

لَا تَزُوْلُ قَدَمَا ابْنِ آدَمَ يَوْمَ الْقِيَامَةِ مِنْ عِنْدِ رَبِّهِ حَتَى يُسْأَلَ عَنْ خَمْسٍ عَنْ عُمْرِهِ فِيْمَا أَفْنَاهُ وَعَنْ شَبَابِهِ فِيْمَا أَبْلَاهُ وَعَنْ مَالِهِ مِنْ أَيْنَ اكْتَسَبَهُ وَفِيْمَا أَنْفَقَهُ وَمَاذَا عَمِلَ فِيْمَا عَلِمَ

Artinya :

"*Tidak akan bergeser kedua kaki anak Adam di hari kiamat dari sisi RabbNya, hingga dia ditanya tentang lima perkara (yaitu**):** tentang umurnya untuk apa ia habiskan**,** tentang masa mudanya untuk apa ia gunakan**,** tentang hartanya dari mana ia dapatkan, dan dalam hal apa (hartanya tersebut) ia belanjakan serta apa saja yang telah ia amalkan dari ilmu yang dimilikinya**.**" (HR. at-Tirmidzi no. 2416, ath-Thabrani dalam* al-Mu'jam al-Kabir *jilid 10 hal 8 Hadits no. 9772 dan Hadits*

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

*ini telah dihasankan oleh Syaikh Albani dalam Silsilah al-AHadits ash-Ashahihah no. 946)*

Penelitian ini memudahkan pengguna dalam memanfaatkan waktunya sebaik mungkin, karena aplikasi ini membantu untuk mempersingkat waktu dalam proses pencarian.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana penerapan ontologi dalam *user interface discovery*?
2. Seberapa akurat *user interface discovery* dalam *interface repository* menggunakan *query* SPARQL?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan yaitu sebagai berikut:

1. Menerapkan ontologi dalam aplikasi *user interface discovery*.
2. Mengukur akurasi *user interface discovery* dalam *interface repository* dengan menggunakan *query* SPARQL.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian yang akan dilakukan yaitu sebagai berikut:

1. Mempersingkat waktu pengembangan sistem informasi.
2. Mempermudah proses pembuatan sistem informasi.

## 1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. *Web service* yang digunakan adalah *web service* ERP Pondok Pesantren yang telah dibangun pada penelitian sebelumnya.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

2. Data diperoleh dari *web service* ERP Pondok Pesantren dalam bidang akademik.
3. Kata kunci yang digunakan yaitu nama *form* untuk pencarian data, sedangkan untuk pencarian komponen yaitu komponen HTML berupa *textfield*, *button*, label, dan jumlah yang diinginkan.
4. Data berupa *file* PHP.
5. Ontologi yang digunakan berupa RDF *map*.

### 1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penelitian ini adalah sebagai berikut:

**BAB 1 PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

**BAB 2 KAJIAN PUSTAKA**

Bab ini berisi tentang teori yang berkaitan dengan penelitian, yaitu mengenai *Software as a Service* (SaaS), web semantik, D2RQ, *User Interface Management System* (UIMS), UI ontologi, UI *discovery*, dan ROC (*Receiver Operating Characteristic*).

**BAB 3 METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang metode yang digunakan dalam penelitian, desain penelitian, perancangan sistem, dan pengujian sistem.

**BAB 4 HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil perancangan sistem serta pembahasan dari penelitian.

**BAB 5 PENUTUP**

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Bab ini merupakan kesimpulan dari penelitian yang telah dilakukan, dan saran yang diharapkan dapat bermanfaat untuk pengembangan aplikasi.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# BAB 2

# KAJIAN PUSTAKA

## 2.1 *Software as a Service* (SaaS)

*Software as a Service* (SaaS) adalah suatu model penyampaian aplikasi perangkat lunak oleh suatu vendor perangkat lunak yang mengembangkan aplikasi *web* yang dioperasikan untuk digunakan oleh pelanggannya melalui internet. *Software as a Service* (SaaS) memiliki keunggulan *Service Oriented Architecture* yang mana aplikasi perangkat lunak berkomunikasi satu sama lain. Model *service* ini menggambarkan sebagai aplikasi tunggal yang berjalan sebagai *service* pada sisi *server* dengan banyak pengguna yang dapat menjalankannya dari *web browser* secara bersamaan [3].

Sebagai contohnya yaitu *Google Play Store* di mana aplikasi yang dapat dibeli dimanapun dan dapat dijalankan dari *web browser*, sejumlah orang dapat membeli dan menjalankan aplikasi secara bersamaan. Hal ini membantu dalam mengurangi biaya pada sumber daya dan pada titik pelanggan yang mengalami kerumitan proses *service* gratis pada permintaan, dan *Cloud* pengguna yang dapat memperoleh aplikasi tanpa perlu instalasi dan pemeliharaan perangkat lunak [4].

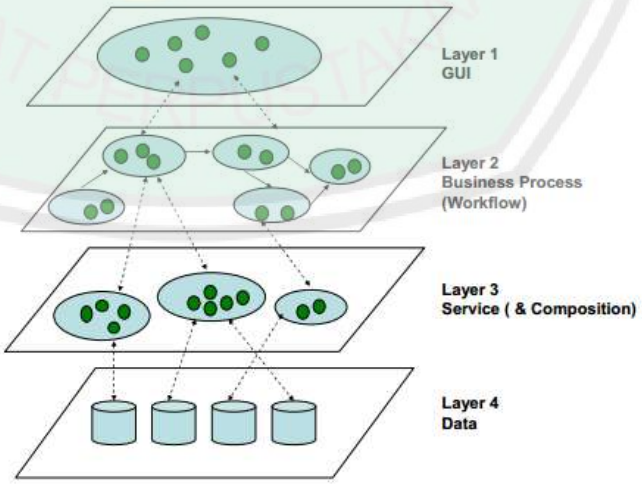
Gambar 2.1 Arsitektur SaaS menjelaskan bahwa SaaS dapat diklasifikasi berdasarkan arsitekturnya. SaaS dengan penggunaan teknologi baru dapat membantu dalam mengurangi waktu dan penghematan biaya dalam mengkonversi di lokal *server* menjadi produk berbasis SaaS.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG


Gambar 2.1 Arsitektur SaaS

*Framework* SaaS memiliki empat lapisan, yaitu lapisan data, lapisan *service*, lapisan bisnis proses (alur kerja), dan lapisan GUI. Semua lapisan memiliki informasi ontologi masing-masing seperti pada gambar 2.2. [2]




Gambar 2.2 Arsitektur *Multi-layer* SaaS

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Macam-macam lapisan arsitektur *multi-layer* SaaS yaitu:

a. Data *Layer*

Pada domain, beberapa data sistem ontologi dapat didefinisikan oleh kelompok komunitas yang berbeda. Sebuah penelitian telah membahas tentang bagaimana menentukan ontologi dalam domain tertentu, membandingkan dan mengintegrasikan sistem ontologi antara komunitas yang berbeda. Sebagai contoh, IEEE dan ACM adalah dua komunitas besar yang memiliki standar yang berbeda, dan dengan demikian apabila sistem ontologi yang sesuai akan mengalami kemiripan tetapi berbeda. Ontologi integrasi dikembangkan untuk memecahkan heterogeneitiesnya, yang mengacu pada pembangunan ontologi yang besar dan lengkap di tingkat yang lebih tinggi menggunakan sistem ontologi yang ada. [5]

b. *Service Layer*

*Service layer* biasa disebut dengan lapisan servis atomik dan lampisan servis komposit. Lapisan servis atomik merupakan layanan dasar yang menyelesaikan operasi dasar, sedangkan lapisan servis komposit yaitu melibatkan beberapa servis atomik yang melakukan tugas-tugas yang lebih kompleks. Kustomisasi servis atomik merupakan hal yang termudah. Untuk membedakan layanan, pengguna harus memahami bagian yang unik sebuah layanan dalam bagian deskripsi/profil. Setiap layanan memiliki kemampuan dasar yang dibawah aturan khusus dan persyaratan. Penelitian dari Justin O'Sullivan menyebutkan bahwa kemampuan yang ditawarkan sebuah lapisan servis dapat memenuhi kebutuhan nasabah dengan kendala-kendala tertentu [6]. Persyaratan untuk

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

menawarkan termasuk beberapa aspek sebagai property misalnya biaya, diskon, ketersidaan, QoS, kemudahan penggunaan, dan lain-lain.

c. *Business Process Layer*

Dalam lapisan ini, layanan sebuah proses dilakukan untuk mencapai tugas bisnis yang lebih kompleks, *workflow*, yang terdiri dari serangkaian kegiatan dan mewakili proses bisnis. Pengguna dapat mencari *workflow repository* menggunakan kata kunci dan mengambil yang relevan berdasarkan kepentingannya.

d. GUI *Layer*

Lapisan GUI merupakan lapisan yang berfungsi untuk menyediakan *interface* antara sistem dan *user* dalam menerima *input* dari pengguna dan menampilkan hasil kembali kepada pengguna. GUI yang tersedia memudahkan pengguna untuk mengubah dan mengkonfigurasi tampilannya, termasuk menambah/ mengubah/ menghapus ikon, warna, *font*, judul, dan menu yang ada didalamnya. Lapisan GUI berada pada lapisan teratas pada arsitektur *multi-layer* seperti yang diilustrasikan pada gambar 2.2. Lapisan GUI menyediakan *interface* antara sistem dan pengguna.

## 2.2 Web Semantik

*World Wide Web* (WWW) telah mengubah cara orang untuk berkomunikasi antara satu dengan yang lainnya dan juga cara berbisnis yang telah diselenggarakan oleh WWW sedang berkembang di dunia menjadi pengetahuan bagi masyarakat (*knowledge society*).

Web saat ini telah memuat isi yang menyesuaikan dengan konsumsi manusia. Meskipun isi web dihasilkan secara otomatis dari basisdata, namun ditampilkan

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

tanpa struktur informasi yang ada dalam basisdata.kebanyakan jenis penggunaan web saat ini mencakup sebuah pencarian dan memanfaatkan informasi yang telah diperoleh.

Pada peningkatan layanan web telah mengalami permasalahan yaitu arti dari isi web tidak dapat dibaca oleh mesin. Dalam istilah, sebuah mesin tidak dapat menginterprestasikan kalimat dan mengekstrasi informasi yang berguna untuk para pengguna.

Ketika manusia dengan mudah untuk memahami kalimat dengan struktur yang berbeda, maka mesin tetap mengalami kesulitan dalam pengolahan data. Oleh sebab itu, dibutuhkan sebuah pendekatan dengan menyatakan isi web dalam sebuah bentuk yang ebih mudah diproses oleh mesin komputer dan menggunakan cara yang cerdas untuk dalam mendapatkan keuntungan dari representasi tersebut. Cara yang digunakan yaitu disebut *Semantic web initiative.*

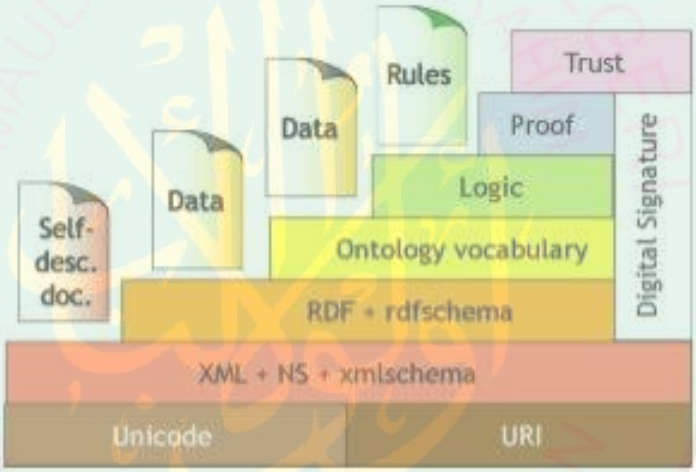
Web semantik dikenalkan oleh Tim Berners-Lee (TBL) sebagai penemu *World Wide Web* (WWW) merupakan suatu informasi yang dikumpulkan untuk menghubungkan berbagai dokumen di internet. Web semantik merupakan suatu teknologi yang membuat komputer agar bisa memahami arti dari sebuah informasi berdasarkan metadata atau isi dari sebuah informasi sehingga menjadi lebih cerdas dengan menyajikan deskripsi yang dapat diterjemahkan oleh mesin. Seperti pada web biasa yang sebagai mesin pencari untuk menggabungkan berbagai macam informasi atau halaman kedalam sebuah koleksi yang sama.

Web semantik adalah teknologi yang dikembangkan dengan berbasis RDF format. Terdapat tiga komposisi pada RDF yaitu *subject*, *predicate*, dan *object*. *Subject* adalah entitas yang ditunjukkan oleh teks, *predicate* yaitu komposisi yang

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

menjelaskan tentang sudut pandang dari *subject* terhadap *object*, sedangkan *object* merupakan masukan yang diberikan oleh *subject* secara jelas.

Web semantik memiliki kesamaan dengan web biasa, perbedaannya hanya ada pada metode pencarian informasi web yang diinginkan. Jika pada web biasa hanya dapat mencari informasi web yang memiliki beberapa kata yang menjadi bahan pencarian, namun dalam web semantik dapat melakukan pencarian dengan lebih terstruktur dan dimengerti oleh mesin. Hal tersebut dikarenakan web semantik menggunakan XML (*Extensible Markup Language*), RDF (*Resource Description Framework*), dan OWL (*Ontology Web Language*).




Gambar 2.3 Arsitektur Web Semantik

Dari gambar arsitekur web semantik tersebut, lapisan yang digunakan untuk membangun web semantik yaitu : [7]

a. Lapisan *Unicode* dan URI yang digunakan untuk memastikan dan menyediakan sarana untuk mengidetifikasi suatu objek dalam web semantik.
b. Lapisan XML (*Exentensible Markup Language*) dan NS (*Namespace*) serta XML *Schema* merupakan pernyataan yang digunakan untuk menyajikan struktur dan mengintegrasikan pada sebuah web. XML berfungsi untuk

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

mendefinisikan data dengan *custom tag* sehingga memberikan makna kata. XML *Schema* merupakan bahasa yang digunakan sebagai pernyataan untuk aturan dalam membuat struktur XML.

c. Lapisan RDF (*Resource Description Framework*) merupakan pernyataan yang digunakan untuk mendeskripsikan sebuah *resource website* dalam bentuk subjek, predikat dan objek sebagai N-Triple. RDF adalah standar W3C yang berfungsi untuk memodelkan informasi. RDF *Schema* merupakan bagian penting dari semantik web yang menggunakan URI untuk mengidentifikasi objek yang berbeda.
d. Lapisan *Web Ontology Language* (OWL) merupakan bahasa standar ontologi yang dibangun pada RDF, dan didasarkan pada XML, digunakan untuk mengekspresikan kebutuhan aplikasi komputer dalam menangani pengetahuan dan informasi dalam dokumen [8].
e. Lapisan *logic and proof* yaitu lapisan yang memberikan kemampuan untuk melakukan penalaran secara logika yang berfungsi untuk memantapkan konsistensi dan lebenaran himpunan data dan mendapatkan kesimpulan-kesimpulan yang membutuhkan himpunan data yang diketahui.
f. *Trust* merupakan lapisan yang tertinggi, oleh karena itu, *trust* berfungsi sebagai alat penyedia otentifikasi identitas data dan mengukur tingkat kepercayaan dalam keamanan dan kualitas sebuah informasi.

Tiap lapisan web semantik dibangun sesuai tingkat lapisannya. Level lapisan tersebut semakin keatas maka semakin kompleks daripada lapisan yang dibawahnya. Lapisan yang berada ditingkat bawah tidak selalu bersambungan terhadap lapisan diatasnya.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Cara kerja web semantik yaitu dalam XML akan mengacu pada ke URI. URI merupakan kode dari segalanya dan sebagai objek yang berbeda, karena XML tidak bisa melakukan relasi data sehingga tidak bisa merepresentasi dengan entity lainnya. Sedangkan RDF bisa merepresentasi dengan entitas lainnya, namun dia membutuhkan skema untuk merepresentasikannya, sehingga disebut RDFS. RDFS memiliki sebuah kelemahan yaitu tidak bisa melakukan relasi *one to many*, sehingga disempurnakan oleh ontologi yang disebut dengan OWL (*Web Ontology Language*). Ontologi merupakan representasi dari XML, RDF, dan URI. Namun, untuk menampilkan semua data tersebut membutuhkan URI.

Web semantik memiliki beberapa ciri-ciri, diantaranya:

- Transformasi dari tempat penyimpanan yang bersifat terpisah menjadi satu.
- Bermodel komputer jaringan, *software-as-a-service business*, *web services*, *cloud computing*.
- *Open technologies*, sebagian besar semuanya berjalan dalam *platform open source / free*.
- *Open identity*, *OpenID*, seluruh informasi memiliki kebebasan.
- Teknologi web semantik yang meliputi seperti RDF, OWL, SWRL, SPARQL, GRDDL, *platform* apikasi semantik.
- *Distributed databases*, database terdistribusi dalam WWD (*World Wide Database*).
- *Intelligent applications*.

### 2.2.1 *Resource Description Framework (*RDF)

*Resource Description Framework* (RDF) adalah suatu kerangka kerja umum untuk menggambarkan setiap sumber daya internet seperti situs web dan isinya.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

*Resource Description Framework* (RDF) biasa disebut dengan istilah metadata, atau "data tentang data", informasi yang menggambarkan tentang isi. Bahasa RDF digunakan sebagai standar untuk mendeskripsikan web *resource*.[9]

XML digunakan oleh RDF sebagai dasar sintaks dalam melakukan pengkodean. Secara umum, XML (*Extensible Markup Language*) adalah bahasa *mark up* yang dikembangkan oleh *World Wide Web Consortium* (W3C) yang digunakan untuk memberikan informasi yang terstruktur pada dokumen web.

Pada RDF, sebuah pernyataan yang akan direpresentasikan dinyatakan dalam bentuk *triple*, dimana terdapat *subject*, *predicate*, dan *object*. *Subject* adalah *resource* yang akan dijelaskan serta menggunakan URI sebagai pengidentifikasinya. *Predicate* ialah *property* yang digunakan untuk menghubungkan *resource* dengan *object*, *object* berupa nilai literal (*literal value*).

Sebuah internet didefinisikan sebagai sumber daya dengan dengan *Uniform Resource Identifier* (URI). Termasuk *Uniform Resource Locators* (URL) yang mengidentifikasi seluruh situs web serta halaman web tertentu seperti *tag* pada HTML. laporan deskripsi *Resource Description Framework* (RDF), terbungkus sebagai bagian dari bagian *Extensible Markup Language* (XML) dapat dimasukkan dalam halaman web.

RDF memiliki bentuk dan kode yang bervariasi. Variasi tersebut sesuai dengan kalimat yang akan dideskripsikan. Variasi ditandai dengan penggunaan tag yang mempunyai fungsi yang berbeda dengan yang lainnya. Berikut adalah beberapa variasi bentuk RDF:

- RDF *Anonim*, yaitu sumber daya tanpa URI yang digunakan untuk menghubungkan banyak elemen (predikat dan objek) dengan subjek. Elemen

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

objek dan predikat dapat berjumlah lebih dari satu namun saling berkaitan dengan subjek asalnya.

