

Perancangan Aplikasi *Source code library* Sebagai Solusi Pembelajaran Pengembangan Perangkat Lunak

Muhammad Wali¹, Lukman Ahmad²

¹ AMIK Indonesia

² AMIK Indonesia

article info

Article history:

Received 15 Oktober 2016

Received in revised form

2 Mei 2017

Accepted 27 Mei 2017

Available *online* 7 Juni 2017

Keywords:

Application, *Source code library*, *software development*

Kata Kunci:

Aplikasi, *Source code library*, pengembangan perangkat lunak

abstract

Source code library allows educators, Programmers or students and software developers to obtain a variety of reference codes on a programming language and software evaluation. Today, most Source code library for the purposes of learning software developers in the form of documentation of the use of a programming language that can be accessed through the official website developer programming languages, forum and various blogs. Because of the complexity of the features most web-based Source code library can only be accessed through the website and some others have provided documentation on each software vendor from the developers company the device. This research tries to construct a model of the application Source code library that can be used as a form of documentation for learning the use of various programming languages flexibly both in online and offline. The application allows the renewal of data/content Source code library at a time when the Internet is still available or at the time of the user's area does not have a network the Internet. In the implementation of this research will be divided in three stages, namely data collection pre development, development and implementation, and data collection of post-war development. Data collection pre development intended to get a preliminary study about the provision the core issue at hand, while the development and implementation phase focuses on model software design into diagrams and make the programming code to implement the design that has been created. While the data collection stage of the post-war development was for revamping the application made in conclusion, withdrawal, and suggestions for further research topics.

abstrak

Source code library memungkinkan pengajar, *programmer* maupun pelajar dan pengembang perangkat lunak untuk mendapatkan berbagai referensi kode-kode pada sebuah bahasa pemrograman perangkat lunak dan memberikan evaluasi. Saat ini, kebanyakan *Source code library* untuk keperluan pembelajaran pengembangan perangkat lunak berupa dokumentasi penggunaan suatu bahasa pemrograman yang dapat diakses melalui *website* resmi pengembang bahasa pemrograman, forum dan berbagai *blog*. Karena kompleksitas fiturnya kebanyakan web-based *Source code library* hanya dapat diakses melalui *website* dan sebagian lainnya telah disediakan dokumentasi pada setiap *software* vendor dari perusahaan pengembang perangkat. Penelitian ini mencoba untuk membangun model aplikasi *Source code library* yang dapat digunakan sebagai bentuk dokumentasi pembelajaran penggunaan berbagai bahasa pemrograman secara fleksibel baik dalam kondisi *online* maupun *offline*. Aplikasi tersebut memungkinkan pembaharuan data/konten *Source code library* pada saat *Internet* masih tersedia atau pada saat pengguna pada area tidak memiliki jaringan *Internet*. Dalam pelaksanaannya penelitian ini akan dibagi dalam tiga tahapan, yaitu pengumpulan data pra pengembangan, pengembangan serta implementasi, dan pengumpulan data pasca pengembangan. Pengumpulan data pra pengembangan dimaksudkan untuk mendapatkan bekal studi pendahuluan tentang inti masalah yang sedang dihadapi, sedangkan tahap pengembangan dan implementasi berfokus pada memodelkan perancangan perangkat lunak ke dalam diagram dan membuat kode pemrograman untuk mengimplementasikan perancangan yang telah dibuat. Sedangkan tahapan pengumpulan data pasca pengembangan adalah untuk membenahan aplikasi yang dibuat, menarik kesimpulan, dan saran untuk topik penelitian selanjutnya.

*Corresponding author. Email: muhammadwali@amikindonesia.ac.id, lukmanahmad@amikindonesia.ac.id

© E-ISSN: 2580-1643.

Copyright©2017. Published by Lembaga Informasi dan Riset (KITA INFO dan RISET), Lembaga KITA (<http://creativecommons.org/licenses/by/4.0/>).

