

REUSE PLUGIN OAI-PMH HARVESTER UNTUK HARVESTING SECARA TERJADWAL PADA UNION CATALOG OMEKA

Tajul Furqan^{#1}, Taufiq Abdul Gani^{#2}, Ramzi Adriman^{#3}

[#] *Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala*

Jl. Tgk. Syech Abdul Rauf No. 7, Darussalam, Banda Aceh 23111 Indonesia

¹ *tajulfurqan@gmail.com*

² *topgan@unsyiah.ac.id*

³ *ramzi.adriman@unsyiah.ac.id*

Abstrak—Penelitian ini bertujuan untuk mengembangkan *plugin* OAI-PMH harvester pada aplikasi OMEKA agar aplikasi ini dapat melakukan *harvesting* secara otomatis dan terjadwal. Proses pengembangan aplikasi OMEKA menggunakan metode *waterfall* model, yaitu pengembangan dari sistem ini dilakukan secara berurutan. *Plugin* yang telah dikembangkan berikutnya diuji dengan menggunakan metode pengujian *whitebox testing*. Hasil pengujian menunjukkan bahwa aplikasi dapat dipakai dan secara umum telah sesuai dengan kebutuhan Perpustakaan Universitas Syiah Kuala, hanya diperlukan beberapa penyesuaian pada bagian-bagian.

Kata Kunci— OMEKA, OAI-PMH Harvester, Union Catalog, Harvesting, Cron.

I. PENDAHULUAN

Pada saat ini Perpustakaan Universitas Syiah Kuala telah membangun sebuah *Union Catalog* disistem informasi, hal ini dilakukan agar dapat lebih mempermudah para pengguna untuk memperoleh informasi yang dibutuhkan. Pengertian dari *Union Catalog* sendiri adalah suatu database katalog yang merupakan gabungan dari beberapa katalog yang dimiliki oleh beberapa repository lokal yang ada. Untuk dapat melakukan sistem *Union Catalog* di suatu sistem informasi diperlukannya suatu aplikasi, agar sistem tersebut dapat berjalan dengan baik. Aplikasi yang digunakan perpustakaan untuk membuat *Union Catalog* adalah dengan menggunakan aplikasi OMEKA.

Adapun permasalahan yang muncul pada Aplikasi OMEKA Perpustakaan Universitas Syiah Kuala saat ini ialah ketika melakukan *Harvesting* masih dilakukan secara manual. Proses *harvesting* akan dilakukan jika *repository-repository* yang tergabung pada sistem *union catalog* perpustakaan Universitas Syiah Kuala melakukan update atau terdapat perubahan pada data-data katalog mereka. Dengan kata lain user harus melakukan pengecekan pada *repository-repository* lokal tersebut. Permasalahannya ialah *repository* yang tergabung pada *union catalog* OMEKA pada perpustakaan lebih dari satu. *Repository* yang tergabung pada *union catalog* OMEKA pada perpustakaan Unsyiah adalah *repository* lokal Universitas Syiah Kuala, ETD dan OPAC. Oleh karena itu untuk melakukan pengecekan satu

persatu pada *repository-repository* tersebut akan memakan banyak waktu. Jika data-data katalog dari *repository* tersebut diperbarui, user akan melakukan proses *harvesting*. Dengan kata lain proses kerja akan dilakukan secara berulang-ulang setiap kali *repository* lokal tersebut diperbarui atau terdapat perubahan pada data katalog mereka.

Berdasarkan permasalahan tersebut dibutuhkan suatu sistem otomatis dan terjadwal untuk melakukan *harvesting* pada Aplikasi OMEKA. Salah satu solusinya ialah dengan mengembangkan atau memodifikasi *plugin* OAI-PMH Harvester di Aplikasi OMEKA, *Plugin* yang telah dikembangkan tersebut akan bertugas untuk melakukan *harvesting* secara otomatis. Dengan adanya pengembangan pada aplikasi OMEKA tersebut, diharapkan Perpustakaan Universitas Syiah Kuala dapat lebih mudah lagi dalam menjalankan *Union Catalog* di lingkungan Universitas Syiah Kuala.