- RDF *Container*, merupakan sumber daya yang dapat menampung banyak elemen. Elemen yang tergabung dapat terdiri dari sumber daya, literal ataupun sumber daya anonim (node kosong). RDF *Container* mempunyai tiga tipe yaitu `rdf:bag`, `rdf:seq` dan `rdf:alt`. `rdf:bag` menyimpan banyak nilai tanpa memperhatikan urutan dan duplikasinya. `rdf:seq` meyimpan nilai secara terurut berdasarkan alfabet tanpa memperhatikan duplikasi nilainya sedangkan `rdf:alt` menyimpan nilai alternatif dari nilai yang sudah ada.
- RDF *Collection*, yaitu grup yang dapat menyimpan kumpulan sumber daya dalam bentuk list yang terstruktur. List ini dibentuk dengan menggunakan tipe rdf:list, sumber daya rdf:nil dan property rdf:first dan rdf:rest. Berbeda dengan RDF:*Container*, pada RDF *Collection* kita mendefinisikan anggota sebuah grup secara lengkap dan spesifik.
- RDF *Reification*, RDF mempunyai kosa kata internal yang digunakan untuk mendeskripsikan pernyataan-pernyataan yang ada pada RDF. Pendeskripsian ini disebut *Reification*. Kosa kata *reification* terdiri dari tipe rdf:statement dan properti rdf:subject, rdf:predicate dan rdf:object. Penggunaan reification diharapkan dapat memperjelas informasi dari suatu pernyataan.

Cara kerja RDF dapat dijelaskan dengan contoh sebagai berikut, misalkan ada pernyataan: “Anisa seorang wanita”, maka “Anisa” adalah *subject*,”seorang” adalah *predicate*, dan “wanita” adalah *object*. Ketiga kata ini merupakan *resources*, “Anisa” adalah sebuah *resource*, “seorang” merupakan sebuah *resource*, dan “wanita”juga merupakan sebuah *resource*. Penamaan ini bersifat umum, sehingga

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

ketika ada pernyataan bahwa “Anisa membaca buku”, maka sistem komputer akan mengetahui bahwa “Anisa” pada pernyataan “Anisa seorang wanita” adalah sama dengan “Anisa” pada pernyataan “Anisa membaca buku”. Oleh karena itu, untuk membedakan setiap *resource* menggunakan URI (*Uniform Resource Identifier*).

Beberapa manfaat *Resource Description Framework* (RDF) meliputi:

- Dengan tersedianya kerangka kerja yang konsisten, *Resource Description Framework* (RDF) akan mendorong pemberian metadata tentang sebuah internet.
- Karena *Resource Description Framework* (RDF) telah mencakup standar sintaks untuk merepresentasikan dan *query* data, maka perangkat lunak yang memanfaatkan metadata akan menjadi lebih mudah ketika menjalankannya dan lebih cepat untuk memberi hasilnya.
- Standarisasi sintaks dan kemampuan permintaan akan memudahkan aplikasi untuk bertukar informasi dengan lebih mudah.
- Mesin pencari akan mendapatkan hasil yang lebih tepat dari pencarian tersebut berdasarkan metadata bukan pada indeks yang berasal dari data teks lengkap yang terkumpulkan.
- Pengembang perangkat lunak akan memiliki data yang lebih tepat untuk bekerja dengan lebih cerdas.

### 2.2.2 Ontologi

Ontologi adalah sebuah deskripsi formal tentang sebuah konsep secara eksplisit dalam sebuah wawasan dari setiap konsep beserta dengan batasannya [10]. Dalam konteks sistem *database*, ontologi dapat dilihat sebagai tingkat abstraksi dari model data, analog dengan model hirarkis dan relasional.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Menurut Gruber, Ontologi sering disamakan dengan hirarki taksonomi dari kelas, definisi kelas, dan hubungan subsumption, tapi ontologi tidak perlu terbatas pada bentuk-bentuk ini. Ontologi juga tidak terbatas pada memiliki arti sebagai logika tradisional yang hanya memperkenalkan terminologi dan tidak menambahkan pengetahuan tentang dunia [11].

Ontologi merupakan bagian dari standar W3C *stack* untuk *Semantic Web*, berfungsi untuk menentukan kosakata standar konsep yang bertujuan untuk pertukaran data pada sistem. Peran ontologi pada sistem *database* adalah untuk menentukan representasi pemodelan data pada tingkat abstraksi atas desain *database* tertentu (logis atau fisik), sehingga data dapat diekspor, diterjemahkan, dan bersatu di seluruh sistem.

Penggunaan ontologi di bidang *web service* telah banyak diterapkan, baik untuk akademisi dan industri. Salah satu aspek penting dari ontologi yaitu kemampuannya untuk memperluas dan berkomunikasi dengan sistem ontologi lainnya. Selain itu, ontologi telah digunakan sebagai bahasa standar di bidang semantik.

Menurut penelitian Tri Nuryati yang berjudul "Teknologi Untuk Peningkatan *Web Service Search* Melalui *Ontology-Based Semantic Interoperabiliy*" menjelaskan bahwa Metode *Ontology* adalah metode pencarian yang membaca semua *field* yang ada pada semua tabel yang ada dalam *database* [12].

Selain itu terdapat penelitian yang berjudul "*Ontology Mapping For ERP Business Process Variations*" yang diteliti oleh Anang Kunaefi dan Riyanarto Sarno menjelaskan bahwa ontologi dapat mewakili berbagai *domain* tertentu. Salah

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

satu fitur penting dari ontologi adalah kemampuannya untuk mengembangkan beberapa perubahan bisnis yang baru. Dalam arsitektur *multi-tenancy*, seperti berdasarkan *service-ERP* (*Enterprise Resource Planning*) [13].

Bahasa ontologi dibedakan menjadi dua, yaitu konvensional dan *web-based*. Konvensional yaitu bahasa ontologi yang tidak digunakan secara langsung pada web, sedangkan bahasa ontologi yang digunakan untuk web semantik yaitu bahasa OWL (*Ontology Web Language*). Perangkat lunak yang digunakan untuk membangun ontologi ada bermacam-macam seperti Protégé, OntoEdit, WebOde, dan Altofa.

*Web Ontology Language* (OWL) adalah bahasa *Semantic Web* yang dirancang untuk mewakili pengetahuan yang kaya dan kompleks tentang hal-hal, kelompok hal, dan hubungan antara hal-hal. *World Wide Web Consortium* (W3C) menciptakan web kelompok kerja ontologi W3C (*Web Ontology Working Group)* sebagai bagian dari kegiatan semantik *web*. OWL merupakan bahasa standar ontologi yang dibangun pada RDF, dan didasarkan pada XML, digunakan untuk mengekspresikan kebutuhan aplikasi komputer dalam menangani pengetahuan dan informasi dalam dokumen [8].

OWL menyediakan tiga subbahasa yang dirancang oleh para pengguna yaitu:

- OWL *Lite*, digunakan oleh pengguna yang membutuhkan suatu hirarki pengklasifikasian dan berbagai *constraints* sederhana.
- OWL DL, digunakan oleh pengguna yang menginginkan tingkat ekpresi maksimal dan semua konklusi yang dihasilkan dapat dihitung dalam waktu yang terbatas.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

- OWL *Full*, digunakan oleh pengguna yang menginginkan tingkat ekpresi maksimal dan kebebasan sintaks dari RDF tanpa mempertimbangkan komputasi yang dibutuhkan.

Penulisan sebuah ontologi yang dapat ditafsirkan dengan jelas dan digunakan oleh perangkat lunak maka membutuhkan sintaks dan formal semantik untuk OWL.

### 2.2.3 SPARQL

SPARQL (*Simple Protocol And RDF Query Language*) merupakan bahasa *query* dari standar ontologi. Bahasa ini merupakan salah satu bahasa yang bisa melakukan *query* terhadap *file* ontologi [14]. SPARQL memungkinkan pengguna untuk menulis permintaan terhadap data yang disebut dengan "*key-value*" yaitu data yang mengikuti RDF spesifikasi W3C SPARQL menyediakan operasi *query* yang analitis seperti *JOIN, SORT, AGGREGATE* untuk skema data yang membutuhkan deskripsi skema yang terpisah. Selain itu, SPARQL memberikan spesifik grafik sintaks traversal untuk data yang dapat dianggap sebagai grafik. Contoh berikut ini menunjukkan *query* sederhana yang memanfaatkan definisi ontologi "foaf", atau sering disebut ontologi "*friend-of-a-friend*". Contoh *query* berikut mengembalikan nama dan email dari setiap orang dalam *dataset*, seperti pada gambar 2.4.

```
PREFIX foaf:http://xmlns.com/foaf/0.1/
PILIH? Nama? Email
MANA {
?seseorang foaf: orang.
?orang foaf: Nama nama.
?orang foaf: ?inbox email.
}
```

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Gambar 2.4 Contoh *Query* SPARQL

*Query* diatas menunjukkan penggabungan dengan subjek yang cocok, di mana jenis predikat, "a", adalah orang (foaf: Orang) dan orang memiliki satu atau lebih nama (foaf: nama) dan kotak surat (foaf: inbox) .

Penulis *query* ini memilih referensi subjek menggunakan nama variabel "? Orang" untuk kejelasan dibaca. Karena elemen pertama dari tiga selalu *subjek*, penulis bisa saja dengan mudah digunakan setiap nama variabel, seperti "? Subj" atau "? X". Apapun nama yang dipilih, itu harus yang sama pada setiap baris *query* untuk menandakan bahwa mesin *query* untuk bergabung tiga kali lipat dengan subjek yang sama.

Bentuk *query* SPARQL yaitu [15]:

a. *SELECT*

   Digunakan untuk mengekstrak nilai mentah dari titik akhir SPARQL, hasilnya dikembalikan dalam *format* tabel.

b. *CONSTRUCT*

   Digunakan untuk mengekstrak informasi dari titik akhir SPARQL dan mengubah hasilnya ke RDF valid.

c. *ASK*

   Digunakan untuk memberikan benar / hasil salah sederhana untuk *query* pada titik akhir SPARQL.

d. *DESCRIBE*

   Digunakan untuk mengekstrak sebuah grafik RDF dari endpoint SPARQL, isi yang tersisa untuk titik akhir untuk memutuskan berdasarkan apa pengelola dianggap sebagai informasi yang berguna.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Masing-masing bentuk permintaan ini membutuhkan permintaan *WHERE* untuk membatasi *query* meskipun dalam kasus *DESCRIBE* permintaan *WHERE* adalah opsional.

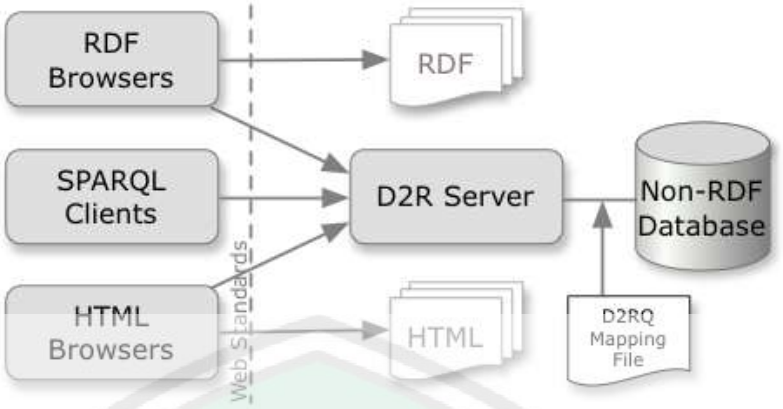
Menurut Maskur pada penelitiannya, menjelaskan bahwa SPARQL (*Simple Protocol And RDF Query Language*) merupakan bahasa *query* untuk RDF/OWL. SPARQL menyediakan fasilitas untuk mengekstrak informasi dalam bentuk URI, *blank node* dan literal, mengekstrak *subgraph* RDF, dan membangun *graph* RDF baru berdasar pada infrmasi dari graf yang dilakukan *query* [16].

## 2.3 D2RQ

D2RQ merupakan suatu *platform* yang memungkinkan pengguna untuk mengakses data dalam model RDF secara virtual dari model basis data relasional tanpa perlu menduplikasinya dalam suatu penyimpanan RDF [17]. Untuk menguji hasil proses yang telah dilakukan, digunakan *tool* d2r-*server* yang merupakan bagian dari *framework* D2RQ.

*Framework* D2RQ juga menyediakan akses ke *data set* dengan menggunakan *query* SPARQL. Hal tersebut dilakukan dengan meggunakan *tool* SNORQL yang merupakan suatu AJAX-based SPARQL untuk mengakses data dalam format RDF dengan *query* SPARQL. Kemudian hasil dari *query* tersebut disajikan dalam bentuk dokumen HTML, selain dalam dalam bentuk HTML hasil *query* tersebut juga dapat disajikan dalam bentuk lainnya, seperti JSON, XML, dan XML+XSLT. Arsitektur *framework* D2RQ dapat dilihat pada gambar 2.5.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 2.5 Arsitektur *platform* D2RQ

D2RQ *Platform* menyediakan lingkungan yang memudahkan dalam menghasilkan *mapping file*. Proses menghasilkan *mapping file* pada prinsipnya cukup sederhana. Beberapa parameter yang dibutuhkan untuk menghasilkan sebuah *mapping file* adalah:

a. Akun koneksi ke *database* relasional yang meliputi nama pengguna dan passwordnya
b. Nama *database* relasional yang akan dikonversi
c. Nama *file output*

Tabel 2. 1 Properti Konfigurasi D2R *Server*

| Properti | Deskripsi |
|---|---|
| `rdfs:label` | Nama *server* ditampilkan di seluruh antarmuka HTML |
| `d2r:baseURI` | URI sebagai dasar *server*. Sama seperti `-b` command line parameter. |
| `d2r:port` | *Port server*. Sama seperti `--port` command line parameter |
| `d2r:kosakataIncludeInstances` | Mengontrol apakah representasi RDF dan HTML dari kelas kosa kata juga akan menampilkan contoh, dan apakah representasi properti juga melakukan tiga kali lipat menggunakan properti (*default true* ) |
| `d2r:autoReloadMapping` | Menentukan apakah perubahan pada *file* pemetaan terdeteksi secara otomatis |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| | |
|---|---|
| d2r:limitPerClassMap | Menentukan maksimum untuk jumlah entitas per peta kelas yang akan ditampilkan di halaman "direktori" dari antarmuka web. Ini akan menghentikan halaman yang terlalu besar, namun mencegah pengguna mengakses keseluruhan data melalui antarmuka web. Pengaturan tersebut tidak mempengaruhi *output* RDF atau *query* SPARQL. Defaultnya adalah 50 . |
| d2r:limitPerPropertyBridge | Menentukan maksimum untuk jumlah nilai dari masing-masing jembatan properti yang akan ditampilkan di antarmuka web. |
| d2r:sparqlTimeout | Menentukan batas waktu dalam hitungan detik untuk mengakhiri SPARQL *server*. |
| d2r:pageTimeout | Menentukan batas waktu dalam hitungan detik untuk menghasilkan halaman deskripsi web |
| d2r:metadataTemplate | Menggunakan *template* metadata web default, mengacu ke file RDF yang dikodekan TTL |
| d2r:datasetMetadataTemplate | Menggunakan template metadata *dataset* default, mengacu ke file RDF yang dikodekan TTL |
| d2r:disableMetadata | Memungkinkan pembuatan dan publikasi otomatis semua dataset dan metadata web. Nilai yang digunakan adalah "*true*" dan "*false*" . |
| d2r:documentMetadata | Alternatif sederhana untuk d2r:metadataTemplate : Nilai harus berupa *node* kosong. Setiap pernyataan yang melibatkan *node* kosong ini akan disalin sebagai metadata ke dalam dokumen RDF yang dihasilkan oleh D2R *Server*, dengan *node* kosong diganti dengan URL dokumen |

### 2.3.1 D2RQ *Mapping Language*

D2RQ *Mapping Language* adalah bahasa deklaratif untuk menggambarkan hubungan antara skema basis data relasional dan kosakata RDFS atau ontologi OWL [18], . D2RQ *mapping* merupakan dokumen RDF yang disimpan dalam bentuk *turtle*. Semantik web memodelkan dan menggambarkan datanya dalam bentuk RDF. Sehingga D2R *Server* menggunakan D2RQ *mapping* untuk memetakan konten *database* kedalam format RDF.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pemetaan terebut digambarkan dengan *virtual* RDF *Graph* yang berisi tentang informasi dari basisdata. *Virtual* RDF *Graph* dapat diakses dengan berbagai cara sesuai pada pengimplementasiannya. *Platform* D2RQ menyediakan akses SPARQL, *Linked* Data *server*, RDF *dump generator*, dokumen HTML, dan API Jena. Contoh D2RQ *mapping* diilustrasikan pada gambar 2.6.

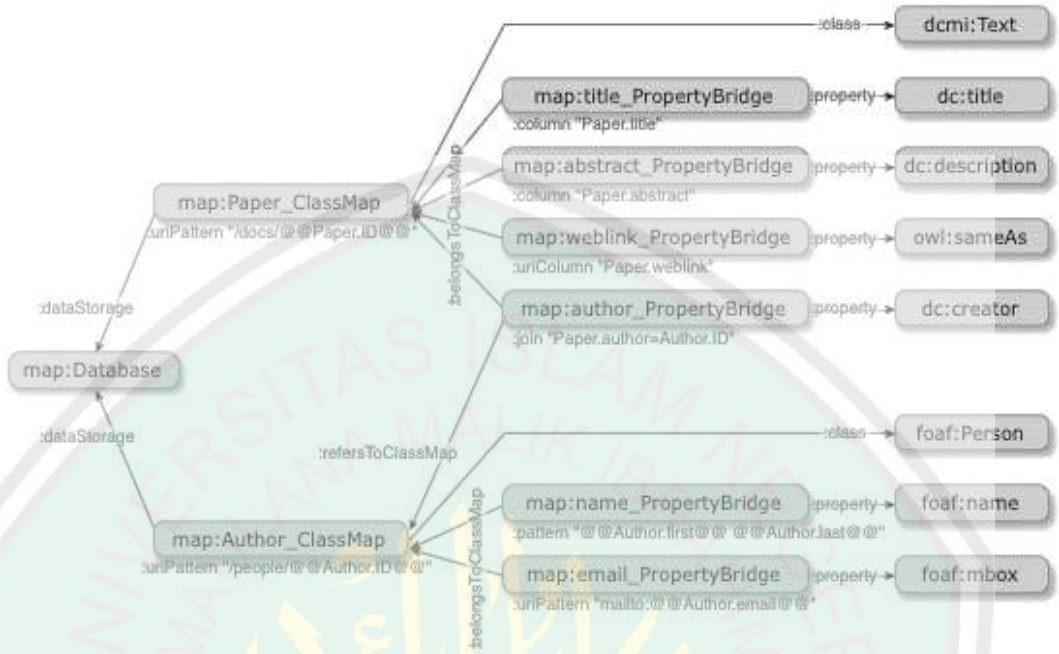
```
# D2RQ Namespace
@prefix d2rq:<http://www.wiwiss.fu-
berlin.de/suhl/bizer/D2RQ/0.1#> .
# Namespace of the ontology
@prefix : <http://annotation.semanticweb.org/iswc/iswc.daml#>
.
# Namespace of the mapping file; does not appear in mapped
data
@prefix map: <file:///Users/d2r/example.ttl#> .
# Other namespaces
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
map:Database1 a d2rq:Database;
    d2rq:jdbcDSN "jdbc:mysql://localhost/iswc";
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:username "user";
    d2rq:password "password";
    .
# ------------------------------------------------
# CREATE TABLE Conferences (ConfID int, Name text, Location
text);
map:Conference a d2rq:ClassMap;
    d2rq:dataStorage map:Database1;
    d2rq:class :Conference;
    d2rq:uriPattern
"http://conferences.org/comp/confno@@Conferences.ConfID@@";
.
map:eventTitle a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:Conference;
d2rq:property :eventTitle;
    d2rq:column "Conferences.Name";
    d2rq:datatype xsd:string;.
map:location a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:Conference;
    d2rq:property :location;
    d2rq:column "Conferences.Location";
    d2rq:datatype xsd:string;
```

Gambar 2.6 *Template D2RQ Mapping*

Basisdata yang dipetakan dalam istilah RDF terbentuk sesuai dengan struktur D2RQ *mapping* seperti yang diilustrasikan pada gambar 2.7 , ditunjukkan di sebelah kanan dengan menggunakan `d2rq:ClassMap` dan `d2rq:PropertyBridge`.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Objek yang paling penting dalam pemetaan adalah *ClassMap*. *ClassMap* menentukan bagaimana URI yang digunakan untuk kelas *instance*.




