
Repositori Publikasi Berbasis OAI-PMH 2.0 dengan Google APP Engine

Hendra¹ Jimmy²

STMIK IBBI Medan

Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548

Email: hendra.soewarno@gmail.com¹⁾, jim8470@yahoo.com²⁾

Abstrak

Publikasi karya ilmiah secara *online* oleh perguruan tinggi bertujuan untuk meningkatkan penyebaran dan akses oleh masyarakat maupun industri kepada hasil penelitian. OAI-PMH 2.0 adalah suatu protokol yang memungkinkan metadata publikasi yang diekspos oleh suatu *data provider* dapat dipanen secara online oleh service provider tanpa intervensi manusia. Suatu portal publikasi yang dilengkapi dengan fasilitas ekspos metadata akan meningkatkan penyebaran dan akses secara luas melalui layanan yang disediakan oleh service provider. Penelitian ini bertujuan mengembangkan suatu Repositori Publikasi yang dilengkapi dengan fasilitas ekspos metadata berbasis OAI-PMH 2.0 yang berjalan diatas Google App Engine. Google App Engine merupakan layanan PaaS yang disediakan oleh Google. Pengembangan aplikasi dilakukan dengan pendekatan SDLC, dan pada tahapan analisa dan perancangan menggunakan OOAD. Aplikasi hasil pengembangan diimplementasi untuk publikasi hasil penelitian dosen-dosen STMIK IBBI yang diberi nama Portal Garuda STMIK IBBI. Berdasarkan hasil pengujian dengan OAI-PMH Validator, BASE OAI-PMH Validator, dan berhasil didaftarkan Portal Garuda STMIK IBBI di OpenArchive.org, OpenDOAR, dan ROAR, serta hasil rating yang mencapai 95% oleh WebArchive, maka diyakini aplikasi telah sesuai dengan standard OAI-PMH 2.0 dan W3C sehingga nantinya membantu institusi perguruan tinggi untuk memenuhi kewajiban publikasi karya ilmiah yang dapat ditelusuri secara online sebagaimana surat Dikti nomor 2050/E.T/2011.

Kata kunci: Data Provider, OAI-PMH, Google App Engine

Abstract

Online publication of scientific papers by the college aims to improve the dissemination and access by the public and the industry to the research result. OAI-PMH standard 2.0 is a protocol that allows the publication metadata exposed by a data provider can be harvested online by a service provider without any human intervention. A publication portal that is equipped with metadata will increase the exposure and wider spread through the service provided by services provider. This study aims at developing a Repository Publications completed with meta data exposure facility base on OAI-PMH 2.0 running on Google App Engine. Google App Engine is a PaaS service provided by Google. Application development is done using SDLC approach, and using OOAD at the analysis and design phases. Application development results implemented for publication of research results lecturers STMIK IBBI named Portal Garuda STMIK IBBI. Based on the results of testing with OAI-PMH Validator, Validator BASE OAI-PMH, and successful registration of portal Garuda STMIK IBBI in OpenArchive.org, OpenDOAR, and the ROAR, as well as the result rating reaching 95% by Webarchive, it is believed the Application complies with OAI-PMH 2.0 and the W3C standard that will help higher education institutions to meet the obligations of the publication of scientific papers that can be searched online as well as letter of Dikti number 2050/ET/2011.

Keywords: Data Provider, OAI-PMH, Google App Engine

1. Pendahuluan

Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) merupakan mekanisme *low-barrier* untuk interoperabilitas repositori [1]. Suatu Repositori yang dilengkapi dengan fasilitas ekspos data menggunakan OAI-PMH 2.0 memungkinkan metadata dipanen secara online oleh suatu *service provider*.

Dikti melalui surat nomor 2050/E.T/2011 mewajibkan publikasi karya ilmiah dan jurnal yang dapat ditelusuri secara *online* oleh perguruan tinggi. Kebijakan ini membutuhkan investasi fasilitas dan sumber daya manusia untuk pengoperasian dan pemeliharaan fasilitas tersebut. Kehadiran dari Cloud Computing dapat menjadi alternatif bagi perguruan tinggi untuk melakukan publikasi online tanpa investasi awal

yang besar. Berdasarkan penelitian bahwa pemanfaatan Cloud Computing pada institusi pendidikan dapat mengurangi kompleksitas dan biaya serta meningkatkan efisiensi [2,3].

Google APP Engine (GAE) merupakan layanan Paas yang memberikan kesempatan kepada programmer untuk mengembangkan aplikasi menggunakan Google App Engine SDK, dan menjalankan aplikasi tersebut diatas infrastruktur Google. GAE memberikan kuota cuma-cuma untuk penyimpanan datastore 1 GB, Blostore 5 GB dan CPU serta bandwidth yang cukup untuk suatu aplikasi efisien melayani sekitar 5 juta tampilan halaman setiap bulannya, sehingga menjadi sesuatu menarik untuk mengawali pemanfaatan Cloud Computing.

Google juga menyediakan layanan Google Apps for Education yang memungkinkan institusi pendidikan untuk mempublikasikan layanan aplikasi berbasis GAE dengan menggunakan domain maupun sub-domain institusi tanpa dipungut bayaran.

Tujuan dari penelitian ini adalah mengembangkan suatu Aplikasi Repositori Publikasi yang dilengkapi dengan kemampuan expose metadata melalui OAI-PMH 2.0. Aplikasi ditargetkan untuk berjalan pada platform Google App Engine.