1. Latar Belakang

Penerapan metode pembelajaran dalam lingkungan pendidikan berubah sedikit demi sedikit dengan adanya peran teknologi. Pembelajaran beralih dari metode tradisional yaitu kegiatan belajar mengajar di kelas, menjadi pembelajaran kolaboratif dan mandiri yang secara adaptif memanfaatkan teknologi *Internet, mobile*, dan jaringan tanpa kabel (Gitakarma dkk, 2011). Teknologi tersebut, melahirkan sebuah paradigma baru dalam pembelajaran yang memungkinkan seseorang untuk belajar dimana saja dan kapan saja yaitu *Ubiquitous Learning* (Yahya dkk, 2010).

Perangkat lunak kini sudah menjadi kekuatan yang menentukan. Perangkat lunak menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis; berfungsi sebagai dasar dari semua bentuk pelayanan serta penelitian keilmuan modern. Hal inilah yang mengubah pandangan masyarakat tentang perangkat lunak. Program-program perangkat lunak sudah tersebar luas, dan masyarakat memandangnya sebagai suatu kenyataan teknologi dalam kehidupan. Akan tetapi banyak individu dan perusahaan masih mengembangkan perangkat lunak secara sembarangan sehingga kualitas perangkat lunak yang dihasilkan menjadi kurang bagus, sehingga diperlukan metode-metode modern dari pendekatan rekayasa perangkat lunak untuk mencapai kematangan secara utuh (Wedhasmara, 2009).

Tujuan utama dari pengembangan perangkat lunak adalah untuk memberikan perangkat lunak berkualitas tinggi secara efisien dan dalam jumlah waktu sedikit. *Programmer* umumnya menggunakan kerangka kerja yang ada atau *libraries* untuk mengurangi upaya pengembangan perangkat lunak. Permasalahan yang ada adalah bahwa *Programmer* mengetahui jenis objek yang mereka butuhkan, tetapi tidak tahu bagaimana untuk mendapatkan objek secara spesifik dengan berbagai metode (Thummalapenta dan Xie, 2007).

Salah satu teknologi yang dapat digunakan sebagai perangkat bantu pembelajaran baik untuk

pengajar, *programmer* maupun pelajar adalah sebuah *source code library*. *Source code library* memungkinkan pengajar, *programmer* maupun pelajar dan pengembang perangkat lunak untuk mendapatkan berbagai referensi kode-kode pada sebuah bahasa pemrograman perangkat lunak dan memberikan evaluasi. Sedangkan di sisi *programmer*, *Source code library* memberikan kemudahan untuk dokumentasi, contoh program, perbaikan bug, deskripsi kesalahan, potongan kode, *template*, pola, atau saran (Goldman dan Miller, 2009).

Saat ini, kebanyakan *source code library* untuk keperluan pembelajaran pengembang perangkat lunak berupa dokumentasi penggunaan suatu bahasa pemrograman yang dapat diakses melalui *website* resmi pengembang bahasa pemrograman, forum dan berbagai *blog*. Karena kompleksitas fiturnya kebanyakan *web-based source code library* hanya dapat diakses melalui *website* dan sebagian lainnya telah disediakan dokumentasi pada setiap *software vendor* dari perusahaan pengembang perangkat lunak.

Di Indonesia, telah mendorong pergerakan dan perubahan perilaku kehidupan masyarakat dalam organisasi maupun perekonomian. Dilihat dari salah satu komponen utama komputer, pengembangan *software* sebenarnya sudah tersedia solusi alternatif yang menjanjikan bagi usaha implementasi teknologi yaitu dengan menggunakan *open source language*, yang dalam perkembangannya telah mempengaruhi banyak perubahan dalam perilaku para praktisi dan tenaga TI di dunia.

Pemanfaatan teknologi dalam pengembangan perangkat lunak memberi kesempatan kepada peneliti untuk membangun aplikasi dan mengetahui efektifitasnya dalam proses pembelajaran pengembangan perangkat lunak. Selain itu, dengan meningkatnya perkembangan teknologi khususnya pada pengembangan perangkat lunak, memicu penelitian mengenai layanan dan aplikasi untuk *source code library* dengan berbagai bahasa pemrograman.

Pengembangan *source code library* juga dilakukan oleh Goldman dan Miller (2009). Dalam hal ini, Goldman mengembangkan aplikasi *Codetrail* yang

memiliki berbagai fasilitas pendukung dokumentasi kode dan lain sebagainya. *Codetrail* menggunakan *Internet* sebagai konektivitas dan model data untuk menerapkan berbagai alat integrative. Namun, karena aplikasi yang dibangun berbasis web, sehingga waktu untuk mengakses tidak lebih cepat dibanding dengan aplikasi berbasis *deskstop* atau *native application*.

