

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian ini berkaitan dengan pembuatan *Institutional Repositories* (Repositori) dengan berbagai macam platform sudah beberapa kali dilakukan. Penelitian yang berjudul “Rancangan Bangun Aplikasi *Search Engine Repository Software Desktop* Menggunakan Teknik *Web Extraction*” oleh Dako Adi Ahmad, Wamiliana, Wisnu Wardhana (Jurusan Matematika FMIPA Unila, CV. Linux Lampung). Dalam penelitiannya dikatakan bahwa pada perkembangan teknologi, *software* memiliki peran sangat penting salah satunya sebagai media informasi dan komunikasi manusia. Apabila mengetikkan kata kunci *software* pada mesin pencarian maka akan akan banyak sekali situs-situs yang menyediakan *software* tersebut. Akan tetapi kekhawatiran tentang sumber dari *software* tersebut karena banyaknya malware/virus yang disisipkan pada *software*.

Selain itu, penelitian lain yang berkaitan dengan pembuatan repositori (Sigit Wahyudi, 2010) membangun sebuah repositori yang bertujuan untuk kebutuhan fungsional Perpustakaan Sains dan Teknologi Universitas Negeri Syarif Hidayatullah Jakarta dan membuat aplikasi digital yang dapat memudahkan dalam mengakses serta melakukan *download file* digital yang dimiliki perpustakaan. Namun, pembangunan repositori tidak hanya sebatas kebutuhan fungsional saja, akan tetapi bagaimana repositori dapat bermanfaat tidak hanya bagi pengguna tetapi bermanfaat juga bagi universitas. Untuk pengguna, repositori bermanfaat sebagai sumber pengetahuan sementara bagi universitas, selain sebagai tempat pengelolaan artikel ilmiah, juga dapat bermanfaat agar universitas lebih dikenal oleh masyarakat luar.

Selanjutnya penelitian (Charles W. Bairley, Ir, 2006) Gerakan *open access* dalam beberapa tahun terakhir sangat meningkat seiring dengan perkembangan

ilmu pengetahuan dan teknologi yang memberikan kemudahan akses. Ketersediaan informasi ilmiah dalam bentuk artikel yang selama ini didominasi oleh publikasi komersial, mulai mendapat 'saingan' dengan gerakan ini. Salah satu yang memelopori gerakan *open access* adalah *Budapest Open Access Initiative* pada tahun 2002. Dengan menetapkan prinsip *open access* maka sebuah karya berarti tersedia secara bebas di internet, memberikan ijin bagi pengguna untuk membaca, mengunduh, menyalin, mendistribusikan, mencetak, mencari, atau menyediakan tautan ke *fulltext* artikel, melakukan pengindeksan, menggunakannya sebagai data untuk *software*, menggunakannya untuk tujuan yang legal, tanpa hambatan teknis, legal dan finansial untuk mengakses melalui internet.

Akan tetapi, pemilihan aplikasi tidak hanya untuk tujuan legal, tanpa hambatan teknis, legal dan finansial untuk mengakses melalui internet. Beberapa keuntungan ketika sebuah *institutional repository* menerapkan sistem *open access* untuk menyediakan sebuah sistem terintegrasi dan terkonsolidasi sehingga memudahkan akses. Serta dapat menjadikan *file software* dapat diakses oleh pengguna.

Dari beberapa penelitian yang dilakukan, disimpulkan bahwa pengembangan repositori selain untuk mempertahankan secara fungsional juga harus mempertahankan aspek-aspek *usability* dan *user experience* (UX) yang dirasakan oleh pengguna (Garrett, 2011). Selain itu, repositori juga memberikan manfaat lebih tidak hanya pada pengelolaan artikel tetapi juga pada pengindeksan *website*.