Gambar 2.7 Struktur *D2RQ Mapping*

`d2rq:ClassMap` mempunyai beberapa properti yang tersusun sesuai dengan struktur D2RQ *mapping* seperti yang diilustrasikan pada tabel 2.2.

Tabel 2.2 Properti *ClassMap*

| Properti | Deskripsi |
|---|---|
| `d2rq:dataStorage` | Sebagai acuan terhadap `d2rq:Database` yaitu dimana contoh data disimpan |
| `d2rq:class` | Sebuah kelas RDF-S atau OWL. Semua sumber yang dihasilkan oleh *ClassMap* merupakan contoh kelas |
| `d2rq:uriPattern` | Menentukan pola URI yang akan digunakan untuk mengidentifikasi contoh peta kelas |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| Properti | Deskripsi |
|---|---|
| d2rq:uriColumn | Sebuah kolom database yang berisi URI untuk mengidentifikasi contoh peta kelas. Nama kolomnya harus dalam bentuk "TableName.ColumnName" |
| d2rq:uriSqlExpression | Sebuah ekspresi SQL yang menghasilkan pengidentifikasi URI untuk *instance* dari peta kelas ini. Mirip dengan d2rq:sqlExpression . Outputnya berupa URI yang valid. |
| d2rq:bNodeIdColumns | Daftar nama kolom yang dipisahkan koma dalam nota "TableName.ColumnName" |
| d2rq:constantValue | Peta kelas yang hanya memiliki satu contoh, yang diberi nama berdasarkan nilai propertinya |
| d2rq:additionalProperty | Menambahkan AdditionalProperty ke semua contoh kelas |
| d2rq:classDefinitionLabel | Menentukan label yang akan dijadikan rdfs:label untuk semua definisi kelas terkait |
| d2rq:classDefinitionComment | Menentukan komentar yang akan disajikan sebagai rdfs:comment untuk semua definisi kelas terkait. |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| | |
|---|---|
| d2rq:containsDuplicates | Menentukan peta kelas apabila menggunakan informasi dari tabel yang tidak sepenuhnya dinormalisasi. Jika d2rq:containsDuplicates nilai properti diatur ke "*true*", maka D2RQ menambahkan klausa DISTINCT ke semua query menggunakan *classMap* ini. "*false*" adalah nilai *default*, yang tidak harus dinyatakan secara eksplisit. Menambahkan properti ini ke peta kelas berdasarkan tabel *database* yang dinormalisasi menurunkan kinerja kueri, namun tidak memengaruhi hasil *queri* |
| d2rq:condition | Menentukan kondisi WHERE SQL. Sebuah *instance* dari kelas ini hanya akan dihasilkan untuk baris *database* yang memenuhi syarat. Kondisi dapat digunakan untuk menyembunyikan bagian database dari D2RQ, misal menolak akses ke data yang lebih tua atau lebih baru dari tanggal tertentu |

## 2.4 *User Interface Management System* (UIMS)

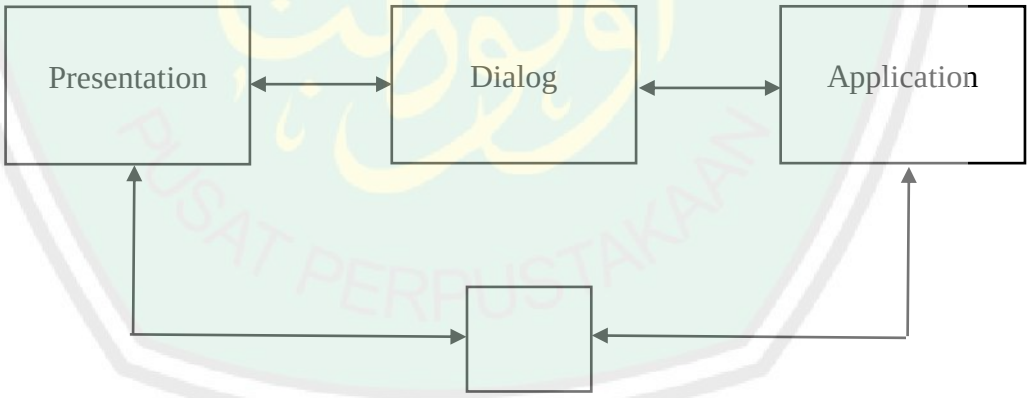
*Interface* merupakan media yang digunakan sebagai interaksi antara manusia dan komputer. *User interface* telah menjadi aspek penghubung antara pengguna dengan sistem, sehingga *user interface* menjadi peranan yang penting dalam sistem. *Interface* adalah hal utama yang dilihat oleh pengguna sehingga sistem harus bisa mengatur kerja dari sistem tersebut dengan pengguna yang saling berkaitan.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

*User Interface Management System* (UIMS) adalah sebuah mekanisme untuk proses memisahkan kode *Graphic User Interface* (GUI) dalam suatu program komputer. Tujuan dari UIMS yaitu menciptakan suatu cara agar bisa diperoleh *interface* yang konsisten yang memiliki "*look*" dan "*feel*" yang sama untuk sejumlah aplikasi yang berbeda namun di dalam sistem yang sama.

Menurut Raymond E. Barber dan Henry C. Lucas, JR dalam workshopnya di Seattle pada tahun 1982 menyimpulkan bahwa *user interface* diimplementasikan dengan menggunakan *tool* pemrograman tertentu dan terpisah dari kode aplikasi. Interaktif aplikasi harus lebih memiliki perangkat lunak sebagai pendukung dalam mengontrol aplikasi (*eksternal control*) daripada kode aplikasi (*internal control*).[19]

Menurut model Seeheim, UIMS memiliki tiga komponen, seperti yang diilustrasikan pada gambar 2.8.




Gambar 2.8 Arsitektur UIMS Model Seeheim

Arsitektur model Seeheim memiliki tiga komponen. Komponen utamanya adalah *control dialog*. *Control dialog* yaitu komunikasi yang terjadi antara pengguna dan aplikasi. Struktur *control dialog* kemudian diubah menjadi sebuah urutan input yang dikirimkan ke model aplikasi *user interface* untuk mengeksekusi

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

perintah dari sistem [20]. Presentation merupakan komponen yang bertanggungjawab terhadap *interface* yang mencakup *output* dan *input* yang tersedia bagi *user*. *Interface* model aplikasi yaitu komopnen yang menjelaskan tentang pandangan dari aplikasi *semantic* yang tersedia sebagai *interface*.

Model Seeheim memiliki fungsi sebagai *output* grafis aplikasi data yang dihasilkan oleh sebuah aplikasi dibawah *control manager dialog*. Model Seeheim merupakan *input* dari aplikasi dan pengguna dalam konteks sebuah system interaktif. Pada gambar 2.8 mengilustrasikan tentang komponen *control dialog* secara eksplisit sehingga menjadikan sebuah aplikasi yang bisa memberikan respon semantic aplikasi yang lebih besar. Kotak kosong yang menjadi penengah dan berada dibawah ada karena menunjukkan bahwa logika tersebut tidak dipisahkan dari implementasi.

*User Interface Management System* (UIMS) memberikan kapabilitas, diantaranya yaitu:

- Memberikan grafis pada *user interface*
- Menyajikan data dengan berbagai format dan output
- Memberikan interaksi antara database dan basis model
- Menyimpan data input dan output
- Memberikan grafis yang menarik
- Memiliki window yang berfungsi untuk ditampilkan
- Memberi interaksi dengan bermacam-macam gaya dialog
- Menangkap, menyimpan, dan menganalisis pemakaian dialog untuk peningkatan sistem dialog

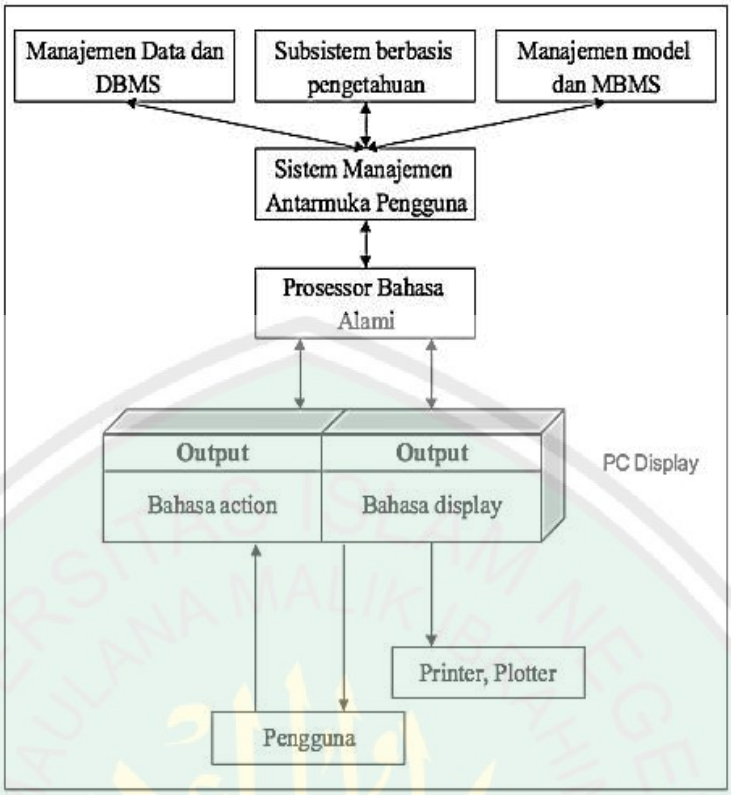
CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Fungsi UIMS yang sebagai pemisah antara semantik aplikasi dan presentasi atau disebut *software architectural pattern*, yaitu sebagai berikut :

- Portabilitas : aplikasi yang sama dapat digunakan pada system yang berbeda.
- Reusabilitas : meningkatkan komponen sehingga dapat digunakan kembali dengan mengurangi biaya dari tingkat kegunaan.
- *Multiple interface*s : meningkatkan fleksibilitas aplikasi yang interakstif, dan *interface* yang berbeda dapat memiliki fungsi yang sama dalam mengakses.
- Kustomisasi : *interface* dapat dikustom oleh *desainer* dan *user*, sehingga meningkatkan keefektifan dengan tanpa mengubah aplikasi.

UIMS juga disebut sebagai generasi dialog dan sistem manajemen. Proses *user interface* dalam *Management Support System* ditunjukkan pada gambar 2.9.

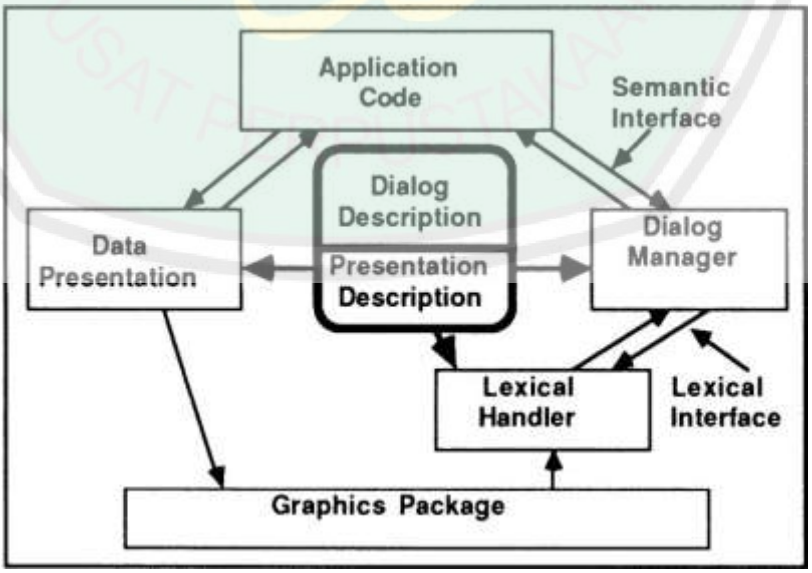
CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 2.9 Skema Sistem *User Interface*

### 2.3.1 Arsitektur UIMS

UIMS memiliki arsitektur dengan berbagai tingkatan seperti akan diilustrasikan pada **Gambar 2.10**.




Gambar 2. 10 Arsitektur UIMS

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Tingkat leksikal yang diliputi oleh *lexical handler* merupakan bagian masukan dari paket grafis, dan spesifikasi *lexical* dalam deskripsi presentasi. Tingkat sintaksis kemudian ditemukan *dialog manager* dan deskripsi dialog. Tingkat semantik terdiri dari apilikasi tersebut dan antarmuka semantik

Tingkat leksikal pada *user interface* secara umum merupakan satu set perangkat input yang logis. Perangkat logis ditandai dengan *interface* perangkat lunak dan jenis nilai masukannya. Salah satu bagian dari informasi yang harus ada dalam deskripsi presentasi adalah pemetaan fisik ke perangkat yang logis. Sebagai contoh, perangkat logis "*Delete*" mungkin dipetakan ke fisik menjadi "F1" sebagai bagian dari deskripsi presentasi.

*Interface* pada *lexical handler* terutama dengan paket grafis dan manajer dialog. Antarmuka antara manajer dialog dan *lexical handler* merupakan suatu yang menarik dan beberapa *property* perangkat logis telah diidentifikasi sebagai bagian dari proses desain. Kegiatan *lexical handler* memberikan fungsi pada dialog *manager* yaitu untuk memperoleh atau melepaskan perangkat logis, mengaktifkan dan menonaktifkan perangkat, dan sebagai sampel masukan.

Sebuah kunci untuk berbagai dialog model adalah hubungan mereka dengan kode aplikasi. Pemetaan untuk kode aplikasi dari UIMS adalah fitur yang memungkinkan penggunaannya dalam berbagai mcam pengaturan. Ada dua model utama yang digunakan untuk antarmuka semantik antara manajer dialog dan kode aplikasi. Ada beberapa variasi pada model perintah untuk antarmuka semantik. Model yang paling sederhana adalah sebuah perintah yang digunakan untuk dipanggil atau nama prosedur dalam sebuah kode aplikasi. Sebagai contoh yaitu,

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

tindakan *EnterPoint* yang akan memanggil *Uxical Handler* untuk mendapatkan sebuah salinan.

## 2.4 UI Ontologi

UI ontologi dibangun untuk mempublikasikan UI dalam sebuah UI *repository* yang menyimpan informasi semantik dengan baik. Selain itu, UI ontologi dapat memberikan konsep, hubungan, penalaran, dan mencari elemen yang terkait dalam UI. Sebuah ontologi harus mencakup informasi UI yang diklasifikasi lima kategori utama yaitu pengumpulan data, penyajian data, *monitoring*, *command and control*, dan *hybrid*.[21]

a. Pengumpulan data

   Pengumpulan data yaitu jenis UI yang mengumpulkan data dari *user* terakhir. Sebuah pengumpulan data UI memiliki alur kerja yang sederhana mengenai data yang dikumpulkan dalam sebuah *interface* karena data tersebut dapat distandarisasi dan dimodelkan dalam ontologi.

b. Penyajian data

   Penyajian data merupakan jenis UI yang menerima data dari aplikasi dan menyajikan kepada *user* terakhir dalam *format* tertentu seperi dalam bentuk tabel atau grafik.

c. *Monitoring*

   *Monitoring* yaitu jenis UI yang menyajikan data dalam *format* khusus untuk *user*.

d. *Command and Control*

   *Command and Control* merupakan jenis UI yang melakukan tugas-tugas interaktif dengan aplikasi. UI melakukan perintah ke aplikasi untuk memicu

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

beberapa tindakan dan menerima kode perintah untuk menghasilkan masukan ke aplikasi.

*e. Hybrid*

*Hybrid* yaitu jenis UI yang merupakan aplikasi dan memiliki beberapa tugas seperti pemantauan UI pada panel atas dan perintah kontrol UI dalam sebuah panel yang lebih rendah, dan pengumpulan data UI dalam panel samping.

## 2.5 UI *Discovery*

UI *Discovery* merupakan bagian penting dalam arsitektur *web service*. Apabila didalam sebuah *service* tidak dapat ditemukan dan tidak dapat dipanggil, UDDI menyediakan sebuah standarisasi dan *discovery* ontologi UI untuk pencarian kata kunci. Pencarian kata kunci tersebut tidak dapat diketahui pada UI *web service* apabila dengan kata kunci yang berbeda melainkan memiliki arti yang sama. Penyelasaian masalah tersebut dapat diselesaikan dengan cara *discovery* sistem untuk mencocokkan *semantic web service* menggunakan ontologi.

UI *discovery* adalah pencarian data *user interface* dengan inputan yang tepat sehingga menghasilkan output yang diinginkan oleh pengguna. Sehingga membantu pengguna dalam mendapatkan keseluruhan dokumen yang memberikan informasi secara lengkap sesuai kebutuhan pengguna.

## 2.6 ROC (*Receiver Operating Characteristic*)

ROC (*Receiver Operating Characteristic*) yaitu sebuah analisa yang digunakan untuk menafsirkan sensitivitas dan spesifisitas suatu tingkat dan nilai-nilai yang terkait sebagai ukuran akurasi pengujian. Analisis ROC berasal dari awal tahun 1950an dengan teori deteksi sinyal elektronik.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Penerapan metodologi ROC dalam radiologi diagnostik dan pencitraan radionuklida dimulai pada awal tahun 1960an. Kurva ROC pertama dalam radiologi diagnostik dihitung oleh Lusted (1960) yang menganalisis kembali data yang telah dipublikasikan sebelumnya mengenai deteksi tuberkulosis paru dan menunjukkan hubungan timbal balik antara presentase positif yang salah (*false positive*) dan hasil negatif yang salah (*false negative*) dari data yang berbeda. interpretasi film.


Gambar 2.11 ROC

Hasil dari implementasi dalam pengukuran akurasi hasil prediksi pada penelitian dilakukan dengan *confusion matrix*. ROC berfungsi untuk mengekspresikan *confusion matrix*. *Confusion matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Evaluasi dengan *confusion matrix* menghasilkan nilai akurasi, presisi dan *recall*.