II. TINJAUAN PUSTAKA

A. *Union Catalog*

Union Catalog merupakan suatu database katalog yang merupakan gabungan dari beberapa katalog yang dimiliki oleh beberapa perpustakaan yang ada. Manfaat dari *Union Catalog* ialah agar dapat menemukan data bahan perpustakaan yang diperlukannya, sekaligus mengetahui lokasi bahan perpustakaan tersebut [1].

B. OMEKA

OMEKA merupakan aplikasi web yang sangat cocok digunakan pada perpustakaan untuk mengelola konten seperti tulisan yang berupa karya ilmiah, buku, dan journal-journal. Selain digunakan pada perpustakaan dapat juga digunakan di museum dan bidang arsip. OMEKA memiliki beberapa catatan seperti tampilan home menarik, mudah diinstal, mudah digunakan dan didesain untuk kebutuhan *repository* yang tematis [2].

OMEKA bersifat *open source* (sumber terbuka). Menyediakan beberapa fitur untuk para pengguna aplikasi tersebut agar dapat berkontribusi bersama dalam mengembangkan aplikasi tersebut. seperti pada fitur design

untuk kontribusi *themes*, fitur *develop* untuk kontribusi *code* dan *plugin* serta fitur *document* untuk menambahkan dokumentasi yang belum tersedia. Selain itu OMEKA merupakan *software* yang multiplatform sehingga dapat digunakan di beberapa oprating sistem (OS) seperti pada *Windows*, *Linux* dan *Mac* [3].

C. OAI-PMH

OAI (*Open Archive Initiative*) merupakan sebuah terobosan mengembangkan dan mempromosikan standart interoperabilitas yang bertujuan untuk memfasilitasi penyebaran konten agar lebih efisien. Istilah “*archieve*” pada *Open Archive Initiative* menunjukkan asal mula OAI pada komunitas pra-cetak (*e-print*). *Archieve* juga bisa dikatakan sinonim dari repositori naskah ilmiah pelajar/mahasiswa. Orang yang bergerak di bidang pengarsipan memiliki kewenangan untuk mencatat definisi sebuah arsip pada domain mereka dengan tujuan pelestarian data dalam jangka waktu yang lama serta sesuai perundang-undangan dan kebijakan institusional. Penggunaan istilah arsip di sini dalam arti luas berarti sebuah *repository* (tempat menyimpan) kumpulan informasi.

OAI-PMH adalah sebuah *harvesting* protokol yang digunakan untuk berbagi metadata antar layanan. Pada konteks OAI, *harvesting* (memanen) berarti mengumpulkan metadata dari berbagai *repository* dalam satu wadah atau tempat penampung data. Pada OAI-PHM ada dua objek yang saling berhubungan yaitu data *provider* dan *service provider* [4].

D. Harvesting

Harvesting adalah teknik yang digunakan untuk menarik atau memanen sejumlah besar data dari situs web dan menggunakannya untuk tujuan tertentu. Sebagai cara yang mudah untuk mengumpulkan data secara online, teknik ini sering digunakan untuk mngambil informasi website seperti teks, foto, alamat email, dan daftar kontak, yang telah di berikan izin pada penggunaannya dalam pengambilan data tersebut.

E. Cron

Cron adalah suatu fasilitas yang telah tersedia di sistem operasi linux atau Unix yang bertugas untuk mengatur tugas secara terjadwal. *Cron* dibuat menggunakan command program yang disebut *crontab*. Eksekusi perintah secara otomatis pada waktu tertentu dapat dilakukan dengan menggunakan *Cron*. Oleh sebab itu *Cron* akan sangat membantu ketika akan melakukan suatu pekerjaan yang memilki interval tertentu [5].