Adapun yang membedakan penelitian ini dengan penelitian sebelumnya (Djohan et al., 2010) terletak pada tahap penelitian. Pada tahap sebelumnya pada tahap penelitian menggunakan tiga aspek sudut pandang dan enam pertanyaan interogatif dari *Enterprise Architecture Zahman Framework*, sementara pada penelitian ini digunakan tahapan penelitian dengan metode *Waterfall (Classic Life Cycle)*.

Dari beberapa penelitian yang telah dilakukan, disimpulkan bahwa pemanfaatan sistem informasi berbasis komputer pada proses manajemen disuatu

organisasi sangat baik untuk diterapkan. Dengan hal tersebut pada penelitian ini, diterapkan *website Repository software* untuk mendukung proses pengunduhan software dengan cepat. Selain itu, solusi yang ditawarkan juga berbeda, pada penelitian ini menggunakan html dengan metode MVC dan ASP.NET.

2.2. Landasan Teori

2.2.1. Repositori

Repositori adalah sistem yang memungkinkan lembaga-lembaga untuk menyimpan dan mengelola dokumen digital mereka serta melakukan interaksi dan kolaborasi antara pengguna dalam satu lembaga. Terdapat beberapa perangkat lunak perpustakaan digital yang tersedia sebagai “*Open Source*” maupun sebagai “*Proprietary format*”. Perangkat lunak *open source* membantu perpustakaan terutama menurunkan biaya awal dan biaya berkelanjutan, dan memungkinkan *fleksibilitas* aplikasi yang lebih besar. Keuntungan utama dari perangkat lunak *Open source* adalah penggunaannya yang umumnya gratis seperti ASP.NET MVC.

2.2.2. Aplikasi Web

Aplikasi *web* merupakan aplikasi yang menggunakan teknologi browser untuk menjalankan aplikasi dan diakses melalui jaringan komputer (Remick,2011). Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemograman yang mendukung perangkat lunak berbasis web seperti HTML, JavaScript, Ruby, Python,Php, Java dan bahasa pemograman lainnya.

Aplikasi berbasis *web* dapat digunakan untuk berbagai macam tujuan yang berbeda. Sebagai contoh aplikasi berbasis *web* dapat digunakan untuk membuat *invoice* dan memberikan cara mudah untuk menyimpan data ke *database*. Aplikasi ini juga dapat dipergunakan untuk mengatur pengelolaan yang dapat bekerja memonitoring dalam hal tampilan.

2.2.3. Institutional Repository (IR)

Institutional Repository (IR) adalah ruang penyimpanan berbasis *online* untuk mengumpulkan, memelihara, dan menyebarluaskan informasi dalam bentuk digital dari keluaran intelektual sebuah institusi, khususnya lembaga penelitian. *Institutional Repository (IR)* menyediakan akses terbuka terhadap keluaran penelitian kelembagaan dengan pengarsipan pada repository, untuk menciptakan visibilitas global sebagai penelitian ilmiah sebuah institusi.

“Sebuah *Institutional Repository* berbasis universitas adalah serangkaian layanan yang ditawarkan universitas kepada anggota masyarakatnya untuk pengelolaan dan diseminasi materi digital yang dibuat oleh institusi dan anggota masyarakatnya. Hal ini pada intinya adalah komitmen organisasi terhadap penata layanan materi digital, termasuk pelestarian jangka panjang jika sesuai, serta organisasi dan akses atau distribusi. Pada suatu titik waktu tertentu, sebuah *Institutional Repository* akan didukung oleh seperangkat teknologi informasi, namun bagian penting dari layanan yang terdiri dari *Institutional Repository* adalah pengelolaan perubahan teknologi, dan migrasi konten digital dari satu set teknologi. (Sumber: Clifford A.Lynch, “*Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age*” ARL, no. 226 (Februari 2003): 1-7.).

Sebuah IR dapat berisi kumpulan dari berbagai macam perangkat lunak (*Software*) yang telah mendapatkan izin untuk disimpan kedalam repository. Oleh karena itu, sebuah IR tidak boleh berisi perangkat yang tidak mendapatkan izin atau aturan lisensi.