Akurasi adalah presentasi ketepatan *record* data yang diklasifikasi secara benar setelah dilakukan pengujian pada hasil klasifikasi. Presisi atau *confidence* adalah proporsi kasus yang diprediksi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar.[22]

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Tabel 2.3 *Confusion Matrix*

| Matriks | Deskripsi |
|---|---|
| $True\ Positive\ Rate = \frac{TP}{TP + FN}$ | Rasio dari semua yang positif yang diklasifikasikan dengan benar dibagi dengan jumlah total yang positif |
| $True\ Negative\ Rate = \frac{TN}{TN + FP}$ | Rasio dari semua yang negatif yang diklasifikasikan dengan benar (baik yang positif maupun yang negatif) dibagi dengan jumlah total instans |
| $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$ | Rasio dari semua instans yang diklasifikasikan dengan benar (baik yang positif maupun yang negatif) dibagi dengan jumlah total instans |
| $Precision = \frac{TP}{TP + FP}$ | Rasio dari semua yang positif yang diklasifikasikan dengan benar dibagi dengan jumlah yang positif yang diklasifikasikan dengan benar dan jumlah yang positif yang diklasifikasikan dengan yang salah |
| $Recall = \frac{TP}{TP + FN}$ | Rasio dari semua yang positif yang diklasifikasikan dengan benar dibagi dengan jumlah yang positif yang diklasifikasikan dengan benar dan yang negatif yang diklasifiksikan dengan salah |

## 2.7 Penelitian Terkait

Tabel 2. 4 Penelitian Terkait

| | Penelitian | Pembahasan | Hasil |
|---|---|---|---|
| RDF | M. M. Ferdila, "Aplikasi Web Semantik Untuk Pencarian Materi Perkuliahan," *Univ. Gunadarma*, vol. 1, no. 10107686, 2012 | RDF (*Resource Description Framework*) dipergunakan sebagai representasi pengetahuan yang digunakan pada aplikasi web semantik untuk pencarian materi perkuliahan. | Aplikasi ini dirancang untuk menghasilkan keluaran yang mudah dimengerti oleh pemakai serta membantu dalam pencarian materi perkuliahan. |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| | Penelitian | Pembahasan | Hasil |
|---|---|---|---|
| | P. S. E. Primagamaplus, "Rancang Bangun Pembangkit Ontologi dan RDF pada Sistem *E-Learning* PrimagamaPlus," no. March 2015, 2017 | Pengelolaan web semantik bersifat otomatis, yaitu setiap konten yang ditambahkan, diubah, atau dihapus maka *file* RDF untuk konten tersebut akan mengalami perlakuan serupa. | Berkas RDF hasil pembangkitan telah sesuai dengan konsep *Triple* RDF karena telah teruji dengan validator RDF. |
| | S. Rikacovs and J. Barzdins, "*Export of Relational Databases to RDF Databases : A Case Study*," Proc. 9th Int. Conf. Perspect. Bus. Informatics Res. (BIR 2010), pp. 203–211, 2010 | Tampilan RDF dengan menggunakan tabel untuk *class* RDFS dan kolom sebagai pendekatan predikat dalam mengambil pertimbangan kasus-kasus tertentu. RDB data direpresentasikan sebagai virtual grafik RDF tanpa RDF *dataset.* | Bahasa *query* SPARQL mendukung notasi SPARQL yang terjadi pada RDF. |
| UIMS | W. Buxton, M. R. Lamb, D. Sherman, and K. C. Smith, "Towards a comprehensive User Interface Management System," *Comput. Graph. (ACM).*, vol. 17, pp. 35–42, 1983 | Penelitian ini menyajikan UIMS yang dikembangkan pada sistem komputer untuk merancang dan mengimplementasikan dialog berbasis menu dan untuk menyediakan dukungan *run-time* untuk program-program interaktif yang lebih bersifat *event-driven*. . | Hasil dari penelitiannya yaitu untuk merancang atau menerapkan dialog berbasis menu memiliki properti memungkinkan program yang ditulis tangan terintegrasi dengan kode yang secara otomatis disintesis dari spesifikasi *programmer.* |
| | J. Foley, C. Gibbs, W. C. Kim, and S. Kovacevic, "A knowledge-based user interface management system," *Proc ACM CHI88 Hum. Factors* | Pada penelitian ini telah mengimplementasikan UIMS dengan cara mengembangkan *User Interface Design Environment* | Desainer diizinkan untuk bekerja pada tingkat abstraksi yang lebih tinggi. Pembuat interface dapat meminta IMS untuk menerapkan |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| | | | |
|---|---|---|---|
| | *Comput. Syst. Conf.*, vol. 87, no. 15, pp. 67–72, 1988 | (UIDE) sebagai sebuah system yang digunakan untuk membantu dalam desain *user interface*. | sintaks seperti awalan, postfix,atau "nofix" sehingga menghilangkan kekhawatiran tentang spesifikasi *syntax* yang tidak konsisten |
| Akurasi | C. Matrix, "Jurnal String Vol . 1 No . 1 Tahun 2016 ISSN : 2527 – 9661 Analisa Model *Support Vector Machine Textmining* Pada Komentar Positif Dan Negatif Untuk Review perbandingan Whatsapp Vs Bbm Pendahuluan ISSN : 2527 – 9661 Tinjauan Pustaka," vol. 1, no. 1, pp. 74–82, 2016. | Dalam penelitian ini dilakukan pengujian model dengan menggunakan *Support Vector Machine* dengan menggunakan data komentar *review* produk smartphone android dan *blackberry* yang positif maupun negatif. Model yang dihasilkan diuji untuk mendapatkan nilai *accuracy, precision, recall* dan AUC dari setiap algoritma sehingga didapat pengujian dengan menggunakan *Support Vector Machine* nilai *accuracy* adalah 71.00% | Dengan demikian dari hasil pengujian model diatas dapat disimpulkan bahwa *Support Vector Machine* dapat memberikan pemecahan untuk permasalahan klasifikasi komentar *review* perbandingan *whatsapp* dan BBM lebih akurat |
| | D. M. Informatika, F. I. Terapan, U. Telkom, J. Telekomunikasi, and B. Batu, "Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining," no. November, pp. 2–3, 2015. | Hasil evaluasi dan validasi pada penelitian tersebut menggunakan *confusion matrix* serta kurva ROC untuk menunjukkan tingkat diagnosa *poor classification* | Penelitian menunjukkan bahwa tidak terdapat hubungan dan pengaruh yang kuat terhadap nilai proyek akhir mahasiswa berdasarkan pencapaian mereka pada mata kuliah pendukung proyek akhir |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pada penelitian ini penulis membuat aplikasi *User Interface* (UI) *discovery* dengan menggunakan pendekatan ontologi dengan *query* SPARQL. Aplikasi UI *discovery* bertujuan untuk memudahkan *user* dalam membuat sebuah aplikasi dan mengotomatiskan sebuah pencarian *form*. Menentukan akurasi *query* pada *interface repository* dengan menggunakan model *confusion matrix*.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# BAB 3

# METODOLOGI PENELITIAN

## 3.1 Desain Penelitian

Desain penelitian menjelaskan tentang alur dan mekanisme penelitian yang akan dilakukan meliputi tipe penelitian, prosedur penelitian, dan metode pengolahan data. Berikut penjelasan dari masing-masing mekanisme penelitian.

### 3.1.1 Gambaran Umum Sistem

Sistem yang dibangun merupakan sistem yang berfungsi untuk melakukan pencarian *user interface* pada sebuah aplikasi secara otomatis dengan tujuan untuk memudahkan *user* dalam memilih *interface* sebuah *form* yang diinginkan. Pencarian yang pertama yaitu dengan memasukkan kata kunci nama sebuah *form*, kemudian pencarian tersebut diolah menggunakan *query* SPARQL. Pencarian kedua yaitu melakukan pencarian secara detail dengan cara memasukkan kata kunci salah satu nama komponen dan jumlah yang diinginkan oleh *user*.

### 3.1.2 Sumber Data

Data yang digunakan diperoleh dari penelitian sebelumnya tentang *web service* pada ERP Pondok Pesantren di bidang akademik yang digunakan sebagai bahan penelitian. Data yang diolah yaitu *file* PHP yang berisi *interface* pada sebuah *form*. *File* tersebut kemudian diparsing untuk mengambil komponen *interface*.

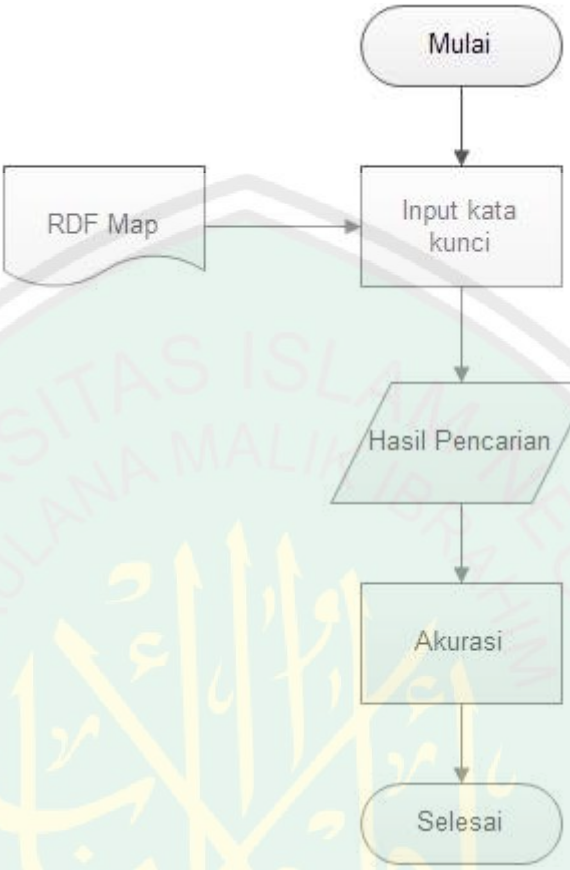
## 3.2 Perancangan Sistem

Perancangan sistem digunakan untuk mempermudah dan menentukan bagaimana sistem akan menyelesaikan masalah dalam penelitian. Pada tahap ini akan membahas semua hal yang terkait dengan perancangan aplikasi yang akan



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

dibuat dalam penelitian ini. Rancangan desain aplikasi PHP, desain D2R *server* untuk pencarian menggunakan SPARQL *query*.




Gambar 3.1 Alur Sistem Pencarian

Pada penelitian ini sistem pencarian dilakukan dengan memasukan objek inputan yang mengandung makna nama *form user interface*. Apabila objek keluar dari makna nama *form* tersebut, maka sistem tidak dapat menemukan data pencarian yang diinginkan oleh pengguna. Berdasarkan gambar tersebut bisa dilihat bahwa data yang diinputkan telah diproses oleh semantik web dengan bantuan D2R *Server* yang digunakan sebagai akses data dari DBMS MySql menggunakan RDF *Map* bertipe *turtle* (.ttl).

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Proses penampilan data dari D2R *server* menggunakan bantuan *query* SPARQL dengan tujuan sistem akan menyaring kata pencarian sesuai kata kunci yang dimasukkan sesuai dengan desain ontologi yang dibuat.

### 3.2.1 Desain *Database*

Aplikasi ini menggunakan *database* dengan nama ui_repository yang berisi tabel *form* dan tabel komponen. Tabel yang digunakan pada aplikasi UI *discovery* ini yakni saling terhubung antara tabel *form* dengan tabel komponen, seperti pada gambar 3.2.


Gambar 3.2 Desain *Database* UI *Discovery*

a. Tabel data *form*

Berfungsi untuk menyimpan identitas *form* yang berupa nama, lokasi *file*, dan *source code* atau isi dari *form* tersebut seperti yang diilustrasikan pada tabel 3.1..

Tabel 3. 1 Struktur Tabel *Form*

| Nama Kolom | Tipe Data | Lebar | Keterangan |
|---|---|---|---|
| id_*form* | Char | 35 | *Primary_key* |
| nama | Varchar | 100 | |
| url | Varchar | 100 | |
| *path_code* | Text | | |
| *code* | Text | | |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

b. Tabel Data Komponen

Berfungsi untuk menyimpan data atribut elemen *interface*, yaitu nama komponen, dan ID komponen. Keterangan tabel komponen dapat dilihat pada tabel 3.2.

Tabel 3. 2 Struktur Tabel Komponen

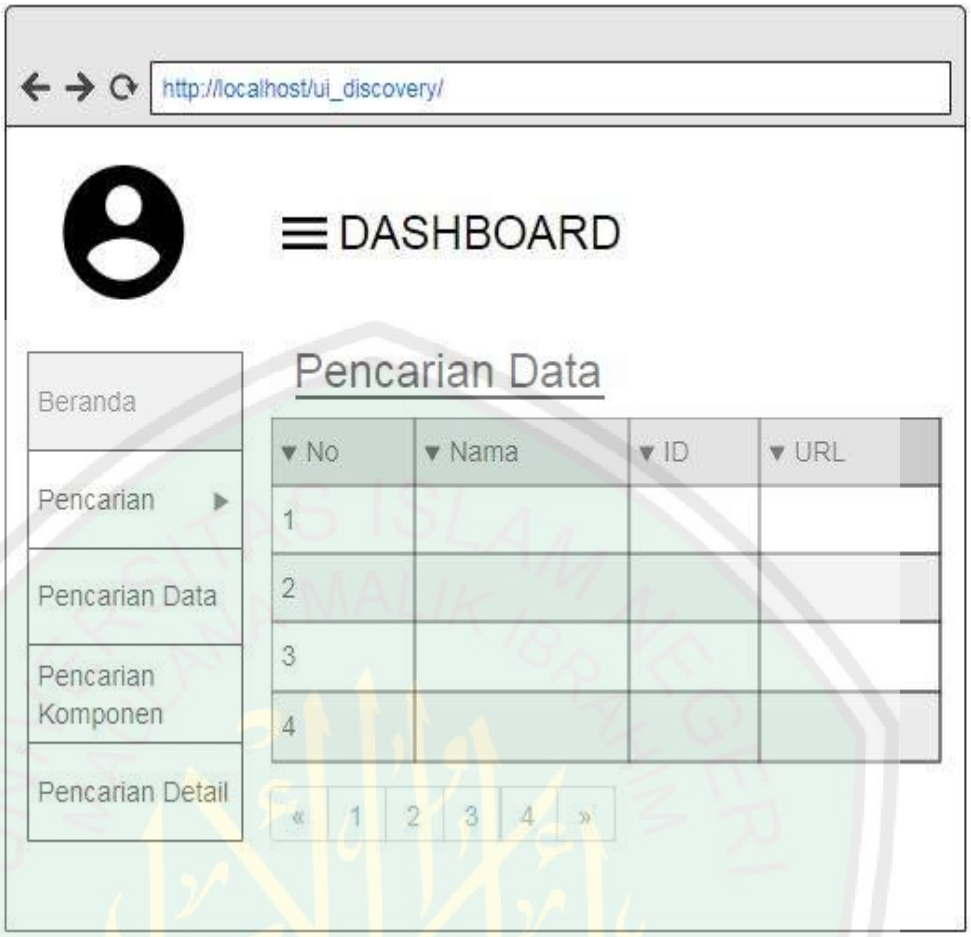
| Nama Kolom | Tipe Data | Lebar | Keterangan |
|---|---|---|---|
| id_komponen | Char | 5 | *Primary_key* |
| id_*form* | Char | 35 | *Foreign_key* |
| nama_komponen | Varchar | 50 | |
| Id | Varchar | 50 | |

### 3.2.2 Desain *Interface*

a. Halaman Beranda

Desain halaman beranda aplikasi ini berisi data *form* yang sesuai dengan basisdata. Rancangan tersebut berisi data dalam bentuk tabel dengan empat kolom dan lima baris pada halaman pertama seperti pada gambar 3.3.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

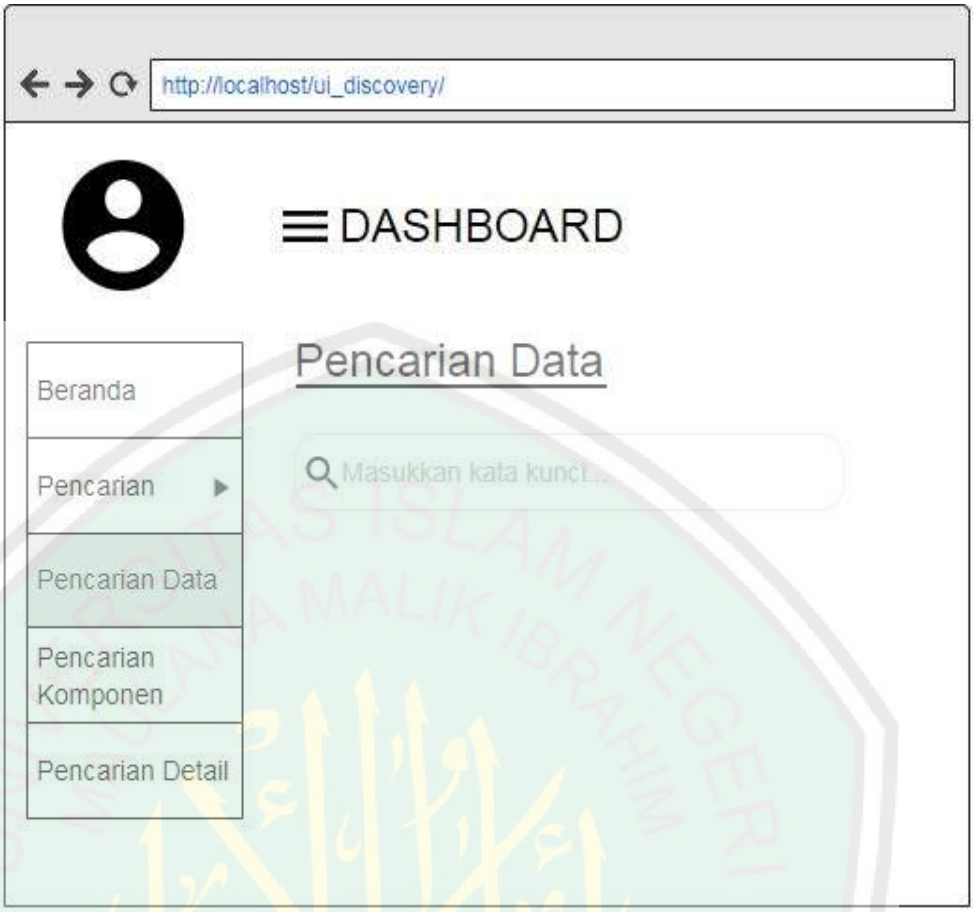



Gambar 3.3 Desain *Interface* Halaman Beranda

b. Halaman Pencarian Data

Desain *interface* halaman pencarian data merupakan rancangan yang akan dibangun untuk menu pencarian pada aplikasi *user interface discovery*. Pada halaman pencarian, *user* diperintahkan untuk memasukkan kata kunci berdasarkan nama *form* yang diinginkan, seperti pada gambar 3.4.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

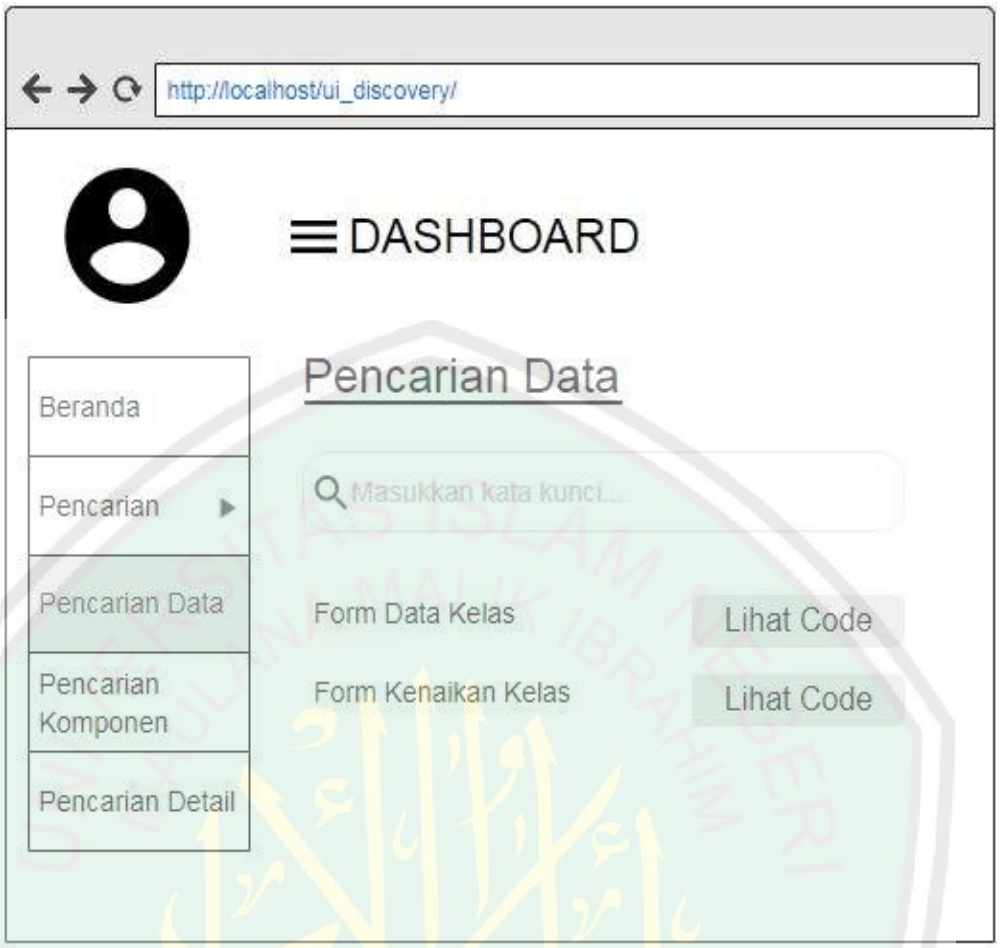



Gambar 3.4 Desain *Interface* Halaman Pencarian Data

c. Halaman Hasil Pencarian Data

Desain *interface* halaman hasil pencarian ialah suatu rancangan untuk menampilkan hasil dari menu pencarian. Hasil dari pencarian tersebut akan menampilkan nama *form* dan *code* dari *form* tersebut seperti yang diilustrasikan pada gambar 3.5.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