Frekuensi waktu terkecilnya adalah 1 menit sekali. Frekuensi terbesarnya hingga beberapa tahun sekali. Dapat juga dipilih ingin menjalankan setiap hari tertentu, setiap berapa jam sekali, pada menit keberapa.

F. Waterfall Model

Waterfal model adalah suatu metode yang digunakan untuk mengembangkan suatu perangkat lunak, metode ini

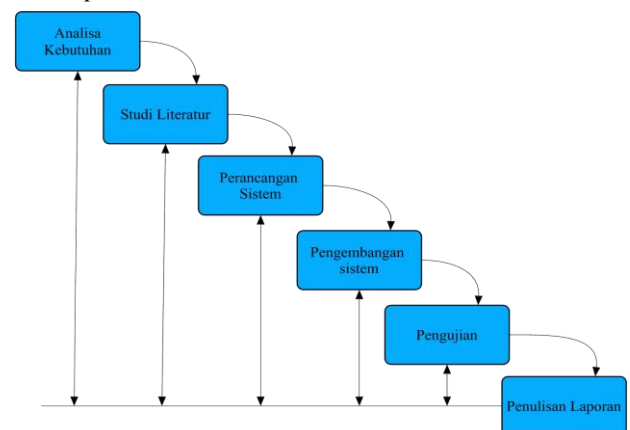
dikenal sejak tahun 1970 dan diperkenalkan oleh Royce, pada saat itu disebut sebagai siklus klasik dan sekarang ini lebih dikenal dengan sekuensial linier. Selain itu Model ini merupakan model yang paling banyak di pakai oleh para pengembang software. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melanjutkan kelangkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

G. Whitebox Testing

Whitebox testing adalah pengujian yang dilakukan untuk mendeteksi kesalahan-kesalahan logika pada kode program. Pengujian whitebox testing dilakukan untuk menguji desain aplikasi pada tingkat terendah (*low level*). Selain itu pengujian whitebox testing juga dilakukan untuk menguji integrasi antar dua atau lebih unit yang saling bekerja sama, serta menguji sistem kerja fungsi tertentu pada desain *high level*. Pengujian ini harus dilakukan oleh orang yang memiliki pengetahuan yang baik terhadap struktur program [6].

III. METEOROLOGI PENELITIAN

A. Tahapan Penelitian



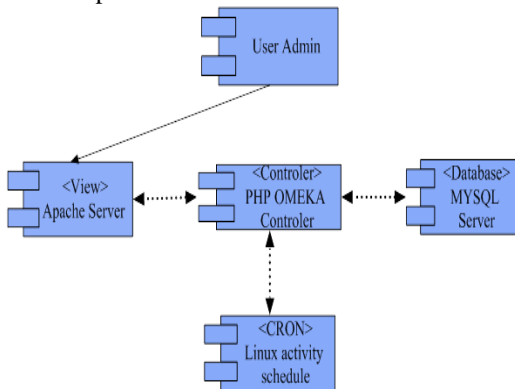
Gambar 1 tahapan penelitian

Tahapan dalam menyelesaikan tugas akhir ini dimulai dengan analisa kebutuhan untuk cara kerja dari objek yang diteliti, berikutnya akan dilanjutkan dengan studi literature untuk mencari informasi yang dibutuhkan, serta journal yang terkait tentang perpustakaan digital, OMEKA, OAI-PMH, *harvesting*, metadata dan *cron*. Berdasarkan hasil dari analisa kebutuhan dan studi literatur yang telah dilakukan sebelumnya, kemudian akan dilakukan perancangan system. Pancangan sistem ini akan meliputi proses kerja aktivitas-aktivitas *user* admin dalam bentuk *activity* diagram dan pemodelan sistem aplikasi yang akan dibangun. Selain itu untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukan akan dibuat suatu *usecase* diagram. Pemodelan sistem dilakukan dalam bentuk *component* diagram.