2.2.4. Visual Studio 2013

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi 18 *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi

(umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio 2013 dapat membuat dan mengedit aplikasi lebih mudah dan cepat sehingga meningkatkan produktifitas dalam pembuatan aplikasi. Microsoft Visual Studio 2013 sudah mendukung sistem operasi windows 8.1. Sehingga *software* ini sangat mendukung dan mudah bagi instansi yang ingin mengembangkan aplikasinya ke versi terbaru.

2.2.5. ASP.NET (MVC)

ASP.NET MVC adalah kerangka kerja untuk membangun aplikasi *web* yang menerapkan pola *Model-View-Controller* umum untuk kerangka ASP.NET (Galloway et al, 2014). Microsoft telah merilis Visual Studio 2013. Ada banyak fitur baru yang diperkenalkan dengan rilis ini dari Visual Studio. Salah satunya adalah MVC 6, itu adalah versi terbaru dari MVC. MVC adalah pola arsitektur yang memisahkan aplikasi menjadi tiga bagian utama yang disebut *Model*, *View* dan *Controller*. MVC adalah cara yang diakui untuk mengembangkan aplikasi.

MVC 5 adalah sebuah *update* terbaru dari Visual Studio dengan kelebihan dapat membangun sebuah *web* dinamis atau situs *data-driven*. Visual Studio telah menambahkan fitur menarik seperti *pages application*, *mobile optimization* dan *adaptive rendering*.

Menurut Joshi (2013) ada beberapa karakteristik penting dalam MVC 5 seperti yang diberikan di bawah ini:

1. One ASP.NET
2. ASP.NET Identity
3. Bootstrap
4. *Filter Authentication*
5. *Filter Authorization*

2.2.6. *Database first entity framework*

Pada pola arsitektur MVC pada ASP.NET secara umum terdapat tiga pendekatan yaitu *model first*, *code first* dan *database first*. Pada pendekatan *model first*, bagian pertama yang dibangun adalah model dalam bentuk diagram yang nantinya diubah menjadi model dalam bentuk kode secara otomatis. Pada pendekatan *code first* yang dibuat terlebih dahulu adalah *class – class model* kemudian tabel-tabel pada *database* akan secara otomatis dibuat saat pertama kali aplikasi web dijalankan. Di *database first* pendekatan ini umumnya dilakukan dimana *database* dan tabel-tabel di dalamnya telah terlebih dahulu dibuat. Kemudian dibuat *class model* berdasarkan tabel-tabel di dalam *database* tersebut. Pada penelitian ini sangat cocok menerapkan pendekatan *database first* karena struktur yang cukup kompleks.

2.2.7. **Microsoft SQL Server Management Studio**

Microsoft SQL Server adalah sistem manajemen *database* relasional yang dikembangkan oleh Microsoft. Sebagai *database*, Microsoft SQL Server adalah produk *software* yang fungsi utamanya adalah untuk menyimpan dan mengambil data seperti yang diminta oleh aplikasi perangkat lunak lain, baik *localhost* atau pada komputer yang sama maupun melalui jaringan (pada komputer lain dalam jaringan, termasuk jaringan *internet*). Banyak edisi yang berbeda dari Microsoft SQL Server ditujukan untuk pengguna yang berbeda dan untuk beban kerja yang berbeda pula (mulai dari aplikasi kecil yang menyimpan dan mengambil data di komputer yang sama ataupun untuk jutaan pengguna dan komputer yang mengakses data dalam jumlah besar dari *internet* pada waktu yang sama).

Microsoft SQL Server termasuk DBMS profesional. Beberapa pesaing seperti MySQL, Oracle, telah mengembangkan *software* serupa dalam beberapa tahun terakhir, tetapi Microsoft SQL Server lebih mudah digunakan dan memiliki lebih banyak fitur. Pemicunya antara lain adalah dukungan penuh dari Microsoft. Perangkat lunak yang ditawarkan oleh Microsoft juga menawarkan integrasi yang erat dengan .NET framework, dan ini tidak dimiliki oleh produk lain (Aiska, 2011).