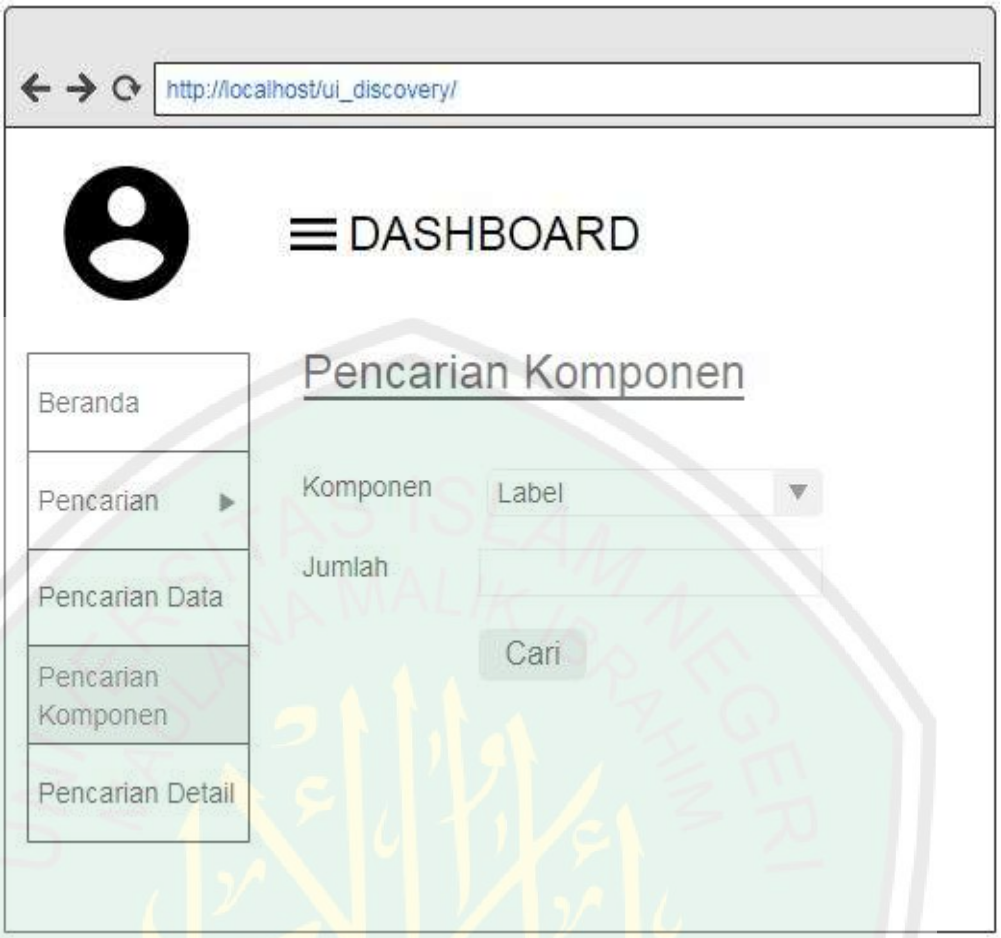



Gambar 3.5 Desain *Interface* Halaman Hasil Pencarian Data

d. Halaman Pencarian Komponen

Rancangan halaman pencarian komponen yaitu berisi list menu komponen untuk dipilih *user* sebagai kata kunci nama komponen yang diinginkan dan sebuah textfield yang berguna untuk inputan angka yang dibutuhkan dari komponen seperti pada gambar 3.6.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

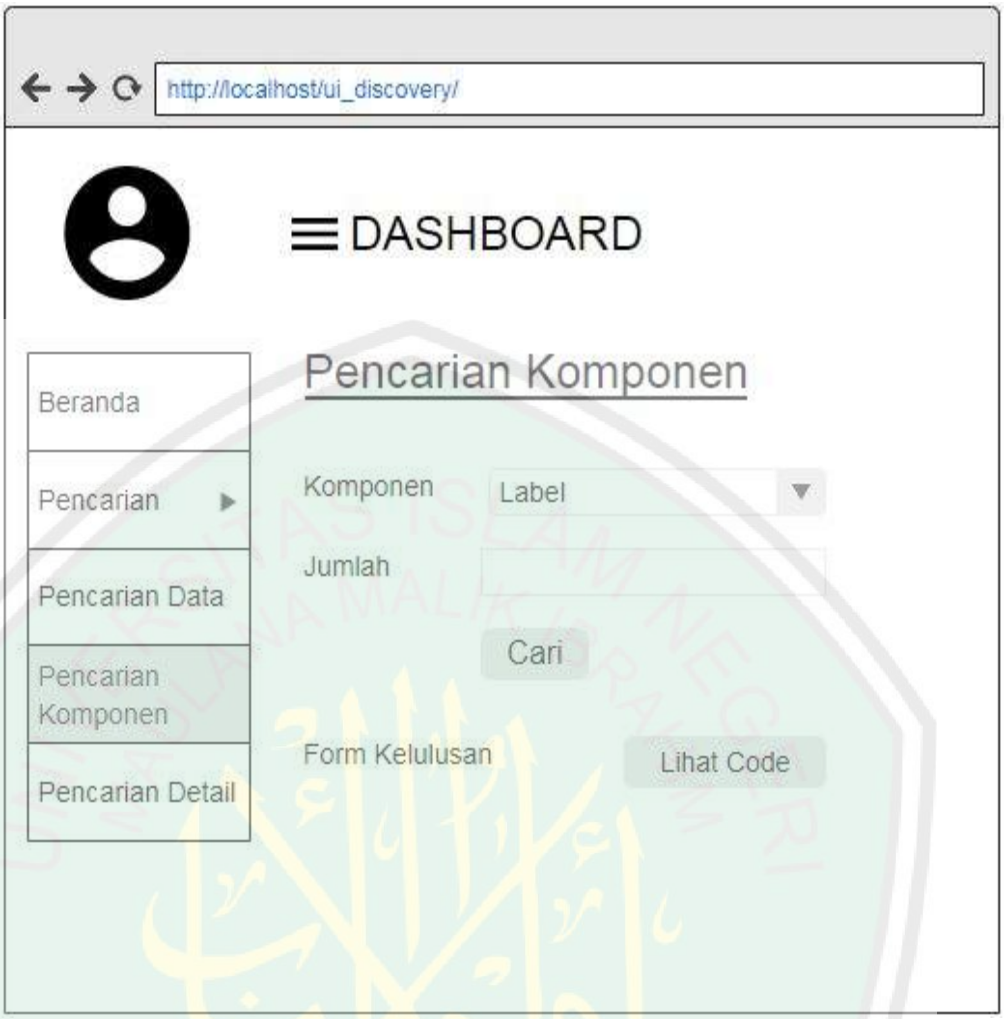



Gambar 3.6 Desain *Interface* Halaman Pencarian Komponen

e. Halaman Hasil Pencarian Komponen

Desain *interface* halaman hasil pencarian komponen merupakan rancangan *output* dari halaman pencarian komponen yang menampilkan nama *form* dengan *code* sesuai kata kunci yang dimasukkan seperti yang diilustrasikan pada gambar 3.7.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

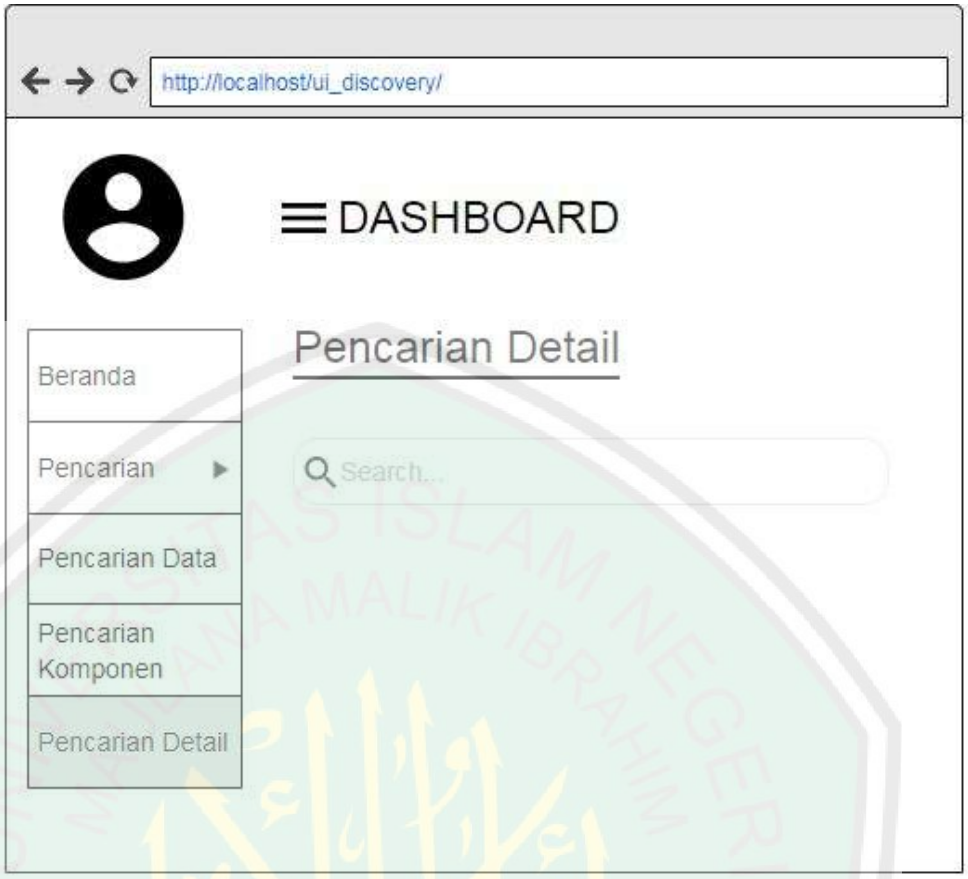



Gambar 3.7 Desain *Interface* Halaman Hasil Pencarian Komponen

f. Halaman Pencarian Detail

Desain *interface* halaman pencarian detail merupakan rancangan yang akan dibangun untuk menu pencarian pada aplikasi *user interface discovery*. Pada halaman pencarian tersebut, *user* diperintahkan untuk memasukkan kata kunci apapun, seperti pada gambar 3.8.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

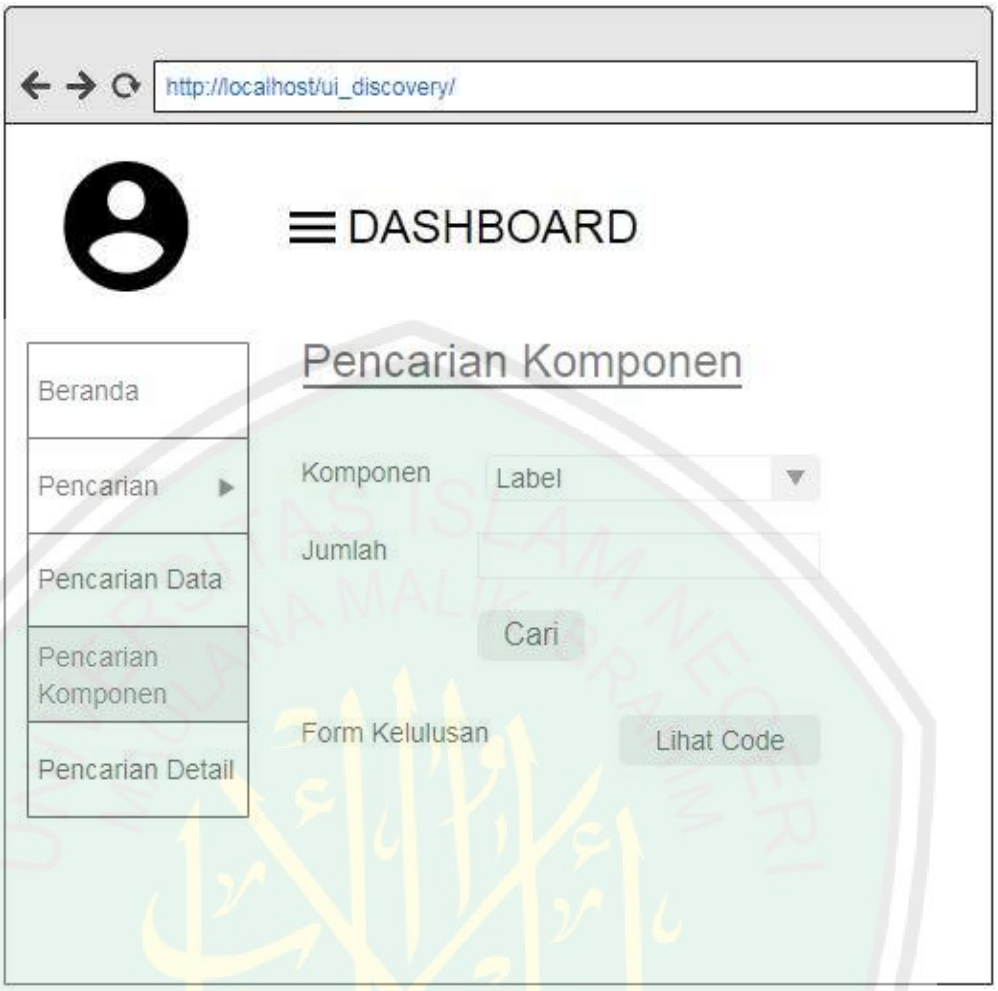



Gambar 3.8 Desain *Interface* Halaman Pencarian Detail

g. Halaman Hasil Pencarian Detail

Desain *interface* halaman hasil pencarian ialah suatu rancangan untuk menampilkan hasil dari menu pencarian. Hasil dari pencarian tersebut akan menampilkan nama *form* dan *code* dari *form* tersebut seperti yang diilustrasikan pada gambar 3.9.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 3.9 Desain I*nterface* Halaman Pencarian Detail

### 3.2.3 Desain Proses

Ada beberapa proses yang dilakukan aplikasi ini dalam pencarian *interface* yakni pembuatan RDF *mapping* dan pembuatan *query* SPARQL, berikut adalah penjelasan tentang proses tersebut.

a. RDF *Mapping*

RDF *Mapping* adalah suatu ontologi dari semantik web yang digunakan sebagai perantara dalam mengakses data pada *database relational*. Dalam penelitian ini data yang digunakan adalah data *web service* pondok pesantren di

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

bidang akademik. Langkah awal sebelum memperoleh *file* RDF yaitu melakukan *generate* dari *database* MySQL seperti gambar 3.10.

```
generate-mapping \n
-u <nama_pengguna_akses_ke_dbms? \n
-p <password akses_ke_dbms> \n
-o <nama_file_output_mapping_file> \n
Jdbc:<dbms_driver>://<alamat_server_basisdata>
/<nama_basisdata>
```

Gambar 3.10 Pembuatan RDF *Mapping*

Hasil dari proses *mapping* tersebut menghasilkan sebuah *file* RDF dalam *format Turtle* (.ttl). Setelah itu, *database* pada SQL dapat diakses melalui *query* SPARQL dengan D2RQ sebagai *tool*. Pada gambar 3.11 menjelaskan tentang pendekatan pencarian menggunakan RDF *mapping*.

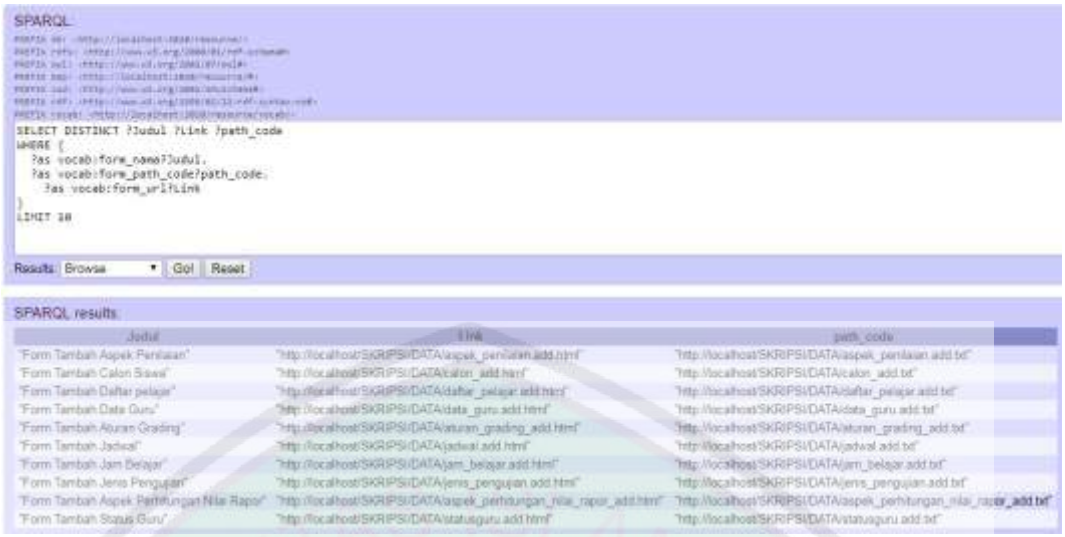

Gambar 3.11 Skema RDF *Mapping*

Setelah melakukan proses *mapping*, *user* dapat mengakses data hasil konversi dari model relasional ke RDF menggunakan D2R *Server* melalui *browser*.

b. SPARQL

SPARQL merupakan bahasa *query* yang digunakan untuk data RDF. Sintaks dalam *query* SPARQL yaitu *select*, *where*, dan *from*. Pada penelitian ini menggunakan tiga variabel yaitu nama *form*, *URL*, dan *path_code*. Penulisan variabel bagian *select* diawali dengan tanda tanya (?). Pada gambar 3.12 menjelaskan tentang cara penulisan *query* SPARQL.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 3.12 Penulisan *Query* SPARQL

### 3.3 Pengujian Sistem

Pada sistem informasi *user interface discovery* akan melakukan pengukuran dengan perhitungan terhadap nilai *Precision* (ketepatan) dan *Recall* (perolehan). Peneliti menggunakan nilai *Precision* dan *Recall* untuk menguji ketepatan *query* dalam melakukan proses pencarian ketika *user* menginputkan kata kunci.

Ukuran *recall* dan *precision* ini juga bergantung pada apa yang sesungguhnya dimaksud dengan "dokumen yang relevan" itu dan bagaimana memastikan relevan tidaknya dokumen.

Menurut Kurniawan [23] *Recall* adalah perbandingan jumlah dokumen relevan yang terambil sesuai dengan *query* yang diberikan dengan total kumpulan dokumen yang relevan dengan *query*. *Precision* adalah perbandingan jumlah dokumen yang relevan terhadap *query* dengan jumlah dokumen yang terambil dari hasil pencarian. *Precision* dapat diartikan sebagai ketepatan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu).