Tahapan berikutnya akan dilakukan pengembangan aplikasi, pengembangan dilakukan sesuai dengan hasil dari perancangan yang telah dilakukan sebelumnya. Setelah pengembangan selesai dilakukan berikut akan dilakukan pengujian, pada pengujian dilakukan dengan menggunakan dua metode, yaitu metode whitebox testing dan usability testing. Tahap akhir dari penelitian yang dilakukan, yaitu penyusunan laporan. Diharapkan peneliti mampu menarik kesimpulan dari hasil penelitian yang dilakukan.

B. Perancangan Sistem

Pada perancangan sistem digunakan bentuk *Component diagram*. *Component diagram* adalah diagram yang menggambarkan komponen-komponen yang terdapat pada suatu sistem. *Component diagram* dari sistem yang dibangun ini dapat dilihat pada Gambar 2.



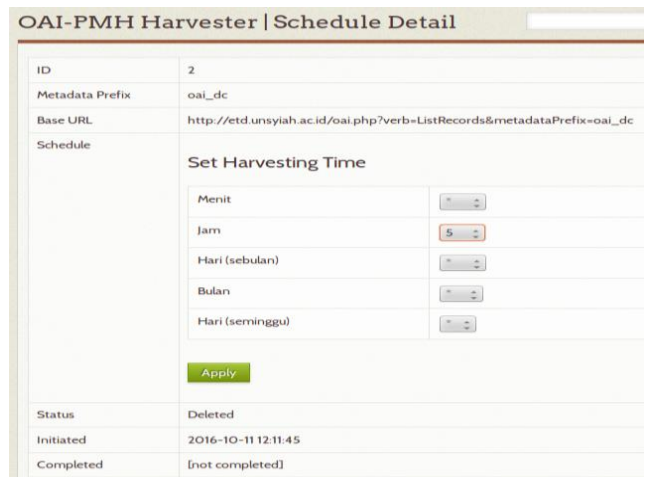
Gambar 2 Component diagram sistem

Seperti gambar 2 dapat dilihat hubungan antar masing-masing komponen yang ada pada aplikasi. Pada aplikasi admin membuka halaman *plugin* OAI-PMH Harvester ,pada bagian ini tampilan yang diterima oleh user admin merupakan *View* yang direspon oleh *Apahce server*,setelah *user* admin mengisikan kolom-kolom yang diperlukan, data dari setiap kolom akan dikirimkan ke *controller* OMEKA dalam variabel-variabel *post* tertentu, data yg diterima selanjutnya diteruskan atau di *input* kedalam *database* MYSQL, jika proses *input* berhasil selanjutnya *controller* akan mengirimkan data yang telah tersimpan di *database* ke dalam *crontab*.

IV. HASIL DAN PEMBAHASAN

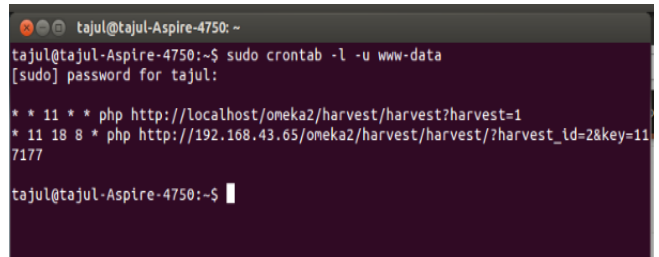
A. Screenshot Halaman pengaturan jadwal harvesting

Pada Gambar 3 adalah *view* atau tampilan untuk melakukan pengaturan jadwal *Harvesting*. *User* dapat melakukan pengaturan waktu dengan frekuensi yang paling terkecil yaitu menit sampai yang paling besar adalah pertahun.



Gambar 3 Tampilan pengaturan jadwal *harvesting*.

Setelah *user* melakukan pengaturan waktu pada halaman *schedule detail*, data dari pengaturan tersebut kemudian akan dikirimkan ke *controller* OMEKA dalam variabel-variabel *post* tertentu, data yg diterima selanjutnya diteruskan atau di *input* kedalam *database* MYSQL, jika proses *input* berhasil selanjutnya *controller* akan mengirimkan data yang telah tersimpan di *database* ke dalam *crontab*. Berikut ini adalah hasil *screenshot* dari terminal *Cron*.