2.2.8. Microsoft Visio

Menurut Helmers dalam bukunya yang berjudul “ Microsoft Visio 2013 Step by Step [2013:3] ”, Microsoft Visio adalah aplikasi utama untuk membuat semua diagram bisnis, mulai dari flowchart, network diagram, dan organization charts, untuk membuat denah dan brainstorming diagram. Microsoft 2013 melanjutkan kegunaan dari kebiasaan user interface, atau dikenal sebagai keterkaitan, hal itu telah diperkenalkan pada Visio 2010. Terlepas dari apa yang mungkin terpikirkan dari hubungannya dengan aplikasi Microsoft Office lainnya, dengan Visio rasanya seperti di rumah, terutama karena tujuan dari keterkaitan user interface gaya presentasi visual dari kelompok yang terkait fungsi, dan Visio termasuk didalamnya, pertama dan terutama, sebuah produk visual.

2.2.9. Balsamiq Mockups

Balsamiq Mockups adalah salah satu software yang berfungsi sebagai pembuatan desain. Software yang dapat mempermudah dalam menggambar sebuah tampilan user interface. Dengan menggunakan Balsamiq Mockups tidak perlu lagi menggambar komponen user interface. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

2.2.10. UML

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya (Nugroho et al, 2014). UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an (Nugroho et al, 2014). Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP (Nugroho et al, 2014). UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM (Nugroho et al, 2014).

Berikut ini adalah definisi dari 4 diagram UML menurut Handayani (2014) yaitu:

1. *Use caseDiagram*

Use caseDiagram merupakan inti fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara *actor* dan *system*.

2. *Sequence Diagram*

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

3. *Activity diagram*

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

2.2.11. Konsep System Development Life Cycle

Jogyanto (1991) menyatakan *System Development Life Cycle* merupakan metode yang digunakan untuk mengembangkan suatu sistem informasi, yaitu suatu proses standar yang diikuti untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan memelihara sistem informasi. Terlihat pada gambar 2.1.



Gambar 2.1 Proses SDLC

Menurut Moeta Alwan (2015), tahapan pengembangan *software* melalui proses SDLC (*Software Development Life Cycle*) mempunyai 6 tahap, *Requirement Analysis* atau Analisis Kebutuhan, *Design* atau Rancangan, Implementasi, *Testing* dan *Evolution* atau bisa diganti *Maintenance Program*.

1. Analisis Kebutuhan

Tahap analisis dilakukan untuk mengumpulkan dan menganalisa data kebutuhan informasi terkait sistem yang akan dibangun dengan melalui observasi lapangan atau wawancara. Tahapan ini dapat menjadi acuan dalam merancang *software* yang akan dikembangkan.

2. Merancang *Software*

Tahap *design* atau rancangan adalah menentukan cara kerja sistem dari menganalisa interaksi objek dan fungsi pada sistem, menganalisa data dan membuat skema *database* serta merancang *user interface*.

3. Implementasi *Software*

Tahap implementasi adalah mengimplementasikan rancangan dari tahap-tahap sebelumnya dengan melakukan pembuatan *database* sesuai skema rancangan dan pembuatan aplikasi berdasarkan desain sistem.

4. *Testing Software*

Sebelum *software* diterbitkan secara keseluruhan perlu dilakukan *testing* untuk memastikan bahwa *software* yang dikembangkan sesuai dengan kebutuhan yang sudah ditentukan.

5. Pemeliharaan *Software*

Tahap ini dilakukan untuk menjaga sistem tetap mampu beroperasi secara benar dan berkerja sebagai mana mestinya.

2.2.12. Black box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode ini, pada skripsi dipilih menggunakan *black box testing* karena anggapan lebih tepat dibandingkan *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam

aplikasi sehingga dalam hal ini *black box testing* lebih tepat dan sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.