Adapun manfaat dari penelitian ini adalah tersedianya suatu Portal Publikasi yang dapat menjadi solusi bagi perguruan tinggi untuk memenuhi kewajiban publikasi hasil penelitian yang dapat ditelusuri secara online.

2. Metode Penelitian

Penelitian ini dilakukan dengan mempelajari berbagai literatur terkait dengan penyebaran dan expose metadata hasil penelitian, mempelajari standard OAI-PMH 2.0, mempelajari meta format Dublin-Core (DC), mempelajari konsep *cloud computing*, mempelajari pengembangan aplikasi dengan Google App Engine SDK. Pengembangan aplikasi menggunakan SDLC, dan menggunakan OOAD pada tahapan analisa dan perancangan.

OAI-PMH 2.0

Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) merupakan mekanisme *low-barrier* untuk interoperabilitas repositori. Suatu repositori adalah suatu jaringan server yang dapat diakses dengan enam permintaan OAI-PMH yaitu: Identify, Listsets, ListMetaFormats, ListIdentifier, ListRecords, dan GetRecords. Suatu repositori adalah sebagai data provider (penyedia data) untuk mengekspose metadata kepada harvester (pemanen). Suatu pemanen adalah aplikasi klien yang mengajukan permintaan OAI-PMH yang dioperasikan oleh suatu *service provider* (penyedia layanan) dalam arti untuk mengumpulkan metadata dari repositori.

Konsep utama dari spesifikasi OAI-PMH adalah item, record dan MetadataFormat, sebuah item adalah representasi dari suatu sumber daya digital atau non-digital yang secara unik diidentifikasi dengan suatu URI. Sebuah record adalah diidentifikasi oleh kombinasi dari identifier unik, sebuah metadataPrefix yang mengidentifikasi format dari record seperti "oai_dc" untuk Dublin Core dan sebuah timestamp untuk record. Masing-masing item secara optional dapat dikelompokkan berdasarkan parameter setSpec.

OAI-PMH diimplementasikan diatas HTTP dengan respon terhadap permintaan HTTP baik GET maupun POST yang memiliki parameter verbs yang menentukan jenis informasi yang diminta seperti Identify adalah untuk mengambil metadata administratif terkait dengan repositori, ListSets, ListMetadataFormats, ListIdentifier, ListRecords, dan GetRecord. Penentuan kriteria dapat berdasarkan timestamp, maupun set yang dinyatakan pada perintah ListRecords dan ListIdentifier. Kriteria berdasarkan timestamp adalah untuk memanen record berdasarkan stempel waktu ketika suatu item dibuat, dihapus ataupun dimodifikasi. Suatu timestamp dikodefikasi menggunakan ISO8601 dan diekspresikan dalam UTC. Setiap header yang dikembalikan oleh permintaan GetRecord, ListRecord ataupun ListIdentifier mengandung suatu timestamp terkait dengan item tersebut. Respon dikembalikan dalam bentuk dokumen XML dengan content type: text/xml dan dikodefikasi dengan menggunakan representasi UTF-8 dari Unicode, jika repositori mendukung set maka informasi set harus disertakan pada header dari item terkait dengan respon terhadap permintaan ListIdentifiers, ListRecords, dan GetRecord. Respon dapat mengembalikan respon error terkait dengan tidak tersedianya argumen maupun argumen yang diberikan yang tidak sah seperti badVerb, badArgument, noRecordsMatch, cannotDisseminateFormat, badResumptionToken

Sejumlah permintaan OAI-PMH mengembalikan suatu daftar dari entitas diskrit seperti ListRecords mengembalikan daftar dari record, ListIdentifier mengembalikan daftar dari header, dan ListSets

mengembalikan suatu daftar dari set, dalam kasus tertentu ukuran list bisa besar dan lebih praktis adalah mempartisi mereka kedalam suatu rangkaian request dan respon dengan suatu resumptionToken[4].

Meta Format Dublin Core

Pada dasarnya penyedia data OAI-PMH memiliki kebebasan untuk mengekspos data dalam berbagai format untuk menjamin interoperabilitas tingkat rendah, semua penyedia data minimal mendukung paling sedikit format Dublin Core walaupun dimungkinkan ekspos metadata dalam format lainnya. Berdasarkan observasi oleh Bernhard Haslhofer et. al. (2009) terhadap 915 repositori selain mengimplementasikan format mandatori Dublin Core, berturut-turut lima format yang sering digunakan adalah sebagai berikut FRC1807 (12%), MARC (11.8%), MARC-21 (10.3%), MIDS (7.5%), dan METS (5.7%).

Dublin Core metadata memiliki lima belas elemen yang sifatnya optional dan dapat diulang, elemen ini adalah Contributor, Coverage, Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Source, Subject, Title dan Type[5]. Berdasarkan best practices yang dipublikasikan oleh IMLS DCC mengidentifikasi delapan dari elemen yang berpengaruh terhadap kelengkapan dari suatu record metadata, dan paling membantu untuk pencarian dan ditemukan adalah Title, Creator, Description, Date, Format, Identifier, dan Rights CDP Metadata Working Group (2006)[6]

Cloud Computing

Cloud computing merupakan istilah untuk pemasaran. Berdasarkan definisi dari NIST, cloud computing merupakan suatu model yang memungkinkan akses jaringan dimana-mana, nyaman, dan bersifat sesuai permintaan ke suatu kumpulan berbagi sumber daya komputasi yang dapat dikonfigurasi (contohnya: jaringan, server, penyimpanan, aplikasi dan jasa), dimana dapat dapat secara cepat ditetapkan dan dilepaskan.[7]