Selain itu, pengembangan aplikasi *source code library* juga dilakukan oleh Thummalapenta dan Xie (2007). Aplikasi *Source code library* dengan melakukan pendekatan *Code Search Engine* (CSE) untuk *programer*, Aplikasi ini membantu *Programmer* untuk mengatasi masalah pada kode, dan mengembangkan pendekatan yang mengambil permintaan dari bentuk "*source object type*" sebagai input, dan menyarankan urutan metode yang relevan yang dapat berfungsi sebagai solusi yang menghasilkan objek tujuan dari objek sumber diberikan dalam *query*. Aplikasi yang dikembangkan oleh Thummalapenta, S, et al (2007) merupakan aplikasi *source code library* mengumpulkan data relevan untuk contoh kode dan melakukan analisis statis.

McMillan, C, et al (2010) juga mengembangkan metode melalui sebuah pendekatan untuk merekomendasikan elemen perangkat lunak yang relevan untuk *Programmer*. McMillan, C, et al (2010) merekomendasikan contoh *source code* melalui *Application Programming Interface* (API). Pada penelitiannya, hanya pada pengembangan bahasa Java yang dibuat tidak dilengkapi perpustakaan kode dan hanya menghasilkan informasi kode yang relevan bagi *Programmer* Java.

OverZone *Software* (2016) mengembangkan *Source code library* dengan melibatkan lebih dari 30 bahasa pemrograman dalam aplikasinya. Aplikasi ini membantu para *programer* untuk mendapatkan informasi kode dengan lengkap. Pada aplikasi yang dibuat bersifat *Shareware*, aplikasi tersebut memiliki batas waktunya. *Source code library* yang dikembangkan OverZone *Software* memiliki waktu pakai trial 30 hari dan masih dalam ujicoba.

Berdasarkan dari berbagai penelitian yang telah dilakukan maka penelitian ini mencoba untuk membangun model aplikasi *source code library* yang dapat digunakan sebagai bentuk dokumentasi

pembelajaran penggunaan berbagai bahasa pemrograman secara fleksibel baik dalam kondisi *online* maupun *offline*. Aplikasi tersebut memungkinkan pembaharuan data/konten *Source code library* pada saat *Internet* masih tersedia atau pada saat pengguna pada area tidak memiliki jaringan *Internet*.

2. Metode Penelitian

Secara garis besar penelitian ini dibagi dalam tiga tahapan, yaitu pengumpulan data pra pengembangan, pengembangan serta implementasi, dan pengumpulan data pasca pengembangan. Pengumpulan data pra pengembangan dimaksudkan untuk mendapatkan bekal studi pendahuluan tentang inti masalah yang sedang dihadapi, sedangkan tahap pengembangan dan implementasi berfokus pada memodelkan perancangan perangkat lunak ke dalam diagram dan membuat kode pemrograman untuk mengimplementasikan perancangan yang telah dibuat. Sedangkan tahapan pengumpulan data pasca pengembangan adalah untuk membenahan aplikasi yang dibuat, penarikan kesimpulan, dan saran untuk topik penelitian selanjutnya.

Model penelitian yang digunakan adalah penelitian kualitatif, artinya data yang dikumpulkan dan digunakan untuk penelitian bukan merupakan data statistik. Dalam penelitian kualitatif, data yang diolah adalah berupa kata-kata yang dikumpulkan dengan metode wawancara atau diskusi kelompok. Metode ini banyak digunakan untuk pengembangan perangkat lunak karena dapat menghasilkan data kebutuhan sistem yang lebih deskriptif dan mudah diimplementasikan ke dalam perangkat lunak.