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

ROC digunakan untuk menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confusion matrix*. ROC adalah grafik dua dimensi dengan *false positive* sebagai garis horizontal dan *true positive* sebagai garis vertikal. Model *confusion matrix* ditunjukkan seperti pada gambar 3.13.

| | | Nilai Sebenarnya | |
|---|---|---|---|
| | | TRUE | FALSE |
| Nilai Prediksi | TRUE | TP<br>(True Positive)<br>Hasil relevan | FP<br>(False Positive)<br>Hasil tidak relevan |
| | FALSE | FN<br>(False Negative)<br>Hasil tidak ditemukan | TN<br>(True Negative)<br>Hasil relevan yang tidak ditemukan |

Gambar 3.13 Model *Confusion Matrix*

$$Precision = \frac{TP}{TP+FP} \quad (3.1)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.2)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.3)$$

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# BAB 4

# HASIL DAN PEMBAHASAN

## 4.1 Implementasi

Tahap ini menjelaskan tentang implementasi antarmuka aplikasi *user interface discovery* pada *interface* repository. Penelitian ini membuat sebuah aplikasi yang dibangun dengan menggunakan bahasa PHP. Sedangkan basisdata yang digunakan untuk membangun aplikasi *user interface discovery* yaitu menggunakan *query* MySQL dan SPARQL.

### 4.1.1 Implementasi Desain *Interface*

1. Desain *Input*

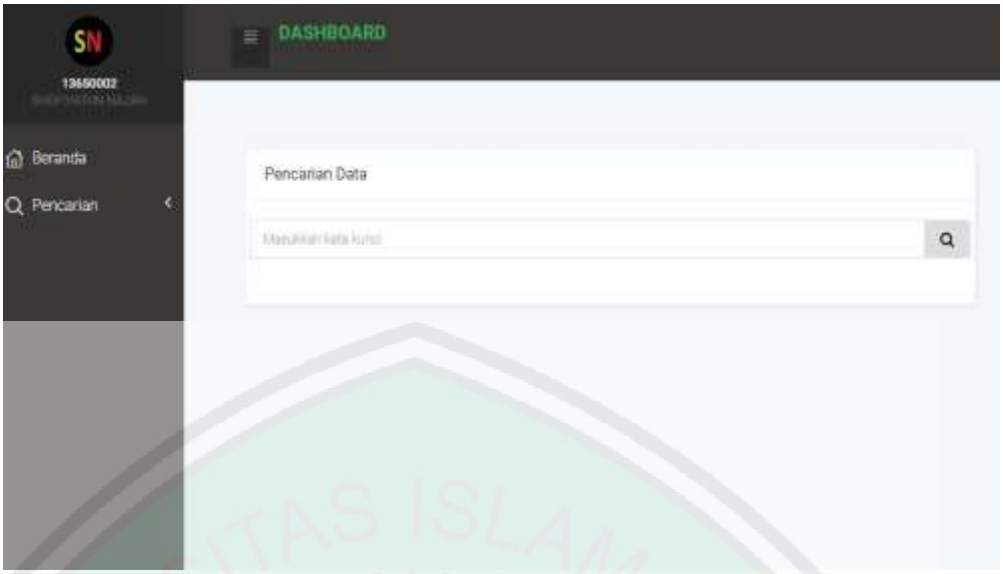
   Implementasi pada sesain input menjelaskan tentang rancangan desain input pada aplikasi *user interface discovery* yaitu halaman pencarian data, halaman pencarian komponen, dan halaman pencarian detail.

   a. Halaman Pencarian Data

   Halaman pencarian adalah sebuah halaman yang memproses pencarian dengan inputan nama *form* yang ingin dicari seperti yang diilustrasikan pada gambar 4.1.



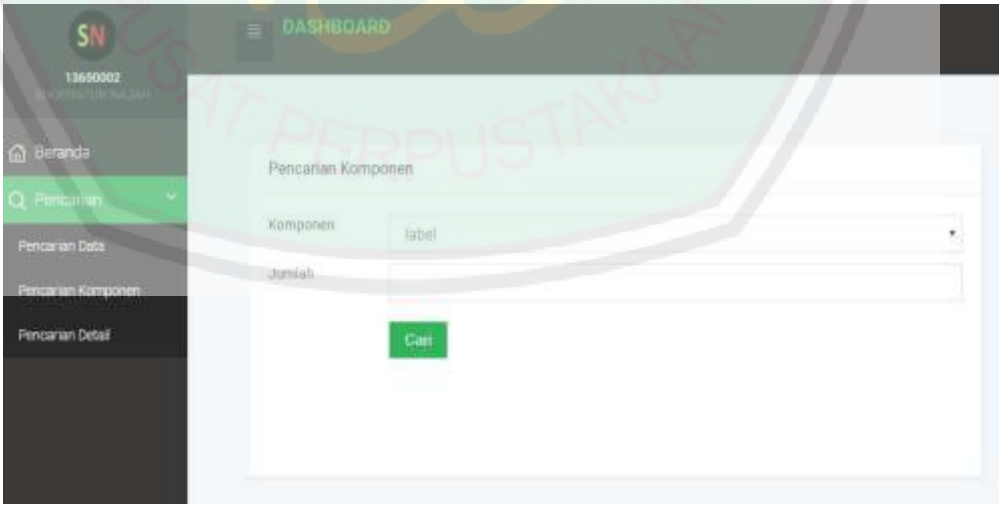
CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 4.1 Halaman Pencarian Data

b. Halaman Pencarian Komponen

Halaman pencarian komponen yaitu halaman yang memerlukan inputan berupa nama komponen dan jumlah yang diinginkan. Pada halaman ini, bertujuan untuk memudahkan pengguna untuk memilih *form* yang sesuai dengan komponen dan jumlah yang dibutuhkan oleh pengguna, seperti pada gambar 4.2.




Gambar 4.2 Halaman Pencarian Komponen

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

c. Halaman Pencarian Detail

Halaman pencarian detail adalah sebuah halaman yang memproses pencarian dengan inputan nama *form* atau data lainnya yang ingin dicari seperti yang diilustrasikan pada gambar 4.3.


Gambar 4.3 Halaman Pencarian Detail

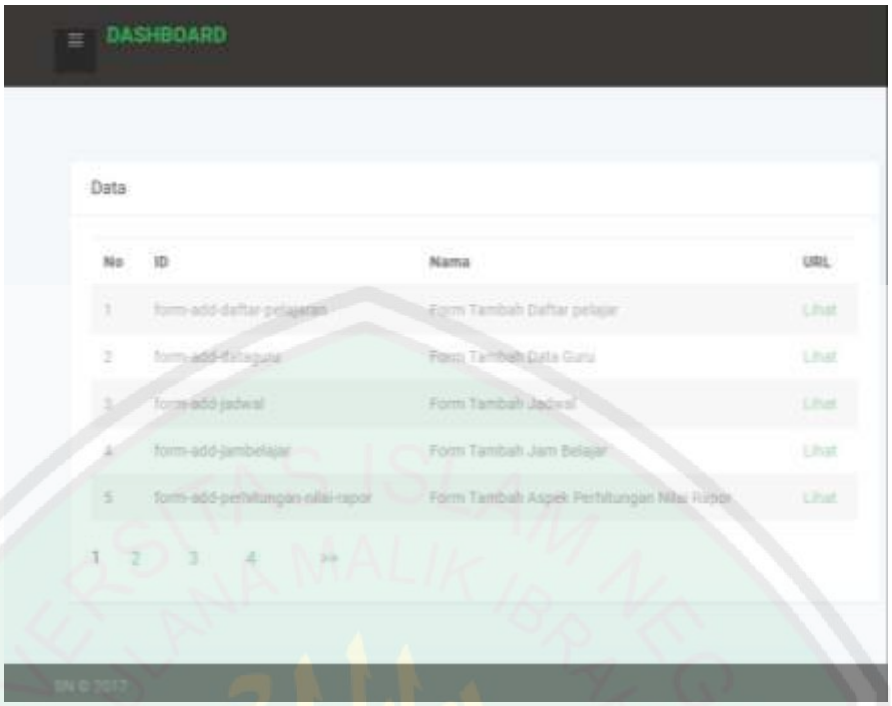
2. Desain *Output*

Implementasi desain output menjelaskan tentang rancangan desain output pada aplikasi *user interface discovery*, ada beberapa implementasi desain output pada aplikasi ini, yaitu halaman beranda, halaman pencarian, dan halaman pencarian detail.

a. Halaman Beranda

Halaman beranda merupakan halaman yang menampilkan data *form* yang tersimpan pada basisdata, seperti pada gambar 4.4.

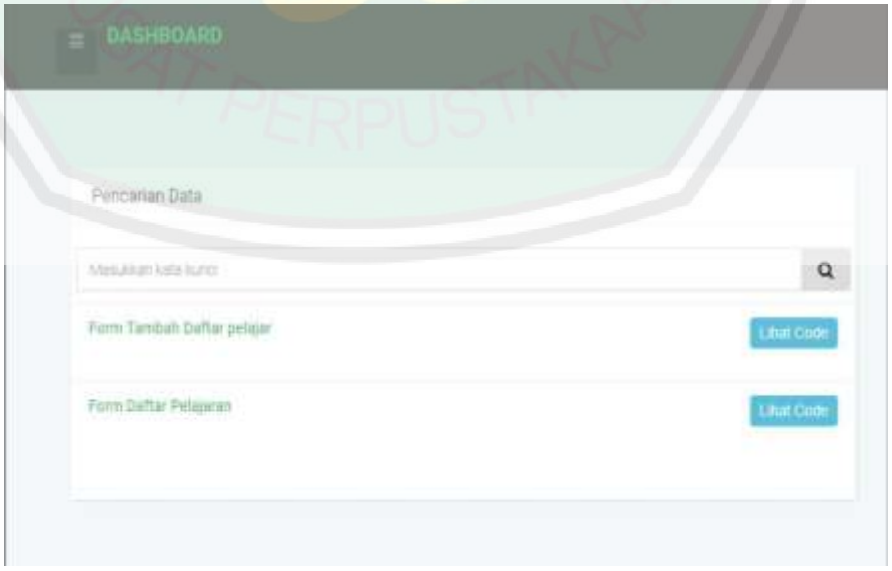
CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 4.4 Halaman Beranda

b. Halaman Pencarian Data

Halaman pencarian data ini merupakan hasil dari inputan pada pencarian data yaitu menampilkan nama *form* dan *pathcode* dari *form* tersebut seperti yang diilustrasikan pada gambar 4.5.




Gambar 4.5 *Output* Halaman Pencarian

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

c. Halaman Pencarian komponen

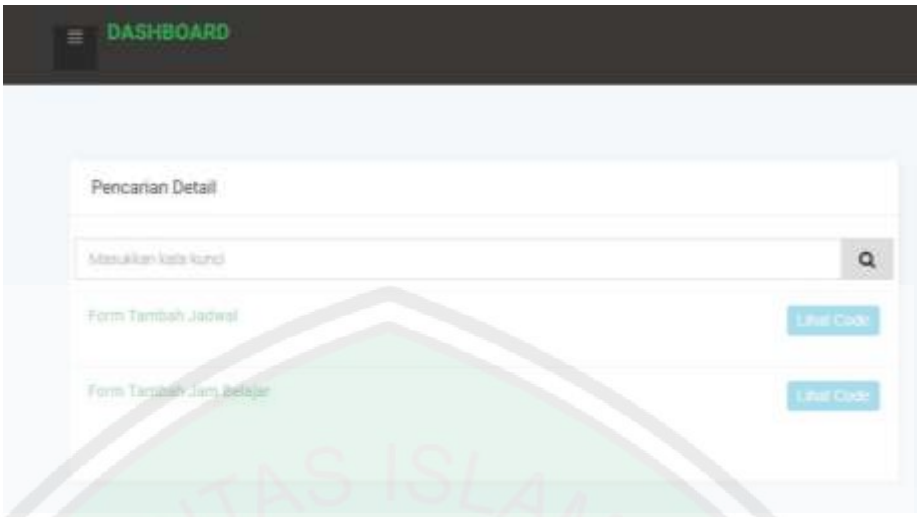
Halaman ini merupakan hasil dari inputan pencarian detail yang menampilkan nama *form* dan *pathcode* sesuai dengan nama komponen dan jumlah yang diinputkan seperti pada gambar 4.6.


Gambar 4.6 *Output* Halaman Pencarian Komponen

b. Halaman Pencarian Detail

Halaman pencarian detai ini merupakan hasil dari inputan pada pencarian detail yang termasuk dari nama *form*, ID *form*, atau *code* dari sebuah *form* yaitu menampilkan nama *form* dan *pathcode* dari *form* tersebut seperti yang diilustrasikan pada gambar 4.7.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 4. 7 *Output* Pencarian Detail

### 4.1.2 Implementasi Semantik Web

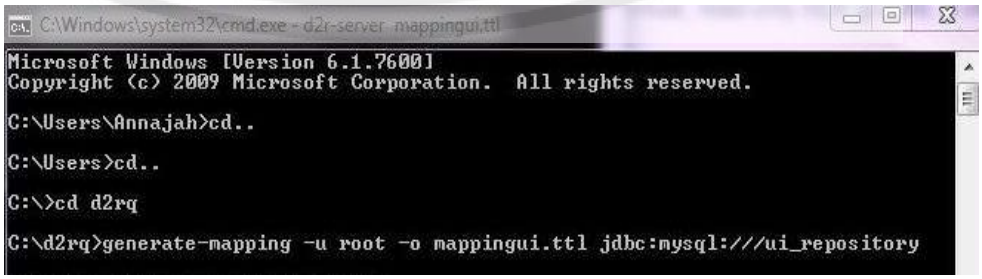
Implementasi semantik web ini akan membahas metode yang dirancang pada aplikasi ini, yaitu implementasi RDF, implementasi *query* SPARQL.

1. Implementasi RDF

   Pembuatan RDF *Map* dengan D2R *server* yaitu dilakukan dengan menggunakan command. Langkah-langkah dalam pembuatan RDF *Map* yaitu sebagai berikut:

   a. Masuk pada *directori* D2R *server*

   b. Untuk membuat *file* RDF *Map*, maka ketikkan "generate-mapping –u root –o mappingui.ttl jdbc:mysql:///ui_repository " , seperti pada gambar 4.8.




Gambar 4.8 Proses Pembuatan RDF *Map*

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pada gambar 4.7 terdapat perintah "generate-mapping" yaitu perintah *mapping* data. Terdapat perintah "-u root -o mappingui.ttl" adalah output dari mapping dan menggunakan sintaks *turtle* (.ttl). Perintah "jdbc:mysql:///ui_repository" adalah pengambilan data dari database ui_repository. Berikut ini adalah RDF Map dari database ui_repository yang disimpan dalam *format turtle* seperti yang diilustrasikan pada gambar 4.9.

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <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.wiwiss.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:///ui_repository";
        d2rq:username "root";
        jdbc:autoReconnect "true";
        jdbc:zeroDateTimeBehavior "convertToNull";.

# Table form
map:form a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "form/@@form.id_form|urlify@@";
        d2rq:class vocab:form;
        d2rq:classDefinitionLabel "form";.
map:form__label a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
        d2rq:property rdfs:label;
        d2rq:pattern "form #@@form.id_form@@";
map:form_id_form a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
        d2rq:property vocab:form_id_form;
        d2rq:propertyDefinitionLabel "form id_form";
        d2rq:column "form.id_form";    .
map:form_nama a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
        d2rq:property vocab:form_nama;
        d2rq:propertyDefinitionLabel "form nama";
        d2rq:column "form.nama";.
map:form_url a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
        d2rq:property vocab:form_url;
d2rq:propertyDefinitionLabel "form url";
        d2rq:column "form.url";.
map:form_path_code a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
```

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