Katz et al. Mengidentifikasi 10 fitur penting dari komputasi awan pada institusi pendidikan tinggi terkait dengan on-demand pada SaaS, PaaS, dan IaaS adalah meningkatkan akses kepada kelangkaan ahli IT dan bakat, meningkatkan skala layanan dan sumber daya IT, mempromosikan standarisasi IT kedepan, meningkatkan waktu ke pasar melalui pengurangan hambatan pasokan IT, jalur dan pemecahan terhadap kekhususan konsumerasi layanan IT diperusahaan, memfasilitasi transparansi kesesuaian biaya IT terkait dengan kebutuhan dan pendanaan, meningkatkan interoperabilitas antar teknologi yang terpisah dan antar lembaga, dukungan terhadap suatu model lingkungan 24x7x365, mengaktifkan siklus pengadaan dan penyimpanan yang didukung dengan energi terbarukan, dan menurunkan modal serta biaya total IT pada pendidikan tinggi.[8]

B. Sosinsky menuliskan pemanfaatan komputasi awan dapat memberikan lima keuntungan yaitu: biaya yang lebih rendah karena beroperasi dengan utilitas yang lebih baik, *quality of service* (QoS) sesuai dengan kontrak, kehandalan jaringan dengan tersedianya *load balancing* dan *failover*, *out-source* manajemen IT dimana infrastruktur ditangani penyedia jasa, pemeliharaan dan upgrade yang lebih sederhana karena sistem tersentralisasi, dan hambatan awal yang rendah karena modal pembelanjaan awal akan turun drastis. Selain keuntungan pemanfaatan komputasi awan juga memiliki berbagai kelemahan seperti kesulitan kustomisasi, masalah latensi pada jaringan, keterbatasan bandwidth dibandingkan dengan jaringan lokal, penambahan biaya bandwidth internet, serta resiko terkait dengan masalah privasi, keamanan data karena lalu lintas data melalui internet dan tersimpan di penyedia.[9]

Google App Engine (GAE)

Google Cloud platform memperbolehkan pembuatan aplikasi dan situs web, menyimpan dan menganalisa data pada infrastruktur yang dimiliki Google dengan mengambil keuntungan kecepatan dan skala kemampuan infrastruktur Google, pemakaian berdasarkan perencanaan kapasitas dengan membayar apa yang digunakan tanpa pembayaran dimuka. Produk dari Google Cloud platform terdiri dari Google App Engine, Google Compute Engine, Google Cloud Storage, Google BigQuery, Google Cloud SQL, Google Prediction API dan Google Translation API.

GAE mendukung pembuatan aplikasi web dengan menggunakan GAE SDK dengan pilihan runtime environment berupa Java, Python maupun Go. Aplikasi yang dikembangkan menurut standar App Engine di upload ke Google, dan kemudian akan disebarkan pada Google Cloud. Google menangani backup, load balancing, lonjakan akses, penyebaran, dan cache sehingga pengembang dapat berkonsentrasi pada analisa dan perancangan.[10] Aplikasi berjalan pada infrastruktur yang elastis, dan skalabilitas secara dinamis sesuai dengan trafik dan bertambah maupun berkurangnya penyimpanan dengan keterbatasan API dimana aplikasi tidak dapat menulis langsung ke sistem file tetapi harus menggunakan Datastore,

aplikasi tidak dapat membuka socket akses langsung ke host lain tetapi harus menggunakan Google URL fetch service, dan suatu aplikasi Java tidak dapat membuat thread baru.

Pada layanan cuma-cuma, aplikasi melakukan respon terhadap permintaan web dengan subdomain .appspot.com, dan juga tersedia pemakaian subdomain atau domain untuk layanan berbayar. App Engine memilih sebuah server dari banyak kemungkinan server untuk menangani permintaan berdasarkan server yang dapat memberikan kecepatan yang paling baik. Aplikasi dapat mendistribusikan trafik melalui banyak server. Aplikasi tidak dapat mengakses server secara tradisional seperti menulis file, membaca file aplikasi lain, mengakses kepada fasilitas jaringan dan perangkat keras server tetapi dapat menggunakannya melalui services. Secara ringkasnya masing-masing permintaan tinggal didalam “sandbox” masing-masing sehingga memperbolehkan App Engine menangani suatu request dengan server tertentu berdasarkan estimasi yang dapat memberikan respon tercepat. Tidak ada cara untuk memastikan suatu permintaan dijalankan pada server yang sama walaupun permintaan tersebut berasal dari klien yang sama. App Engine membatasi suatu permintaan sampai kepada 30 detik untuk mengembalikan respon kepada klien.[11]