Gambar 4 Hasil *screenshot* terminal *Cron*

B. Whitebox Testing

Pengujian ini dilakukan untuk mendeteksi kesalahan-kesalahan logika pada program sehingga menghasilkan *output* yang sesuai dengan spesifikasi yang dibutuhkan. Pengujian ini dilakukan terhadap sistem kerja aplikasi pada setiap aktivitas-aktivitas *user* admin. Dari hasil pengujian terdapat sebanyak 5 aktivitas yang seluruhnya telah berhasil diuji. Dari hasil pengujian ini dapat disimpulkan bahwa hasil *output* dari seluruh aktivitas telah sesuai dengan spesifikasi yang dibutuhkan.

V. PENUTUP

A. kesimpulan

Adapun kesimpulan yang didapatkan pada penelitian ini adalah, setelah dilakukannya Pengujian dengan menggunakan metode *white box*, didapatkan bahwa *plugin* telah berjalan sesuai dengan yang diharapkan.

B. Saran

Berdasarkan kesimpulan penelitian, maka penulis merekomendasikan berupa saran yaitu, Disarankan pada pengembangan selanjutnya agar program dapat mengenal kondisi ketika *harvesting* gagal dilakukan.

UCAPAN TERIMA KASIH

Dengan terselesaikannya Karya Ilmiah ini, penulis mengucapkan terimakasih yang sedalam-dalamnya kepada :

1. Allah SWT atas limpahan karunianya sehingga penulis dapat menyelesaikan Karya Ilmiah ini.
2. Orang tua dan keluarga yang telah banyak memberikan bantuan, dukungan, semangat dan doa.
3. Bapak Dr. Taufiq A Gani, S.Kom.,M.Eng.Sc dan Bapak Dr. Ramzi Adriman, ST.,M.Sc selaku Dosen Pembimbing I dan Dosen Pembimbing II yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan karya ilmiah ini.
4. Bapak Fardian, S.T., M.Sc. dan Bapak Rahmad Dawood, S.Kom.,M.Sc. selaku Dosen Penguji I dan Dosen Penguji II, serta Bapak Sayed Muchallil, ST.,M.Sc selaku komite seminar karya ilmiah ini.
5. Teman-teman mahasiswa teknik elektro, khususnya angkatan 2010 dan seluruh pihak yang telah membantu.

Akhir kata, semoga Allah SWT membalas segala kebaikan semua pihak yang telah membantu dalam penyusunan skripsi ini. Semoga skripsi ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

REFERENSI

- [1] Hendriyadi "Pengembangan Katalog Induk *Online* Perpustakaan Lingkup Badan Litbang Pertanian," Jurnal perpustakaan Pertanian Vol 18, Nomor 2,2009 .
- [2] Wicaksono, H. Memilih Digital Library Management System. Jakarta: Slide Persentasi Seminar Manajemen Repositori Institusi Dalam Meningkatkan Peringkat Webometrics. (2015).
- [3] Saputra, D.F. Repositori institusi berbasis omeka(studi kasus di Medical Knowledge Center-FK UPNVJ). *Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan Khizanah Al-Hikmah*, 3(1), 11-19, 2015
- [4] Wiji Suwarno "Memperbincangkan Penerapan *Open Acces* Untuk Koleksi *Institutional Repository*" *Libraria*, Vol 2, No 1 (2014)
- [5] Slamet Yogo Prasetyo, "OpenWRT Dengan MJPG Streamer Dan Motion Di Router HG553 Untuk Mengembangkan CCTV," Jurusan Teknik Elektro, Universitas Negeri Semarang , Indonesia, 2015
- [6] Nidhra, Srinivas, Dondeti, Jagruthi Black Box and White Box Testing Techniques-A Literature Review, *International Journal of Embedded Systems and Applications (IJESA)* Vol.2, No.2, 2012