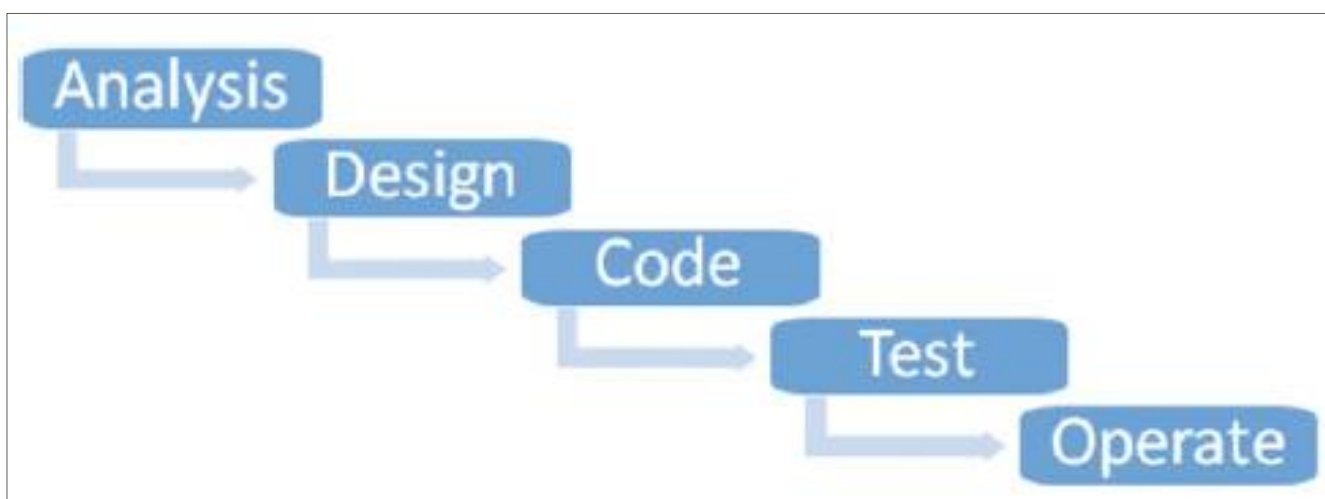
Data yang dikumpulkan dalam penelitian kualitatif adalah data tekstual berupa kata dan kalimat. Cara yang paling banyak digunakan untuk pengumpulan data kualitatif dalam penelitian ini adalah dengan melalui studi literatur dan diskusi kelompok. Pada penelitian ini pengumpulan data awal yang digunakan untuk mendapatkan daftar kebutuhan sistem adalah dengan melakukan studi literatur tentang *source code library*, Bahasa Pemrograman, dan Sistem Operasi Windows. Selanjutnya data studi literatur dibuat dalam bentuk daftar untuk dianalisis antara standar *source code library* yang ingin dicapai dan *source code library* yang sudah ada saat ini. Selanjutnya

dilakukan juga studi dokumentasi *source code library* dan analisis pola pengembangan komponen tambahan (*plugin*) yang dapat diakomodasi oleh *source code library*. Ini diperlukan untuk menentukan strategi komunikasi antara *source code library* dan pihak pengembang perangkat lunak yang akan dikembangkan pada sistem operasi *windows*.

Sedangkan pengumpulan data akhir setelah proses implementasi akan dilakukan dengan cara membuat checklist fitur yang direncanakan dan yang telah dipenuhi, serta uji performa untuk mendapatkan data kebutuhan perangkat keras implementasi sistem. Kemudian purwarupa diuji cobakan kepada mahasiswa untuk mendapatkan *feedback* dalam

sebuah perusahaan lebih dari sekedar menulis kode. Kolaborasi dengan tim lain dan anggota tim dengan berbagai ilmu khusus, merupakan langkah-langkah tambahan dalam mendokumentasikan proses pengembangan perangkat lunak. (Crookshanks, 2015).

Model Sekuensial Linier sering disebut Model Air Terjun (*Waterfall*) merupakan paradigma rekayasa perangkat lunak yang paling tua dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekunsial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.



Gambar 1. Tahapan dalam Model Air Terjun (Waterfall)

bentuk kuisisioner dan juga wawancara langsung. Hasil dari kuisisioner akan diinputkan ke dalam tabulasi dan dilakukan penghitungan sederhana untuk mendapatkan data kuantitatif keberhasilan sistem. Sedangkan data hasil wawancara akan didokumentasikan dan dikategorisasikan untuk membedakan evaluasi untuk penyempurnaan program yang dapat dilakukan pada penelitian ini, atau akan dijadikan masukan pada penelitian berikutnya.