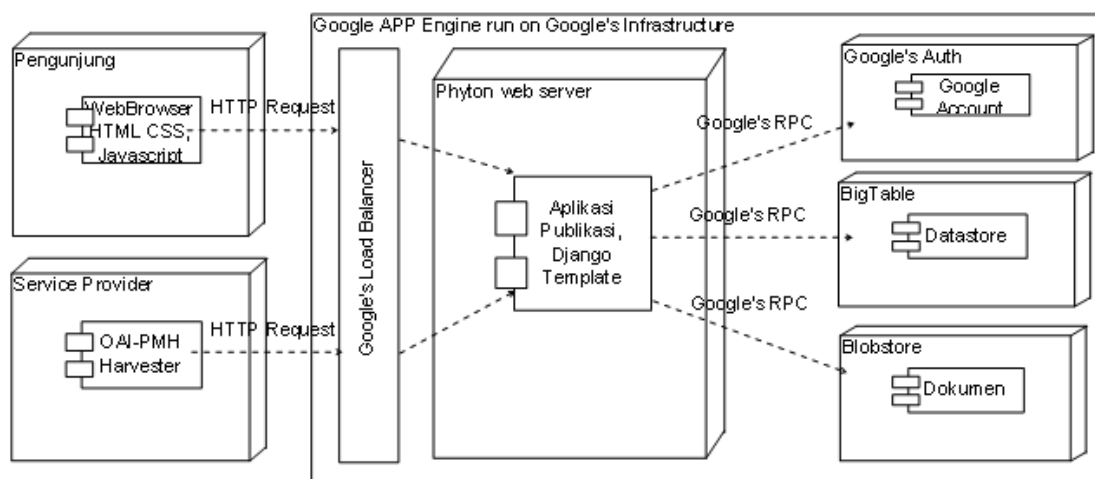
App Engine memberikan kuota cuma-cuma untuk penyimpanan sampai 1 GB dan blob-store sampai 5 GB, CPU dan bandwidth yang cukup untuk suatu aplikasi efisien melayani sekitar 5 juta tampilan halaman setiap bulannya. Ketika layanan berbayar dapat diaktifkan, maka batasan tersebut akan dinaikan, dan penanganan hanya dilakukan atas kelebihan pemakaian diatas. Jumlah maksimal sumber daya dapat dikendalikan sehingga tetap berada dalam budget.

Penyimpanan data dapat dilakukan dengan App Engine Datastore yang terintegrasi dengan layanan GAE menyediakan suatu objek penyimpanan data NoSQL schemaless dengan kemampuan query engine dan atomic transaction yang didasarkan pada BigTable. Penyimpanan data juga dapat dilakukan dengan layanan terpisah Google Cloud SQL yang merupakan database relational yang didasarkan pada MySQL RDBMS familiar, maupun Google Cloud Storage untuk penyimpanan data objek dan file yang dapat diakses melalui aplikasi Python dan Java.

Blobstore API digunakan pada aplikasi untuk melayani objek data yang disebut sebagai blob yang mana lebih besar dibandingkan dengan ukuran yang diperbolehkan untuk objek pada layanan Datastore. Blob adalah bermanfaat untuk melaynai file besar seperti video dan file gambar, dan memperbolehkan pemakai untuk mengupload file besar.

3. Hasil dan Pembahasan

Aplikasi dikembangkan dengan menggunakan App Engine SDK. Bahasa pemrograman yang digunakan Python yang berjalan diatas Python webserver, HTML5, CSS dan JavaScript yang berjalan di, XML untuk Django Template dan ekspose metadata melalui OAI-PMH 2,0 dan penyimpanan data berbasis BigTable pada App Engine datastore dan blob-store API, berikut ini adalah *diagram deployment* dari aplikasi yang ditunjukkan pada Gambar 1.



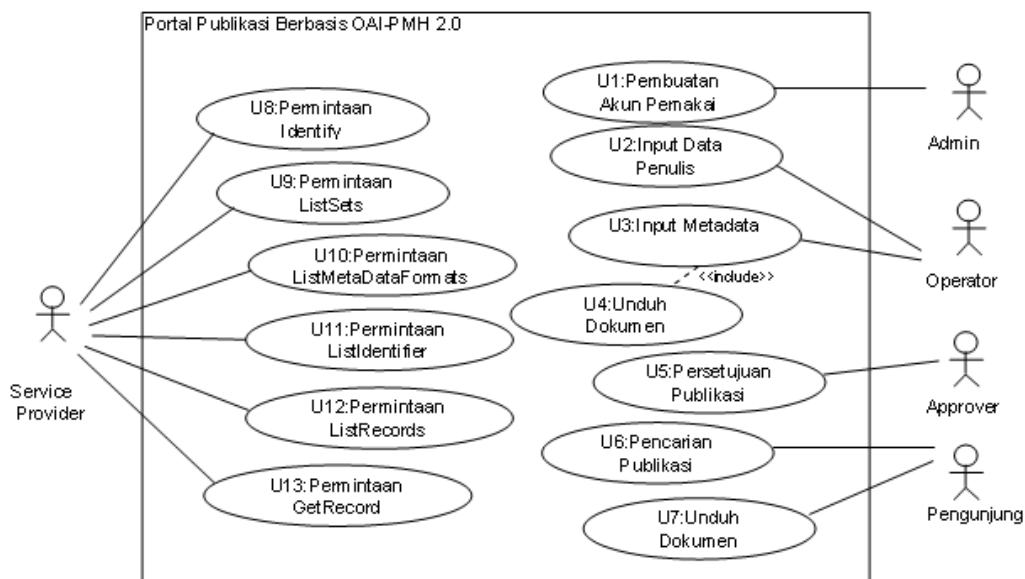
Gambar 1. Deployment Diagram Aplikasi

A. Pernyataan Tujuan

Repositori Publikasi berbasis OAI-PMH 2.0 adalah aplikasi berbasis web untuk mempublikasikan laporan hasil penelitian dosen dalam bentuk dokumen PDF dan ekspos metadata yang dispesifikasikan pada Panduan Kontributor Portal Garuda. Aplikasi web harus memiliki fasilitas pencarian berdasarkan kata kunci yang dimasukkan oleh pemakai, dan dokumen hasil pencarian dapat diunduh oleh pemakai. Aplikasi juga harus menyediakan kemampuan ekspose metadata berdasarkan protokol OAI-PMH 2.0 sehingga ekspose metadata tersebut dapat dipanen secara online oleh suatu *service provider*. Aplikasi ditargetkan untuk beroperasi pada platform Google App Engine yang merupakan layanan Cloud Computing Platform as Service (PaaS) dari Google.