```
        d2rq:property vocab:form_path_code;
        d2rq:propertyDefinitionLabel "form path_code";
        d2rq:column "form.path_code";.
map:form_file a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:form;
        d2rq:property vocab:form_file;
        d2rq:propertyDefinitionLabel "form file";
        d2rq:column "form.file";
        .
# Table komponen
map:komponen a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern
"komponen/@@komponen.id_komponen|urlify@@";
        d2rq:class vocab:komponen;
        d2rq:classDefinitionLabel "komponen";  .
map:komponen__label a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:komponen;
        d2rq:property rdfs:label;
        d2rq:pattern "komponen #@@komponen.id_komponen@@";
        .
map:komponen_id_komponen a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:komponen;
        d2rq:property vocab:komponen_id_komponen;
        d2rq:propertyDefinitionLabel "komponen
id_komponen";
        d2rq:column "komponen.id_komponen";
        .
map:komponen_id_form a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:komponen;
        d2rq:property vocab:komponen_id_form;
        d2rq:propertyDefinitionLabel "komponen id_form";
        d2rq:column "komponen.id_form";
        .
map:komponen_nama_komponen a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:komponen;
        d2rq:property vocab:komponen_nama_komponen;
        d2rq:propertyDefinitionLabel "komponen
nama_komponen";
        d2rq:column "komponen.nama_komponen";
        .
map:komponen_id a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:komponen;
        d2rq:property vocab:komponen_id;
        d2rq:propertyDefinitionLabel "komponen id";
        d2rq:column "komponen.id";
```

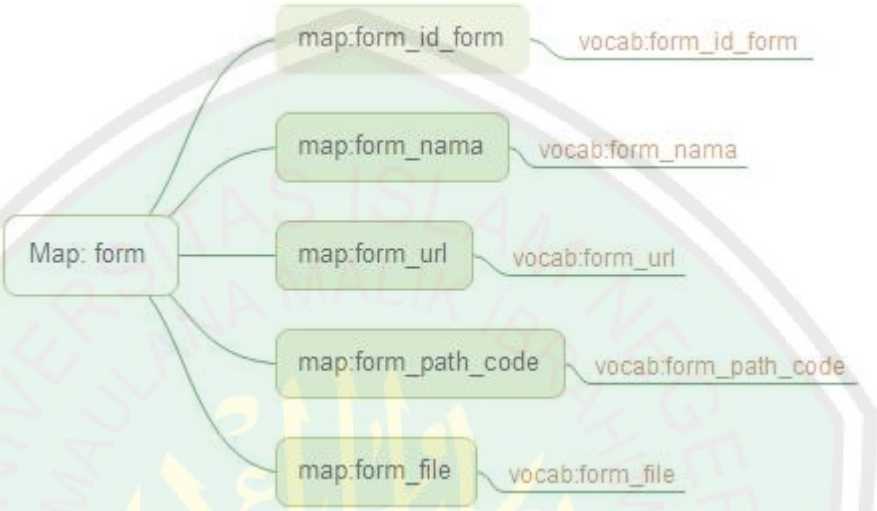
Gambar 4.9 RDF *Map* UI *Repository*

`Map: database` digunakan untuk melakukan koneksi ke database ui_repository, pada *mapping* tersebut terdapat satu buah `map: database`. `Map:ClassMap` berfungsi sebagai pendefinisian kolom pada *database*. Setiap

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

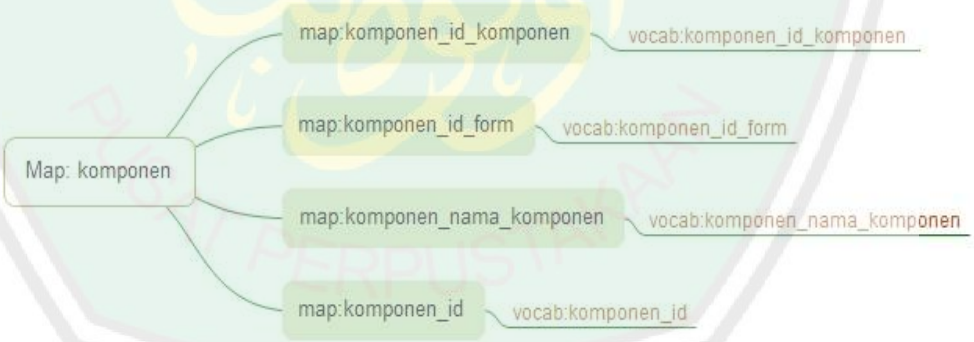
classMap mempunyai *Property Bridge*. *Property Bridge* berfungsi untuk menginisialisasikan setiap kolom yang terdapat dalam satu tabel. `Vocab:` yaitu *property bridge* yang berfungsi sebagai variabel dalam *query* SPARQL.

a. RDF Skema pada tabel form




Gambar 3.14 Skema RDF pada tabel barang

b. RDF Skema pada tabel komponen



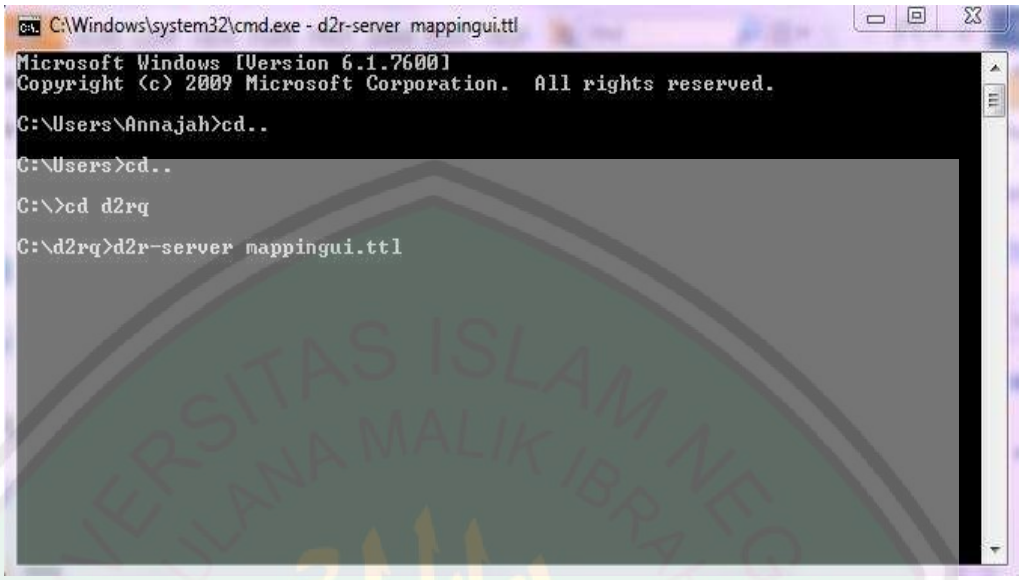

Gambar 3.15 Skema RDF pada tabel komponen

2. Implementasi SPARQL

*Query* SPARQL digunakan untuk mengakses *database* dengan alat D2RQ. Sebelum mengqueri SPARQL, terlebih dahulu mengaktifkan D2R *server*

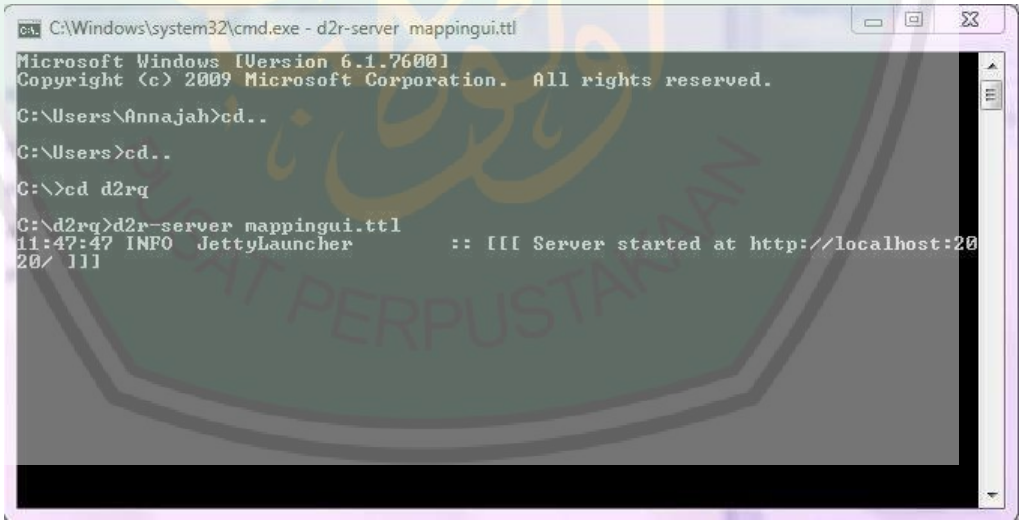
CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

dengan mengetikkan "d2r-*server* mappingui.ttl". mappingui.ttl adalah nama *file* hasil *generate* RDF yang dilakukan sebelumnya, seperti pada gambar 4.10.




Gambar 4.10 Proses *Start* Servis D2R *Server*

Setelah itu, akan muncul tulisan *server* started, yang menunjukkan bahwa D2R sudah berhasil diaktifkan. Seperti pada gambar 4.11.




Gambar 4.11 D2R *Server* Berhasil Dijalankan

Pada gambar 4.11 dijelaskan bahwa RDF *Map* berjalan pada alamat http://localhost:2020. Setelah itu D2R *server* dapat diakses pada web *browser*, seperti pada gambar 4.12.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG


Gambar 4. 12 Tampilan D2R *Server* pada *Browser*

Pada halaman D2R *server* terdiri empat bagian, pada bagian pertama menunjukkan tiga cara untuk mengakses data RDB menggunakan D2R *server*. Bagian kedua adalah tampilan HTML yang digunakan pada web *browser*. Bagian ketiga adalah tampilan RDF pada semantik web *browser*. Bagian keempat adalah alat untuk mengakses *query* SPARQL pada D2R *server*.


Gambar 4. 13 *Home Page* D2R *Server*

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pada D2R *server*, tabel-tabel *database* digenerate menjadi RDF *Map* berubah menjadi menu di D2R *server* dan berisi metadata dari database, seperti pada gambar 4.14.


Gambar 4.14 Metadata *Database*

Pada tampilan SPARQL *Endpoint*, digunakan untuk penulisan *query* SPARQL, seperti pada gambar 4.15.


Gambar 4. 15 *Endpoint* SPARQL

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pada penelitian ini, implementasi *query* SPARQL digunakan pada halaman pencarian. Pada gambar 4.16 menjelaskan implementasi *query* SPARQL pada halaman pencarian.

```
PREFIX db: <http://localhost:2020/resource/>
PREFIX map: <http://localhost:2020/resource/d2r-
mappings/uirepository.ttl#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>

SELECT DISTINCT ?Judul ?Link
WHERE {
  ?as vocab:form_nama?Judul.
  ?as vocab:form_url?Link