Sedangkan dalam pengembangan perangkat lunak menggunakan model sekuensial linier sering disebut model air terjun (*Waterfall*). Pengembangan perangkat lunak merupakan industri terbaru, dalam sebuah industri, perkembangan dan pertumbuhan benar-benar diperlukan untuk menjadi lebih baik (Kort, 2016). Pengembangan perangkat lunak pada

Model Sekunsial Linier mengikuti aktivitas-aktivitas yaitu:

- a. Rekayasa dan Pemodelan Sistem/Informasi
 Karena perangkat lunak merupakan bagian dari suatu sistem maka langkah pertama dimulai dengan membangun syarat semua elemen sistem dan mengalokasikan ke perangkat lunak dengan memerhatikan hubungannya dengan manusia, perangkat keras dan database.
- b. Analisis Kebutuhan Perangkat Lunak
 Proses menganalisis dan pengumpulan kebutuhan sistem yang sesuai dengan domain informasi tingkah laku, unjuk kerja, dan antar muka (*interface*) yang diperlukan. Kebutuhan-kebutuhan tersebut didokumentasikan dan dilihat lagi dengan pengguna.

c. Desain

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada; struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural.

d. Pengkodeaan (*Coding*)

Pengkodean merupakan prses menerjemahkan desain ke dalam suatu bahasa yang bisa dimengerti oleh komputer.

e. Pengujian

Proses pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah diuji. Pengujian eksternal fungsional untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

f. Operasi

Merupakan bagian dari instalasi, migrasi, dukungan, dan pemeliharaan sistem yang lengkap (Kort, 2016).

3. Pembahasan

Berdasarkan metode penelitian, terdapat beberapa Analisa dari pengembangan *source code library*, seperti terlihat pada tabel 1, 2 dan 3 berikut:

Tabel 1. Daftar Masalah pada *Current System*

No	Permasalahan <i>Current System</i>
1	SCL yang ada susah diakses dengan menggunakan koneksi internet yang lambat
2	Web SCL hanya bisa diakses secara <i>online</i> , jika internet tidak memiliki koneksi yang bagus maka <i>database</i> tidak bisa diakses lagi
3	SCL kebanyakan hanya dapat digunakan secara <i>online</i>
4	SCL dapat men-download materi, namun untuk sinkronisasi materi, dll masih harus diakses <i>online</i>
5	SCL belum bisa mengerjakan <i>updating</i> secara <i>offline</i>

Tabel 2. Daftar Kebutuhan Fungsional Sistem

No	Permasalahan <i>Current System</i>
1	SCL dapat diinstall pada system operasi <i>windows</i>
2	SCL terkoneksi dengan salah satu web aplikasi SCL
3	SCL dapat menyimpan data <i>online</i> di <i>dektop</i>

4	SCL diakses <i>student</i> dengan <i>user</i> yang sama dengan <i>user</i> pada web aplikasi SCL
5	SCL dapat melakukan sinkronisasi data secara aktif sesuai dengan keinginan user
6	SCL dapat digunakan untuk menjawab materi dari user secara <i>offline</i>
7	SCL dapat digunakan untuk mengakses properti bahasa pemrograman secara <i>offline</i>
8	SCL hanya memerlukan akses <i>internet</i> untuk sinkronisasi dengan <i>server</i> saja

Berdasarkan tabel 1 daftar masalah pada *Current System* dan Daftar Kebutuhan Fungsional Sistem maka peneliti membuat sebuah perbandingan SCL yang ada pada saat ini seperti terlihat pada tabel 3 berikut:

Tabel 3. Perbandingan SCL Saat ini

	SCL Overzone Software	CodeLocker Source Code Library	Sublime Text	Visual Studio Code	Notepad++
General	Release Date: January 18, 2012 Platform: Overzone Software	Created: September 25, 2003 Concept: Encrypted Technology	July 01, 2013 Platform: Text	April 14, 2015 Microsoft	August 14, 2011 Din BO
Download Information	Version: 1.1.1.31 File Size: 14.43MB	Version: 1.1040 File Size: 1.1MB	Version: 1.1.1.31 File Size: 1.1MB	Version: 1.1.1.31 File Size: 1.1MB	Version: 1.1.1.31 File Size: 1.1MB
Pricing	License Model: Free to try Duration: 30-day trial	License Model: Free to try Duration: 5-day trial	License Model: Free to try Duration: Not available	License Model: Free to try Duration: Not available	License Model: Free to try Duration: Not available
Operating System	OS: Windows 98, XP, Vista, 2000, XP, 2003	OS: Windows 98, XP, Mac OS, XP, 2000, XP	OS: Windows 98, XP, Vista, 2000, XP, 2003	OS: Windows 98, XP, Vista, 2000, XP, 2003	OS: Windows 98, XP, Vista, 2000, XP, 2003
Additional Requirements	None	Windows XP SP3, Mac OS X 10.5.7, Java 1.6.0_20	None	None	None
Category	Business Software	Developer Tools	Productivity Software	Developer Tools	Developer Tools
Category	Education	Source Code Tools	Text Editing Software	Coding Utilities	Coding Utilities
Tasks	<ul style="list-style-type: none"> • Built as library with 50,000+ lines of code • Support for more than 15 programming languages • Manage all of your source code in a single source place. • Optional password protection with super-strong 4096 bit encryption. • Full integration with all major programming IDEs. • Built as AutoInstall utility • Quality search for code using a simple toolbar • The sub-structured and nested a free trial of the software before purchasing • Download available for: Windows 2000, XP, 2003, Vista (32 and 64 bit), 7 (32 and 64 bit) • Get all this at a discounted price! 	<ul style="list-style-type: none"> • CodeLocker Copy-paste source • Ability to license productivity by cloning and restoring only objects • Don't need to be compiled 	<ul style="list-style-type: none"> • Code Aesthetics • Command palette • Multiple selections • Plugin-based slight API • Project-specific preferences • Extensive extensibility via 100+ settings files, including project specific settings • Cross platform (Windows, OS X, Linux) • Compatible with many language processes from JetBrains. 	<ul style="list-style-type: none"> • Keyboard shortcuts • 130 Code Emmetons • Marketplace • IntelliSense • Code Navigation • Debugging • Version Control • Integrated Terminal • Tasks • Creating new code snippets • Emmet • Command Line • Accessibility • 12 Languages • Multi / language • Emmetons editing • Extensibility reference • OTHER: NET Core, Java, and Office 	<ul style="list-style-type: none"> • Syntax Highlighting and Folding • User Defined Syntax Highlighting and Folding • PORE: Part Compatible Syntax Expression Search/Replace • GUI extensibility windows with close buttons, multi-line tabs, vertical tab and vertical document list • Document Map • Auto-completion: Word completion, function completion and Function parameter list • Document Tab (syntax) • Multi-View • GYFWTU (Printing) • Zoom in and zoom out • Multi-Language environment supported • DockPanel • Macro recording and playback • Launch with different arguments
Concepts	Model	Source Code Tools	Editor Tools	Editor Tools	Editor Tools

Berdasarkan tabel 3 yang menjadi perbandingan SCL yang ada saat ini produk software yang dibandingkan meliputi SCL Overzone Software, CodeLocker Source Code Library, Sublime Text, Visual Studio Code, dan Notepad++. Maka peneliti membagi

beberapa software diatas yaitu; SCL Overzone Software dan CodeLocker Source Code Library merupakan sebuah Source Code Tools. Sedangkan Sublime Text, Visual Studio Code, dan Notepad++ merupakan sebuah Editor Tools.

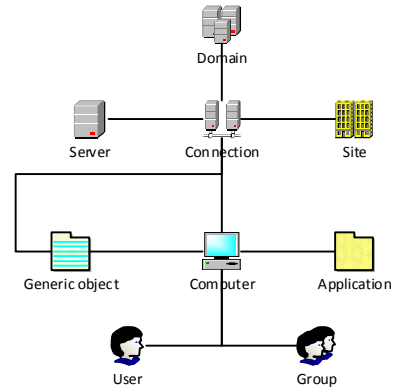
Berdasarkan tabel 3, maka peneliti juga membuat daftar perbandingan *framework* untuk pengembangan SCL berbasis *desktop* seperti ditunjukkan pada tabel 4 berikut:

Tabre 4. Perbandingan *Framework* Pengembangan Perangkat Lunak *Dekstop*

	Programming language	Emulator available	Integrated development environment available	Cross-platform deployment	Installer packaging options	Development tool cost
NET Compact Framework	C#, VB.NET, Basic4ppc	Free emulator, source code available, also bundled with IDE	Visual Studio 2016, 2015, 2013, 2010, 2008, 2005, 2003, Basic4ppc IDE	Windows Mobile, Windows CE, Symbian-based devices via third party tools	OTA deployment, CAB files, ActiveSync	Most tools free, but commercial editions of Visual Studio needed for visual designers
NET/XAML	C#, C++, F#, VB	Free emulator, source code available, also bundled with IDE	portable class libraries	Windows Store, Windows Phone, Windows desktop applications, and other Microsoft platforms	NET Framework code and libraries	Most tools free, but commercial editions of Visual Studio needed for visual designers
C++ and Win32	C++ with COM	When writing a desktop application in C++, you can choose MFC or Win32 for the UI, or a host of third-party application frameworks that also support non-Windows platforms	C++ provides a first-class development experience across a wide range of platforms, both Windows and non-Windows, without depending on a high-level runtime environment like NET	You can share C++ code between Windows, Windows Phone, and Windows Store, as well as with non-Windows platforms.	NET Framework code and libraries	Most tools free, but commercial editions of Visual Studio needed for visual designers or both Windows and non-Windows

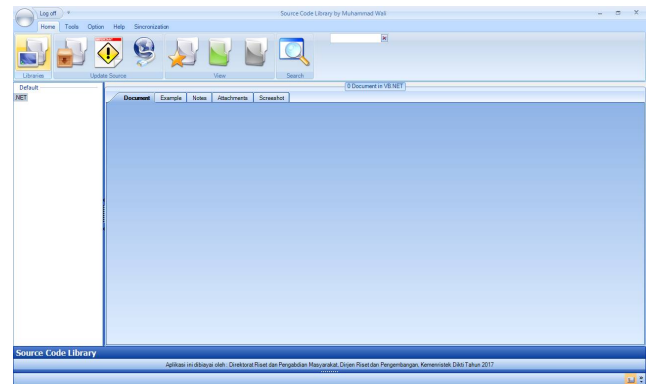
Tujuan melakukan perbandingan untuk mencari alternatif solusi untuk permasalahan yang sudah ada. Metode yang digunakan untuk mencari alternatif solusi adalah penelitian kualitatif dengan melakukan studi literatur tentang pengembangan aplikasi dan modifikasi *source code library*. Hasil dari studi literatur kemudian akan dikelompokkan menjadi tiga bagian yaitu hasil analisis kebutuhan sistem, saran desain permodelan sistem, dan teknologi relevan yang akan digunakan.

Adapun perancangan prototipe *source code library* ini seperti yang terlihat pada *active directory object application* sistem pada Gambar 2.



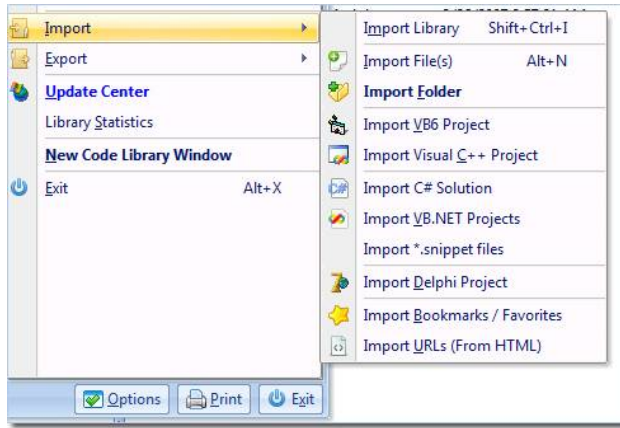
Gambar 2. *Active Directory Object Application*

Dari penelitian yang telah dilakukan, dihasilkan suatu sistem pembelajaran baru mengenai *source code library* dalam pembelajaran pengembangan perangkat lunak. Pada aplikasi ini menampilkan bahasa pemrograman, konten dan sinkronisasi materi.