B. Use Case

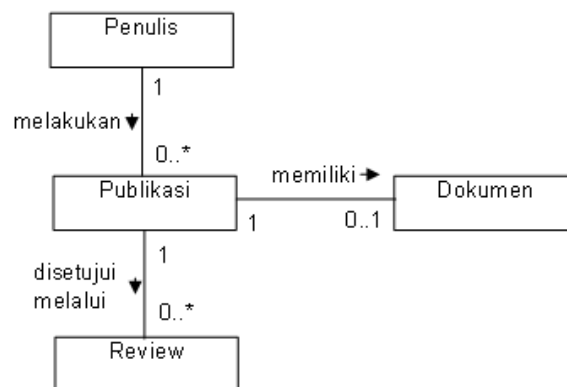
Aktor-aktor yang terlibat didalam pengoperasian dan pemanfaatan sistem adalah Admin yang bertugas setting parameter portal publikasi dan pembuatan account pemakai, Supervisor yang bertugas melakukan persetujuan ataupun pembatalan terhadap metadata dan dokumen yang ajukan, Operator yang bertugas untuk memelihara data usulan publikasi, Pengunjung yang melakukan pencarian meta serta unduh dokumen, dan *Service Provider* yang melakukan panen metadata penelitian melalui OAI-PMH 2.0. Adapun Use Case Portal publikasi ditunjukkan pada Gambar 2.



Gambar 2. Use Case Diagram

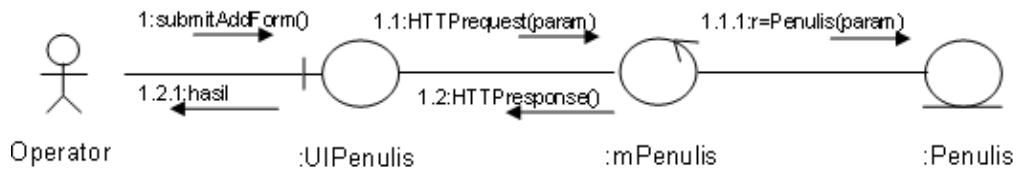
C. Analisa Sistem

Static class-class yang menyusun sistem adalah Class Penulis, Class Publikasi, Class Dokumen, dan Class Review sebagaimana yang ditunjukkan pada Gambar 3 sebagai berikut:

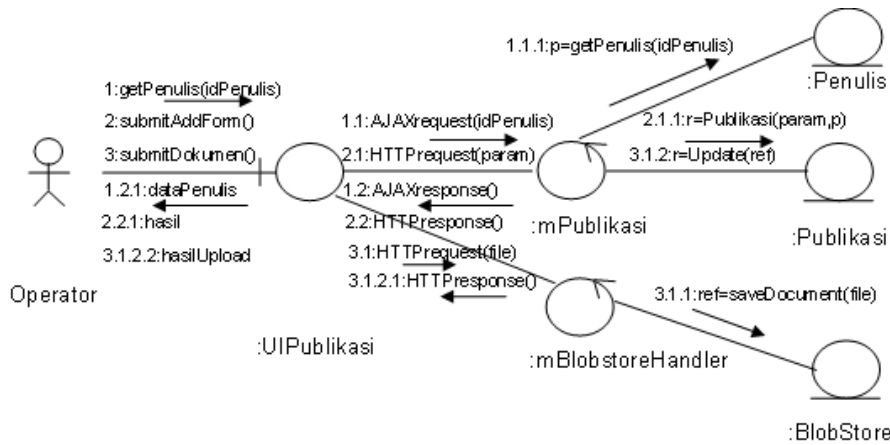


Gambar 3. Class Diagram

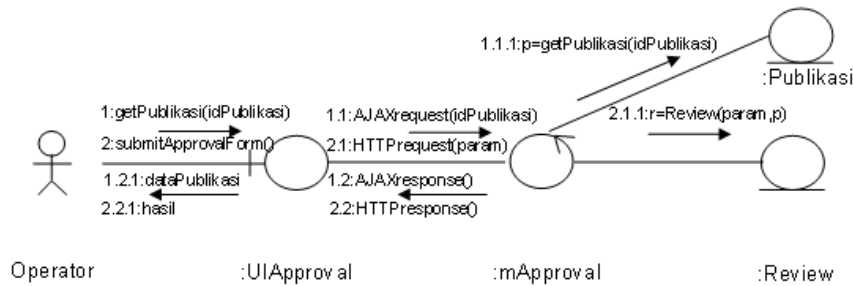
Aplikasi dikembangkan dengan pendekatan Model View Controller (MVC), adapun gambaran dinamis komunikasi antar Class-Class yang berinteraksi pada sistem ditunjukkan pada Gambar 4, Gambar 5, dan Gambar 6 sebagai berikut:



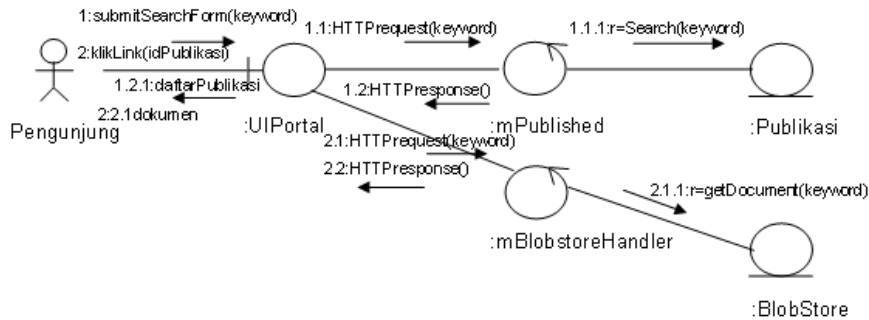
Gambar 4. Diagram Komunikasi untuk Use Case Input Penulis



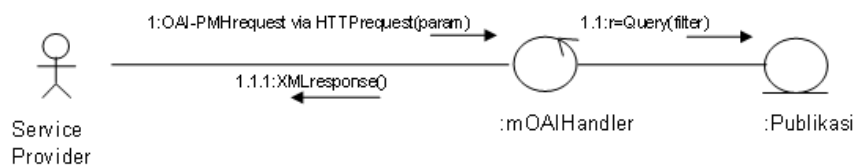
Gambar 5. Diagram Komunikasi untuk Use Case Input Meta Data dan Unggah Dokumen



Gambar 6. Diagram Komunikasi untuk Persetujuan Publikasi



Gambar 6. Diagram Komunikasi untuk Pencarian dan Unduh Dokumen oleh Pengunjung



Gambar 7. Diagram Komunikasi Panen Metadata oleh *Service Provider*

D. Rancangan Sistem

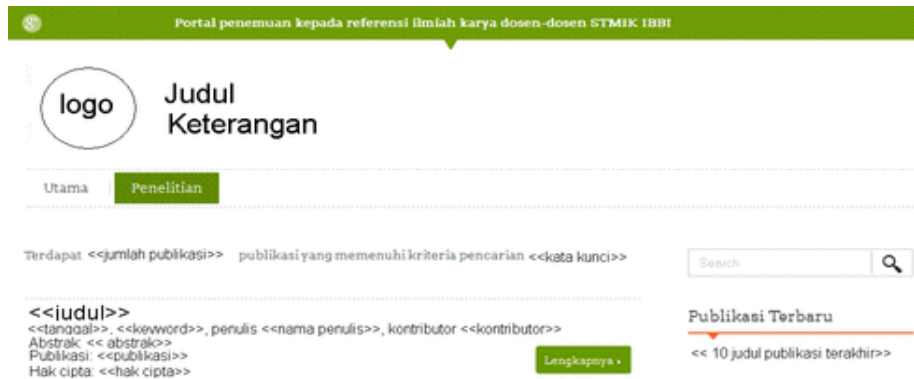
App Engine Datastore merupakan objek penyimpanan data yang *schemaless* yang didasarkan pada BigTable, dimana model penyimpanan data didefinisikan langsung pada program aplikasi. Struktur penyimpanan masing-masing entitas dalam bentuk *aggregate* adalah sebagai berikut:

```

Review: {
  review_no:
  review_time:
  review_by:
  status: [baru, disetujui, ditahan, ditolak]
  description:
  Publikasi:{
    publish_no:
    status: [baru, disetujui, ditahan, ditolak]
    approve_no:
    approve_by:
    approve_date:
    Penulis:{
      name:
      instituton:
      contact:
    }
    subject:
    description:
    publisher:
    contributor:
    publish_date:
    material_type:
    permalink:
    right:
    journal_title:
    document: blobstore.blobReference
  }
}

```

Untuk entitas Publikasi merupakan subclass dari `search.SearchableModel` untuk memungkinkan fulltext search, sedangkan Review dan Penulis adalah subclass dari `db.Model`.



Gambar 8. Antar Muka Hasil Pencarian dan Download

Rancangan untuk template XML dalam format Django untuk respon metadata atas permintaan GetRecord yang diajukan oleh suatu *Service Provider* adalah sebagai berikut ini:

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>{{ responsedate }}</responseDate>
  <request verb="GetRecord"
    {% if not identifier == "" %}indetifier="{{ identifier }}"{% endif %}
    {% if not metadataprefix == "" %}metadataPrefix="{{ metadataprefix }}"{% endif %}
    >{{ alamatakses }} </request>
  {% if not errormessage == "" %}
  <error code="{{ errormessage }}" />
  {% else %}
  <GetRecord>
  <record>
  <header>
  <identifier>oai:{{ alamatweb }}:{{ entity.key.name }}</identifier>
  <datestamp>{{ entity.status_date|date:"Y-m-d" }}T00:00:00Z</datestamp>
  </header>
  <metadata>
  <oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title>{{ entity.title }}</dc:title><dc:creator>{{ entity.creator }}</dc:creator>
  <dc:subject>{{ entity.subject }}</dc:subject><dc:description>{{ entity.description }}</dc:description>
  <dc:publisher>{{ entity.publisher }}</dc:publisher>
  <dc:contributor>{{ entity.contributor }}</dc:contributor>
  <dc:date>{{ entity.publish_date|date:"d-M-Y" }}</dc:date>
  <dc:type>{{ entity.material_type }}</dc:type>
  <dc:identifier>http://{{ alamatweb }}/permalink?key={{ entity.publish_no }}</dc:identifier>
  <dc:source>{{ entity.journal_title }}</dc:source>
  <dc:rights>{{ entity.right }}</dc:rights><dc:language>Indonesian</dc:language>
  </oai_dc:dc>
  </metadata>
  </record>
  </GetRecord>
  {% endif %}
```


</OAI-PMH>

E. Implementasi

Aplikasi diimplementasikan untuk publikasi hasil penelitian dosen-dosen STMik IBBI yang diberi Portal Garuda STMik IBBI. Adapun hasil penelitian yang dimasukan adalah penelitian dari tahun 2011 dan 2012. Berikut ini adalah tampilan antar muka hasil pencarian oleh pengunjung berdasarkan kata kunci tertentu. Portal Garuda STMik IBBI dapat dicapai dengan URL <http://research.lppm-stmik.ibbi.ac.id>.

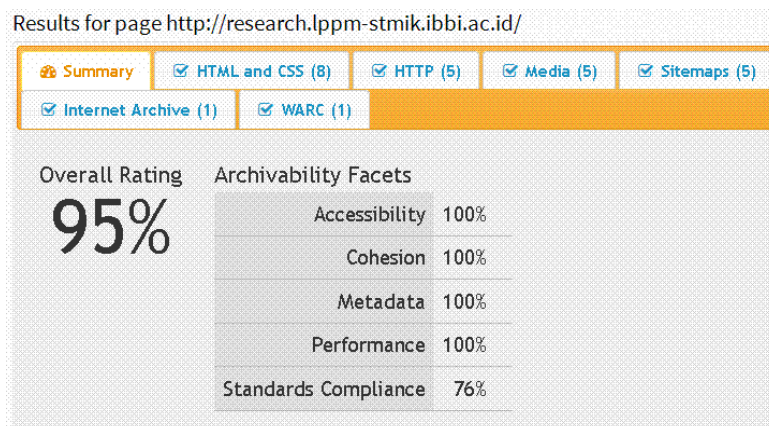


Gambar 8. Tampilan Portal Garuda STMik IBBI

F. Pengujian

Untuk memastikan bahwa aplikasi hasil rancangan dan pengembangan oleh peneliti telah memenuhi standar OAI-PMH 2.0 maka perlu dilakukan pengujian dengan menggunakan OAI-PMH Validator yang menguji kesesuaian respon dengan OAI-PMH XML Schema, pengujian dengan menggunakan BASE OAI-PMH Validtor yang mensimulasikan proses panen oleh suatu *service provider* untuk memastikan kemampuan portal menangani *granuality harvesting* dan pemenuhan metadata terhadap minimal elemen DC(*date, creator, identifier, type, dan title*). Pada setiap tahapan pengujian akan diberikan berbagai saran perbaikan, sehingga pengujian ini bersifat iteratif, yaitu pengujian kemudian dilakukan berbagai penyesuaian dan diuji kembali sampai tidak ditemukan lagi isu yang berarti.

Setelah melalui proses pengujian pemenuhan standard OAI-PMH, maka berikutnya perlu diukur rating portal terkait dengan tingkat keberhasilan portal dapat diarchive dengan baik oleh mesin pencarian. Pengujian akan menggunakan WebArchivability yang melakukan pemeriksaan terhadap kesesuaian standard W3C terkait dengan HTML dan CSS, pemeriksaan terhadap death-link, pemeriksaan terhadap pemakaian media seperti format gambar yang terkompres dan standard, ketersediaan sitemap untuk meningkatkan akses seperti robot.txt, sitemap.xml, RSS, Etag. Berdasarkan hasil rating yang berhasil dicapai adalah 95% dengan isu terkait dengan pemenuhan standard W3C 76%, dimana peneliti banyak menggunakan standard CSS3 padahal pemeriksaan menggunakan standard CSS2 sehingga dianggap bukan menjadi isu. Hasil rating ditunjukkan pada Gambar 9 berikut ini:



Gambar 9. Hasil Rating Portal Garuda STMik IBBI dengan WebArchivability

Untuk meningkatkan penyebaran dan akses Portal Garuda STM IK IBBI, maka portal perlu didaftarkan pada beberapa portal yang mempromosikan repositori berbasis OAI-PMH seperti OpenArchive.ORG, OpenDOAR, dan ROAR. Pendaftaran pada OpenArchive akan diterima jika berhasil melalui dua tahapan validasi yaitu *Comformance Testing for Basic Functionality*, dan *Comformance Testing for Error and Exception Handling*. Hasil pendaftaran ditunjukkan pada Gambar 10.



element name	element value
Base URL	http://research.lppm-stmik.ibbi.ac.id/oai2_0
Repository Name	Portal Garuda STM IK IBBI
Protocol Version	2.0
Email	admin@lppm-stmik.ibbi.ac.id
Registration Date	2013-09-13T15:59:52Z
Date Last Validated	Fri Sep 13 15:59:52 2013
OAI Repository ID	

If you are the maintainer of this repository, you may to update the information recorded to match new information exposed via the Identify response by running the validation/registration process again. Go to the [validation page](#) and select "Register this site".

Wed Sep 18 04:12:54 2013

Gambar 10. Portal Garuda STM IK IBBI terdaftar di OpenArchive.ORG

Pendaftaran pada OpenDOAR, dan ROAR adalah melalui proses moderator, yang artinya akan dikunjungi oleh staf institusi tersebut, kemudian melakukan validasi kebenaran data yang dimasukkan pada saat pendaftaran dengan Portal Publikasi yang tersedia. Hasil Pendaftaran pada OpenDOAR dan ROAR ditunjukkan pada Gambar 11 dan Gambar 12 sebagai berikut:

[Portal Garuda STM IK IBBI \(STM IK IBBI Repository\)](#)

Organisation: [STM IK IBBI](#), Indonesia

Description: This site provides access to the research output of the institution. The interface is available in Indonesian and English.

OAI-PMH: http://dosen.publikasistmikibbi.lppm.org/oai2_0

Software: GAE

Size: 44 items (2013-09-20)

Subjects: Computers and IT

Content: Articles

Languages: Indonesian

Policies: Metadata re-use permitted for not-for-profit purposes; Re-use of full data items permitted for not-for-profit purposes; Content policies defined; Submission policies defined; Preservation policies defined

OpenDOAR ID: 2771 , [Suggest an update for this record](#),
Link to this record: <http://opendoar.org/id/2771/>

Gambar 11. Portal Garuda STM IK IBBI terdaftar di OpenDOAR

Record

ROAR ID:	6427								
Home Page:	http://research.lppm-stmik.ibbi.ac.id								
Repository Type:	Research Institutional or Departmental								
Organisation:	STMIK IBBI								
Additional Information:	Portal Garuda is a portal to discover scientific reference works from STMIK IBBI's lecturer, which allows access to e-journals and e-books domestic. This portal was developed by the STMIK IBBI Research and Community Service (LPPM STMIK IBBI, Hendra Soewarno).								
Software:	Other softwares (various) (version other)								
Country:	Asia > Indonesia								
Location:	<table border="1"> <thead> <tr> <th>Country</th> <th>City</th> <th>Latitude</th> <th>Longitude</th> </tr> </thead> <tbody> <tr> <td>Indonesia</td> <td>Medan</td> <td>3.58524</td> <td>98.6756</td> </tr> </tbody> </table>	Country	City	Latitude	Longitude	Indonesia	Medan	3.58524	98.6756
Country	City	Latitude	Longitude						
Indonesia	Medan	3.58524	98.6756						

Gambar 12. Portal Garuda STMIK IBBI terdaftar di ROAR

4. Kesimpulan dan Saran

Berdasarkan hasil validasi Repositori Publikasi dengan OAI-PMH Validator, BASE OAI-PMH Validator dan berhasil didaftarkan ke OpenArchive, OpenDOAR dan ROAR maka dapat disimpulkan bahwa aplikasi repositori yang dikembangkan telah memenuhi standard OAI-PMH 2.0 dan dapat diharvest dengan baik oleh OAI-PMH *service provider*. Dengan hasil *rating* 95% dari WebArchivability maka dapat diyakini bahwa Repositori Publikasi dapat diarchive dengan baik oleh mesin pencarian. Akhirnya dengan tersedianya Aplikasi Repositori Publikasi hasil rancangan peneliti dapat membantu perguruan tinggi untuk memenuhi kewajiban publikasi karya ilmiah dan jurnal sesuai dengan surat edaran Dikti nomor 2050/E.T/2011 yang dapat diakses secara luas.

Berdasarkan pengalaman selama pembuatan prototipe aplikasi, ketersediaan beberapa fitur pada Google App Engine seperti autentikasi menggunakan Google Account, *full-text* search dan *blobstore* dapat memudahkan pengembangan aplikasi Repositori Publikasi.

Untuk meningkatkan akses terhadap portal publikasi, maka disarankan untuk mendaftarkan portal ke Google WebMaster Tools, dan BING Webmaster Tools, dan Google Scholar.

Ucapan Terima Kasih

Terima kasih kepada Direktorat Jenderal Pendidikan Tinggi yang mendukung pembiayaan penelitian ini sesuai dengan Surat Perjanjian Penugasan Dalam Rangka Pelaksanaan Program Penugasan Penelitian Dosen Pemula Bagi Perguruan Tinggi Swasta di Lingkungan Kopertis Wilayah-I Tahun Anggaran 2013 Nomor: 279/K.1.2.1/PS/2013.

Daftar Pustaka:

- [1] Carl I, Herbert Van de S., Michael N., & Simeon W. (2008), The Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0, [Online] Tersedia <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>
- [2] CDW-G. (2011). *From tactic to strategy: The CDW 2011 cloud computing tracking poll*. Tersedia <http://webobjects.cdw.com/webobjects/media/pdf/Newsroom/CDW-Cloud-Tracking-Poll-Report-0511.pdf>
- [3] Sasikala, S., & Prema, S., *Massive Centralized Cloud Computing (MCCC) Exploration in Higher Education*. *Advances in Computational Sciences and Technology*, 3 (2), pp. 111-118, 2010.
- [4] OpenArchives, Open Archives Iniatif – Protocol for Metadata Harvesting, Tersedia Online: <http://www.openarchives.org/OAI/openarchivesprotocol.htm>
- [5] (2012) Dublin Core Metadata Element Set, Version 1.1, Tersedia <http://dublincore.org/documents/dces/>
- [6] CDP Metadata Working Group (2006), Dublin Core Metadata Best Practice Version 2.1.1
- [7] Marinela, M., & Anca Ioana, A., *Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis*, IBIMA Publishing, Vol. 2011, Article ID 875547, 15 pages.
- [8] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, NIST Special Publication 800-145, September 2011.

- [9] Katz, R. N., Goldstein, P. J. & Yanosky, R. *Demystifying cloud computing for higher education*, EDUCAUSE Center for Applied Research Bulletin, 19, 1-13. 2009.
- [10] B. Sosinsky, *Cloud Computing Bible*, Willy Publishing, Inc, Indiana, 2011.
- [11] *Google Apps for Education*, Tersedia <http://www.google.com/enterprise/apps/education/>, diakses pada 19 Agustus 2013