      filter (regex(?Judul,'$kkunci','i')||
            regex(?Link,'$kkunci','i'))
      }
```

Gambar 4.16 *Source code* SPARQL pada Halaman Pencarian

*Source code* pada gambar 4.16 tersimpan pada *file* proses.php. *file* tersebut berisi *query* SPARQL yang berfungsi untuk memanggil data dari RDF *mapping*. Dan untuk menampilkan hasil dari *query* tersebut dijelaskan pada gambar 4.17.

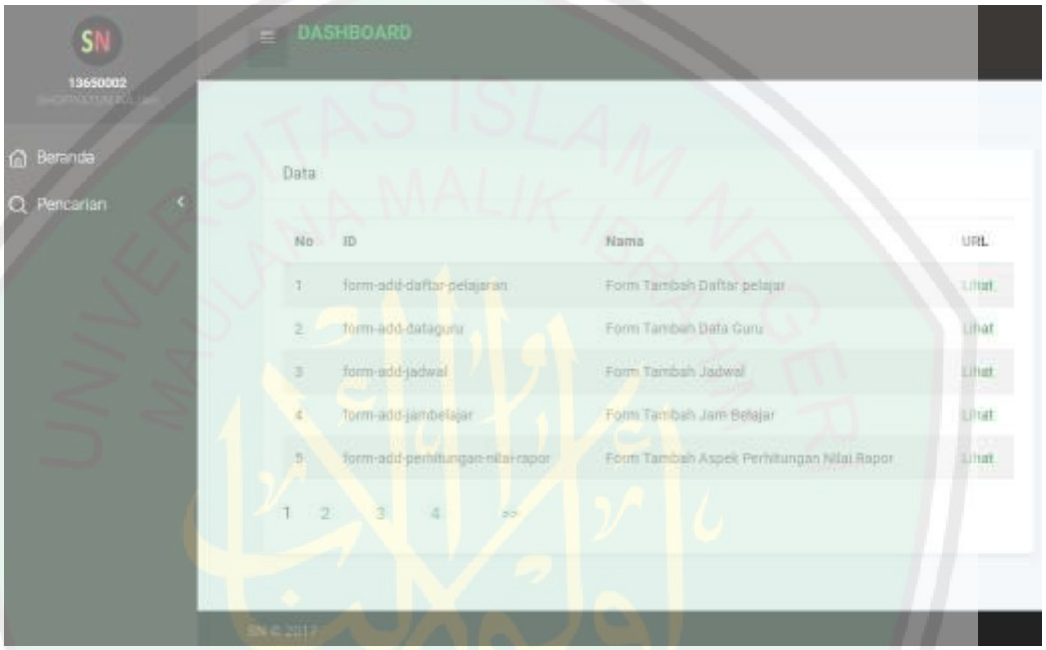
```
<?php
if ((($i<count($responseArray["results"]["bindings"]))== null ){
echo "Data yang anda cari berdasarkan kata kunci tidak ada,
silahkan coba lagi";
}else{
for($i=0;$i<count($responseArray["results"]["bindings"]);$i++)
{
?>
<table class="table">
<tr><td        class="col-md-10"><?php        echo        '<a
href="'.$responseArray["results"]["bindings"][$i]["Link"]["val
ue"].'"
class="">'.$responseArray["results"]["bindings"][$i]["Judul"][
"value"].'</a>'; ?></td>
```

Gambar 4.17 Menampilkan Hasil *Query* SPARQL

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

## 4.2 Langkah Uji Coba

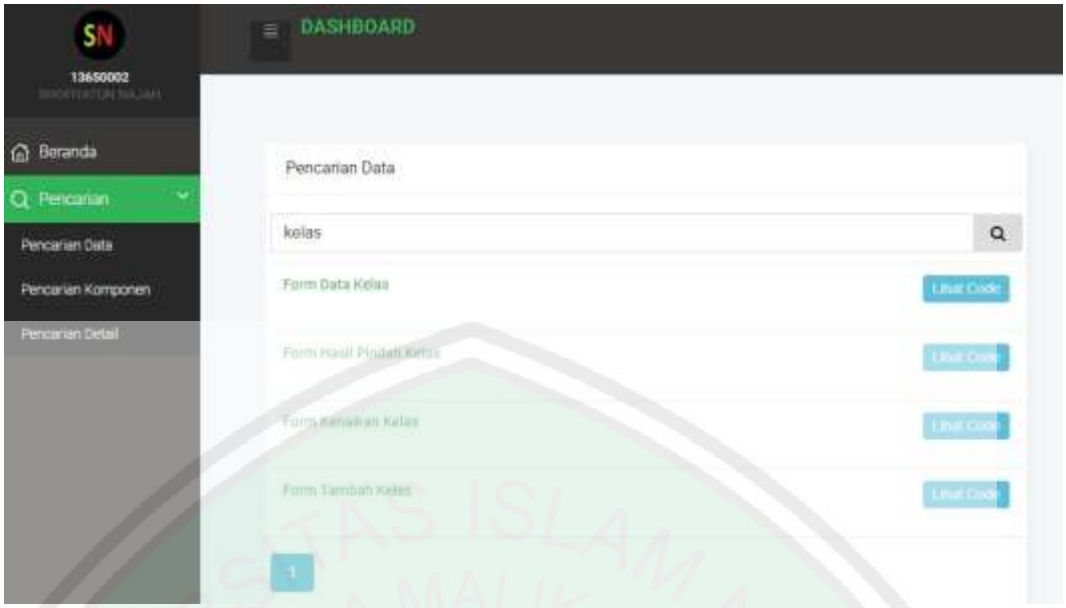
Pada tahap ini dilakukan pengujian program untuk mengetahui tingkat keefektifan sistem. Pengujian sistem pada tahap ini telah dilakukan uji coba tampilan sistem dan uji coba *query* pada *interface repository*. Tampilan awal program berisi halaman awal atau beranda yang berisi *form* yang tersimpan pada *interface repository*. Tampilan awal program seperti yang diilustrasikan pada gambar 4.18.




Gambar 4. 18 Tampilan Awal Program

Skenario awal yaitu proses pencarian data *interface repository* pada menu pencarian data. Kata kunci yang digunakan berupa nama *form*. Pada tahap ini dilakukan proses pencarian data yang proses pencariannya hanya berdasarkan nama *form* yang tersimpan pada *interface repository* dan hasilnya berupa nama *form* yang mengandung unsur kata yang dijadikan kata kunci oleh *user*. Proses ini mengambil contoh *query* yang dimasukkan yaitu "kelas" pada *repository* data seperti yang digambarkan pada gambar 4.19 dan pada proses pencarian detail yang ditunjukkan pada gambar 4.20.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

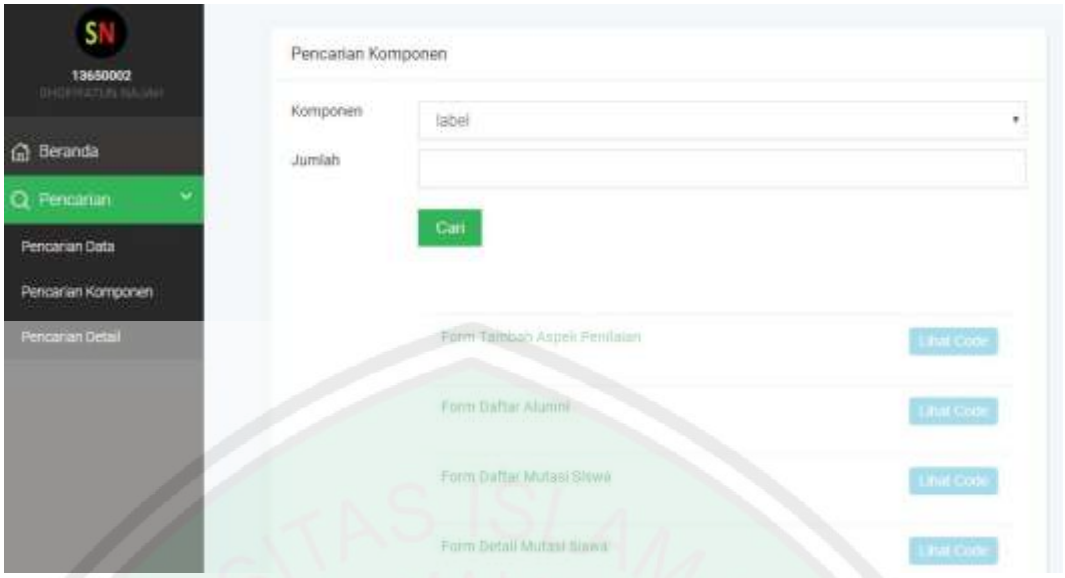



Gambar 4. 19 Uji Coba *Query* "kelas" pada Pencarian Data


Gambar 4. 20 Uji Coba *Query* "kelas" pada Pencarian Detail

Pada tahap selanjutnya yaitu proses pencarian komponen, yaitu proses yang menggunakan *query* berupa nama komponen dan jumlah yang diinginkan. Pada halaman ini, bertujuan untuk memudahkan pengguna untuk memilih *form* yang sesuai dengan komponen dan jumlah yang dibutuhkan oleh pengguna, seperti pada gambar 4.21.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG




Gambar 4. 21 Uji Coba Pencarian Komponen

## 4.3 Hasil Uji Coba

Berdasarkan skenario uji coba pada aplikasi *user interface discovery* ini dilakukan untuk mengetahui hasil *output* dari *user interface discovery* yang dibangun menggunakan semantik web. Uji coba sistem ini meliputi percobaan pencarian *query* data dalam *interface repository*. Nilai *precision* dan *recall* akan menunjukkan nilai relevansi dokumen yang ditemukan melalui proses *discovery*. Pengujian dilakukan dengan menggunakan 30 data *form* yang ada di bidang akademik sistem informasi pondok pesantren. Seperti yang ada pada tabel 4.1.

Tabel 4. 1 Data *Form UI Discovery*

| No. | Nama |
|---|---|
| 1 | *Form* Tambah Aspek Penilaian |
| 2 | *Form* Tambah Daftar pelajar |
| 3 | *Form* Tambah Data Guru |
| 4 | *Form* Tambah Jadwal |
| 5 | *Form* Tambah Jam Belajar |
| 6 | *Form* Tambah Aspek Perhitungan Nilai Rapor |
| 7 | *Form* Buat Pesan |
| 8 | *Form* Angkatan |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

| | |
|---|---|
| 9 | *Form* Tambah Angkatan |
| 10 | *Form* Aspek Penilaian |
| 11 | *Form* Cari Alumni |
| 12 | *Form* Daftar Alumni |
| 13 | *Form* Daftar Mutasi Siswa |
| 14 | *Form* Daftar Pelajaran |
| 15 | *Form* Data Kelas |
| 16 | *Form* Detail Konfgurasi Pendataan PSB |
| 17 | *Form* Detail Mutasi Siswa |
| 18 | *Form* Detail Pencarian Siswa |
| 19 | *Form* Detail Penempatan Calon Siswa |
| 20 | *Form* Detail Presensi Harian |
| 21 | *Form* Detail Laporan Presensi Harian |
| 22 | *Form* Hasil Pindah Kelas |
| 23 | *Form* Kelulusan |
| 24 | *Form* Kenaikan Kelas |
| 25 | *Form* Kirim Dokumen |
| 26 | *Form* Mutasi Siswa |
| 27 | *Form* Pegawai |
| 28 | *Form* Pendataan Alumni |
| 29 | *Form* Pesan |
| 30 | *Form* Tambah Kelas |

Hasil pengujian *precision*, *recall*, dan *accuracy* pada sistem ini menggunakan kata kunci "kelas" dengan melakukan *query* pencarian pada pada tabel data *interface repository*. Hasil pengujian didapat tabel seperti pada tabel 4.2.

Tabel 4. 2 Pencarian Data dengan *Query* "Kelas"

| No. | *Form* | *Match* |
|---|---|---|
| 1 | *Form* Data Kelas | ✓ |
| 2 | *Form* Hasil Pindah Kelas | ✓ |
| 3 | *Form* Kenaikan Kelas | ✓ |
| 4 | *Form* Tambah Kelas | ✓ |

Tabel 4.2 menunjukkan bahwa hasil yang keluar dengan kata "kelas" memiliki tingkat relevan yang tinggi. Data diuji dengan perhitungan nilai *recall*,

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

*precision*, dan *accuracy* yang menunjukkan nilai 1 atau 100% seperti yang diilustrasikan pada tabel 4.3.

Tabel 4. 3 Perhitungan Akurasi Query Pencarian Data

| | Ret | TP | TN | FP | FN | *Precision* | *Recall* | *Accuracy* |
|---|---|---|---|---|---|---|---|---|
| Tabel Data *Interface* | 4 | 4 | 26 | 0 | 0 | $\frac{4}{4+0}=1$ | $\frac{4}{4+0}=1$ | $\frac{4+26}{4+26+0+0}=1$ |

Selain itu hasil percobaan uji akurasi pada pencarian detail berbeda dengan hasil percobaan pada pencarian data. Apabila pada pencarian detail memasukkan kata kunci "kelas", maka hasil pencarian tersebut seperti yang ditunjukkan pada tabel 4.4.

Tabel 4. 4 Pencarian Detail *Query* "Kelas"

| No. | *Form* | *Match* |
|---|---|---|
| 1 | *Form* Tambah Jadwal | ✓ |
| 2 | *Form* Data Kelas | ✓ |
| 3 | *Form* Detail Mutasi Siswa | ✓ |
| 4 | *Form* Detail Penempatan Calon Siswa | ✓ |
| 5 | *Form* Hasil Pindah Kelas | ✓ |
| 6 | *Form* Kelulusan | X |
| 7 | *Form* Kenaikan Kelas | ✓ |
| 8 | *Form* Kirim Dokumen | ✓ |
| 9 | *Form* Mutasi Siswa | X |
| 10 | *Form* Tambah Kelas | ✓ |

Tabel 4.4 menunjukkan hasil bahwa hasil pencarian *query* tidak relevan. Perhitungan *recall*, *precision*, dan *accuracy* pada tabel 4.4 ditunjukkan seperti pada tabel 4.5.

Tabel 4.5 Perhitungan Akurasi *Query* Pencarian Detail

| | Ret | TP | TN | FP | FN | *Precision* | *Recall* | *Accuracy* |
|---|---|---|---|---|---|---|---|---|
| Tabel Data *Interface* | 10 | 8 | 20 | 2 | 0 | $\frac{8}{8+2}=0{,}8$ | $\frac{10}{10+0}=1$ | $\frac{8+20}{8+20+2+0}=0{,}93$ |

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Perhitungan akurasi dengan *query* "kelas" pada pencarian detail telah diilustrasikan pada tabel 4.5 yang menunjukkan bahwa pencarian tersebut menghasilkan rata-rata *precision* adalah 0,8, nilai rata-rata *recall* adalah 1, dan nilai rata-rata akurasi yaitu 0,93 Jika ditunjukkan dalam nilai presentasi didapat nilai rata-rata *precision* sebesar 80%, nilai rata-rata *recall* sebesar 100%, dan nilai rata-rata akurasi yakni 93%.

Tabel 4. 6 Hasil Uji Coba *Query* Pencarian Data

| No. | *Keyword* | Ret | TP | TN | FP | FN | P | R | A |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Cari | 2 | 2 | 28 | 0 | 0 | 1 | 1 | 1 |
| 2 | Alumni | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 3 | Nilai | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 4 | Add | 8 | 8 | 22 | 0 | 0 | 1 | 1 | 1 |
| 5 | Jadwal | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 6 | Guru | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 7 | Aspek | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 8 | Hitung | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 9 | Rapor | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 10 | Pesan | 2 | 2 | 28 | 0 | 0 | 1 | 1 | 1 |
| 11 | Buat | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 12 | Dokumen | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 13 | Kirim | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 14 | Kelas | 4 | 4 | 26 | 0 | 0 | 1 | 1 | 1 |
| 15 | Ajar | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 16 | Lulus | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 17 | Akademik | 0 | 0 | 30 | 0 | 0 | 0 | 0 | 1 |
| 18 | Detail | 6 | 6 | 24 | 0 | 0 | 1 | 1 | 1 |
| 19 | Hari | 2 | 2 | 28 | 0 | 0 | 1 | 1 | 1 |
| 20 | Naik | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| Rata-rata | | | | | | | 0,95 | 0,95 | 1 |

Ret = Retrieval, TP = *True Positive*, FP = *False Positive*, FN = *False Negative*, TN = *True Negative*, P = *Precision*, R = *Recall*, A = *Accuracy*

Pada tabel 4.6 menunjukkan hasil uji coba perhitungan nilai *recall*, *precision*, dan *accuracy* pada 20 *query* yang telah diuji pada tahap pencarian

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

data. Sedangkan pada tabel 4.7 menunjukkan hasil perhitungan *precision* dan *recall* dari 20 *query* yang sama dan telah dilakukan uji coba pada pencarian detail. Berdasarkan hasil tabel perhitungan tersebut diketahui bahwa nilai rata-rata *precison* adalah 0,9, nilai rata-rata *recall* adalah 0,95, dan nilai rata-rata *accuracy* adalah 1. Jika digambarkan dengan nilai presentase maka diperoleh nilai rata *precison* adalah 90%, nilai rata-rata *recall* adalah 95%, dan nilai rata-rata *accuracy* adalah 100%.

Tabel 4.7 Hasil Uji Coba *Query* Pencarian Detail

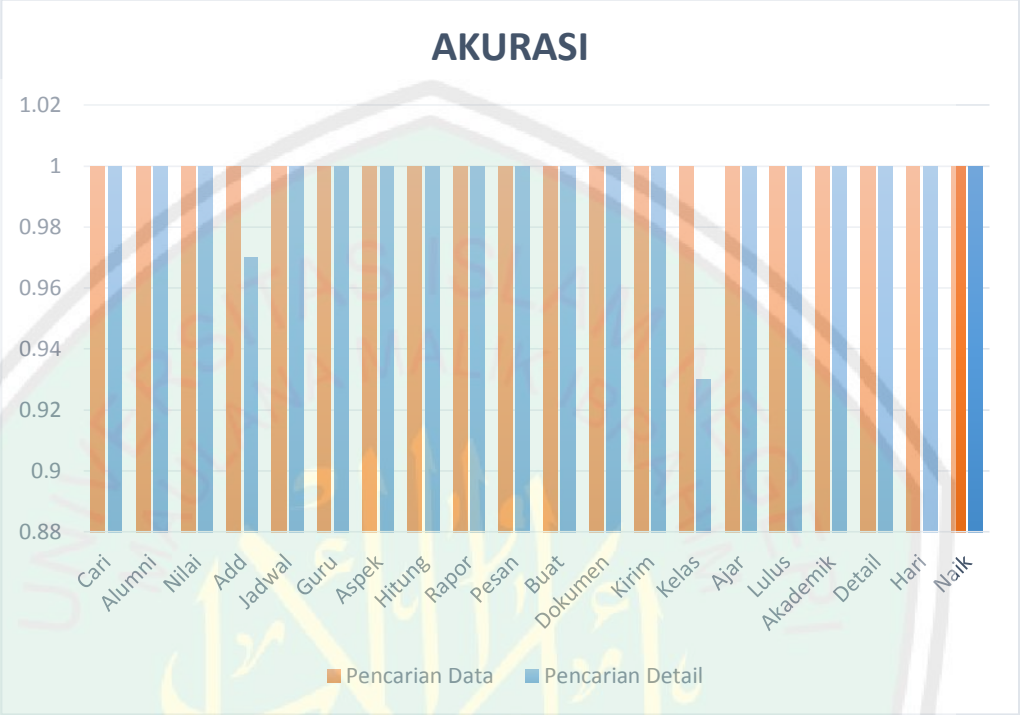
| No. | Keyword | Ret | TP | TN | FP | FN | P | R | A |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Cari | 5 | 5 | 25 | 0 | 0 | 1 | 1 | 1 |
| 2 | Alumni | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 3 | Nilai | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 4 | Add | 10 | 9 | 20 | 1 | 0 | 0,9 | 1 | 0,97 |
| 5 | Jadwal | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 6 | Guru | 6 | 6 | 24 | 0 | 0 | 1 | 1 | 1 |
| 7 | Aspek | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 8 | Hitung | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 9 | Rapor | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 10 | Pesan | 2 | 2 | 28 | 0 | 0 | 1 | 1 | 1 |
| 11 | Buat | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 12 | Dokumen | 1 | 1 | 29 | 0 | 0 | 1 | 1 | 1 |
| 13 | Kirim | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 14 | Kelas | 10 | 8 | 20 | 2 | 0 | 0,8 | 1 | 0,93 |
| 15 | Ajar | 17 | 17 | 13 | 0 | 0 | 1 | 1 | 1 |
| 16 | Lulus | 5 | 5 | 25 | 0 | 0 | 1 | 1 | 1 |
| 17 | Akademik | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 18 | Detail | 6 | 6 | 24 | 0 | 0 | 1 | 1 | 1 |
| 19 | Hari | 3 | 3 | 27 | 0 | 0 | 1 | 1 | 1 |
| 20 | Naik | 5 | 5 | 25 | 0 | 0 | 1 | 1 | 1 |
| Rata-rata | | | | | | | 0,9 | 0,95 | 1 |

Ret = Retrieval, TP = *True Positive*, FP = *False Positive*, FN = *False Negative*, TN = *True Negative*, P = *Precision*, R = *Recall*, A = *Accuracy*

Pada hasil uji coba yang telah dilakukan berdasarkan skenario sebelumnya dapat diketahui bahwa terdapat perbedaan antara pencarian kata *string-matching*

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

yang diinputkan oleh *user* pada UI *discovery*. Dengan menggunakan 20 *keyword* uji coba pada pencarian data dan pencarian detail pada *interface repository* telah menunjukkan perbandingan nilai akurasi seperti yang diilustrasikan pada gambar 4.22.




Gambar 4. 22 Grafik Perbandingan Akurasi *Query*

Perbedaan nilai akurasi disebabkan karena terdapat beberapa analisa pada pencarian detail ada beberapa dokumen tidak relevan yang ditemukan dan terjaring *query*, sehingga pada pencarian data menunjukkan hasil yang lebih relevan dan menunjukkan kemampuan sistem lebih baik.

## 4.4 Integrasi UI *Discovery* dengan Islam

Islam merupakan agama yang mewajibkan umatnya untuk saling tolong-menolong. Menolong orang lain hakikatnya yaitu menolong untuk diri sendiri, yaitu dengan mengundang datangnya bantuan atau pertolongan dari Allah SWT. Rasulullah Saw menegaskan, orang yang membantu orang lain yang sedang dalam kesulitan, maka akan mendapatkan bantuan Allah di hari akhirat kelak, seperti yang dijelaskan pada kandungan hadits shahih berikut ini:

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللهُ عَنْهُ۞ عَنِ النَّبِيِّ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ قَالَ : مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا نَفَّسَ اللهُ عَنْهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ الْقِيَامَةِ۞ وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ يَسَّرَ اللهُ عَلَيْهِ فِي الدُّنْيَا وَالآخِرَةِ۞ وَمَنْ سَتَرَ مُسْلِماً سَتَرَهُ اللهُ فِي الدُّنْيَا وَالآخِرَةِ وَاللهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيْهِ .

Artinya:
*Abu Hurairah ra. berkata, Rasulullah saw. bersabda, "Barangsiapa yang berusaha melapangkan suatu kesusahan kepada seorang mukmin dari kesusahan-kesusahan dunia, maka Allah akan melapangkannya dari suatu kesusahan di hari kiamat dan barang siapa yang berusaha memberi kemudahan bagi orang yang kesusahan, maka Allah akan memberi kemudahan baginya di dunia dan akhirat. Barang siapa yang berusaha menutupi kejelekan orang Islam, Allah akan menutupi kejelekannya di dunia dan akhirat. Allah selalu membantu hamba-Nya selama hamba itu menolong sesame saudaranya." (HR. Muslim)*

Hadits diatas menjelaskan tentang berbagai ilmu, prinsip-prinsip agama, dan akhlak. Hadits tersebut berisi keutamaan memenuhi kebutuhan-kebutuhan orang mukmin, memberi manfaat kepada sesama umat dengan fasilitas ilmu, harta, bimbingan atau petunjuk yang baik, atau nasihat dan sebagainya.

Kalimat "Sesungguhnya Allah akan selalu menolong seorang hamba selama dia gemar menolong saudaranya" menjelaskan bahwa seseorang apabila memiliki keinginan untuk menolong saudaranya, maka seharusnya dikerjakan, baik dalam bentuk kata-kata maupun pembelaan atas sebuah kebenaran dengan didasari rasa beriman kepada Allah ketika melaksanakannya.

Aplikasi pada penelitian ini salah satunya bertujuan untuk menolong para pengguna web servis dengan memudahkan pengguna dalam pencarian tampilan sebuah *form* pada web servis sehingga bisa memanfaatkan waktu sebaik mungkin.

Islam menganjurkan agar manusia bisa memanfaatkan waktu dan kesempatan yang dimiliki sehingga ia tidak termasuk dalam golongan orang yang merugi. Hal tersebut tercantum dalam Al-Qur'an Surat Al-'Ashr .

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

وَٱلۡعَصۡرِ ۝١ إِنَّ ٱلۡإِنسَٰنَ لَفِي خُسۡرٍ ۝٢ إِلَّا ٱلَّذِينَ ءَامَنُواْ وَعَمِلُواْ ٱلصَّٰلِحَٰتِ وَتَوَاصَوۡاْ بِٱلۡحَقِّ وَتَوَاصَوۡاْ بِٱلصَّبۡرِ ۝٣

Artinya:
"*Demi masa (1). Sesungguhnya manusia itu benar-benar dalam kerugian (2). Kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran (3).*" *(QS. Al-'Ashr:1-3)*

Pada ayat pertama menjelaskan bahwa Allah memulai surat ini dengan sumpah. Ketika manusia bersumpah atas nama Allah SWT, maka Allah SWT bersumpah atas nama makhlukNya. Hal tersebut disebabkan tidak ada selain Dia kecuali makhlukNya. Dan sumpah Allah SWT demi masa ini menunjukkan bahwa waktu itu sangat penting sehingga Allah SWT bersumpah dengannya. Sebagaimana sumpah manusia untuk meyakinkan seseorang akan kebenaran, maka Allah SWT pun meyakinkan manusia akan pentingnya sebuah waktu bagi manusia.

Pada ayat kedua, "Sesungguhnya manusia benar-benar berada dalam kerugian" menunjukkan bahwa manusia banyak yang merugi. Namun manusia sedikit yang menyadari akan hal tersebut, sehingga Allah SWT bersumpah untuk meyakinkan manusia bahwa mereka sungguh berada dalam kerugian karena tidak dapat menggunakan waktu di dunia sebaik-baiknya di dunia.

Pada ayat ketiga, menjelaskan bahwa ada tiga syarat agar manusia tidak dikategorikan sebagai orang merugi. Yaitu beriman, mengerjakan amal sholeh dan saling menasehati dalam kebenaran dan kesabaran. Iman adalah syarat pertama manusia sebelum syarat yang lain.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# BAB 5

# PENUTUP

## 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Penggunaan semantik web pada aplikasi *user interface discovery* dilakukan dengan menggunakan RDF *Map* sebagai pendekatan ontologi.
2. Aplikasi *user interface discovery* pada *interface* repository berdasarkan hasil percobaan RDF *Map* yang telah dijalankan dengan D2R *server* menggunakan *query* SPARQL dapat memudahkan dalam mengakses data.
3. Berdasarkan perhitungan uji coba yang telah dilakukan dengan dua model pencarian yakni pencarian data dan pencarian detail memperoleh *recall* 95% dan akurasi 100%, namun presisi pada pencarian data didapatkan nilai 95% dan pada pencarian detail diperoleh nilai 90%. Perbedaan tersebut diperoleh karena pada pencarian detail ada beberapa dokumen tidak relevan yang ditemukan dan terjaring *query*, sehingga pada pencarian data menunjukkan hasil yang lebih relevan dan menunjukkan kemampuan sistem lebih baik.

## 5.2 Saran

Dari penelitian yang telah dilakukan, ada beberapa hal yang perlu dikembangkan dari penelitian ini, diantaranya:

1. Menambahkan tahap pengerjaan sistemnya menggunakan metode *indexing* terhadap hasil pencarian, salah satunya dengan teknik pembobotan agar mendapatkan hasil pencarian yang lebih relevan.



CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

2. Menggunakan standarisasi OWL saat pembuatan deskripsi dan persamaan kata sehingga *discovery* menjadi lebih mudah.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

# DAFTAR PUSTAKA


[1] V. Choudhary, "Software as a Service : Implications for Investment in Software Development The Paul Merage School of Business," *Proc. 40th Hawaii Int. Conf. Syst. Sci.*, vol. 40, no. 7, pp. 1–10, 2007.

[2] Q. Shao, "Towards effective and intelligent multi-tenancy SaaS," no. May, 2011.

[3] A. L. Bento and R. Bento, "Journal of Information Technology Management A Publication of the Association of Management Cloud Computing : A New Phase In Information Technology Management," vol. XXII, no. 1, pp. 39–46, 2011.

[4] M. Godse and T. Infotech, "An Approach for Selecting Software-as-a-Service ( SaaS ) Product," pp. 155–158, 2009.

[5] H. S. Pinto and J. P. Martins, "Ontologies: How can They be Built?," *Knowl. Inf. Syst.*, vol. 6, no. 4, pp. 441–464, 2004.

[6] J. O'Sullivan, D. Edmond, and A. Ter Hofstede, "What's in a service? Towards accurate description of non-functional service properties," *Distrib. Parallel Databases*, vol. 12, no. 2–3, pp. 117–133, 2002.

[7] B. Matthews, "Semantic Web Technologies," no. January 2005, 2014.

[8] M. M. Taye, "Web-Based Ontology Languages and its Based Description Logics," *Res. Bull. Jordan ACM*, vol. 2, no. 2, pp. 1–9, 2012.

[9] A. J. Ilmiah, "Studi Tentang Pemodelan Ontologi Web Semantik Dan Prospek Penerapan Pada Bibliografi Arti ," no. March, 2017.

[10] P. Widodo, J. A. Putra, S. Afiadi, A. Z. Arifin, and D. Herumurti, "Klasifikasi Kategori Dokumen Berita Berbahasa Indonesia Dengan Metode Kategorisasi Multi- Label Berbasis Domain Specific Ontology," vol. II, no. 2, 2016.

[11] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *Int. J. Hum. Comput. Stud.*, vol. 43, no. 5–6, pp. 907–928, 1995.

[12] T. Nuryati, "Teknologi Untuk Peningkatan Web Service Search Melalui Ontology- Based Semantic Interoperability ( Studi Kasus Aplikasi SIATer Employee Universitas Bandar Lampung ) Tri Nuryati," pp. 18–28, 2012.

[13] A. Kunaefi and R. Sarno, "Ontology Mapping for Erp Business Process," pp. 1–6, 2013.

[14] C. A. Djeni, R. Sarno, D. Sunaryono, and A. W. M. System, "System pada Kasus Enterprise Resource Planning ( ERP )," vol. 2, no. 1, 2013.

[15] D. Hutchison and J. C. Mitchell, *The Semantic Web – ISWC 2013*, no. October. 1973.

[16] P. T. Informatika, F. Teknik, U. M. Malang, and J. R. Tlogomas, "Hasil




CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG

Pencarian Dari Mesin Pencari Yang Relevan Dengan Query Expansion Menggunakan Ontologi," vol. 5, no. 3, 2016.

[17] Y. Chen, X. Zhao, and S. Zhang, "Publishing rdf from relational database based on d2R improvement," *WSEAS Trans. Inf. Sci. Appl.*, vol. 10, no. 8, pp. 241–248, 2013.

[18] S. Rikacovs and J. Barzdins, "Export of Relational Databases to RDF Databases : A Case Study," *Proc. 9th Int. Conf. Perspect. Bus. Informatics Res. (BIR 2010)*, pp. 203–211, 2010.

[19] R. E. Barber and H. C. Lucas, "System response time operator productivity and job satisfaction," *Commun. ACM*, vol. 26, no. 11, pp. 972–986, 1983.

[20] M. Green, "The University of Alberta user interface management system," *SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 205–213, 1985.

[21] W. Tsai, Q. Huang, J. Elston, and Y. Chen, "Service-Oriented User Interface Modeling and Composition 1," pp. 21–28, 2008.

[22] D. M. Informatika, F. I. Terapan, U. Telkom, J. Telekomunikasi, and B. Batu, "Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining," no. November, pp. 2–3, 2015.

[23] D. Kurniawan, "Evaluasi Sistem Temu Kembali Informasi Model Ruang Vektor," vol. 16, no. 3, pp. 155–162, 2010.

CENTRAL LIBRARY OF MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY OF MALANG