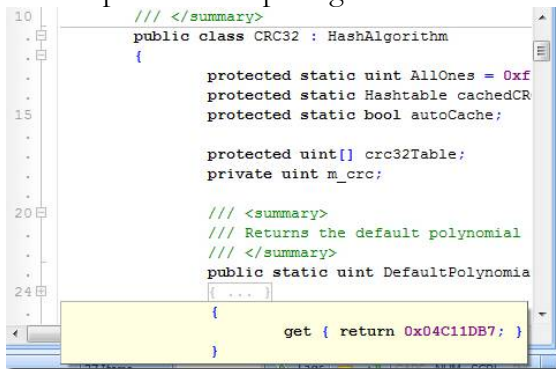
Gambar 3. Halaman Awal

Pada halaman awal menampilkan beberapa *tools* seperti *libraries*, *update source*, *view*, dan *search*. Pada menu *libraries* untuk menampilkan berbagai bahasa pemrograman yaitu; VB.NET, C#, C++, ASPX, HTML, CSS, PHP, JQuery, SQL dan JSON. Untuk penambahan bahasa pemrograman dapat menggunakan *tools imports* seperti terlihat pada gambar 4.

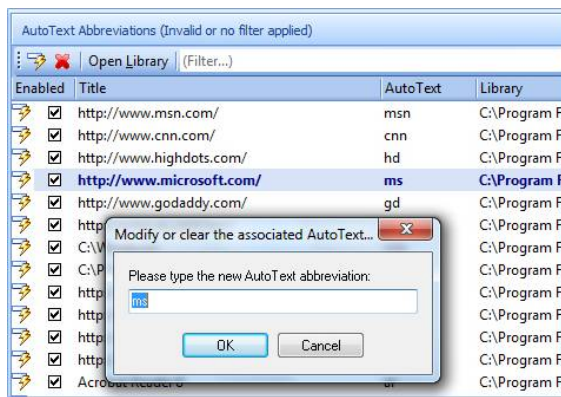


Gambar 4. Halaman Impor Bahasa Pemrograman

Dengan memilih bahasa pemrograman yang akan dipelajari maka tampilan pembelajaran bersifat konten seperti terlihat pada gambar 5.

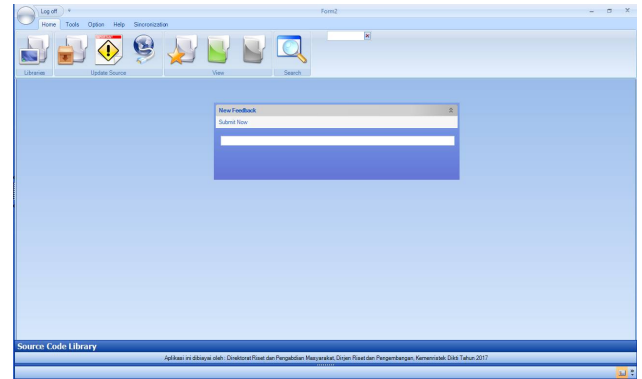


Gambar 5. Halaman Konten

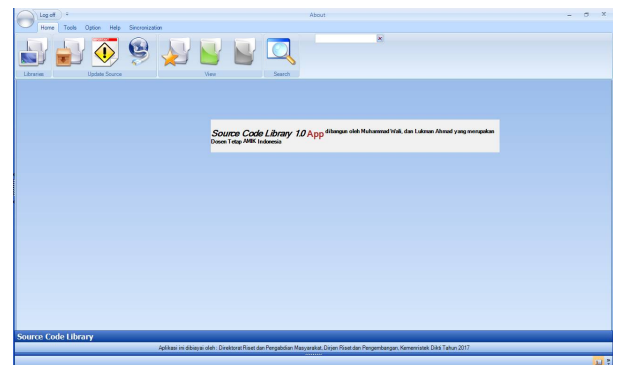


Gambar 6. Halaman Sinkronisasi

Selain dengan menggunakan fasilitas *import*, penambahan koleksi bahasa pemrograman dapat dilakukan dengan sinkronisasi seperti terlihat pada gambar 6. Tujuan sinkronisasi sebagai implementasi bahwa konten dapat ditambah dan dijalankan pada kondisi *online* dan *offline*.



Gambar 7. Halaman *Feedback*



Gambar 8. Halaman Info

Halaman *feedback* dan *info* merupakan umpan balik dan informasi dari aplikasi SCL dari pengguna.

4. Kesimpulan dan Saran

Dari hasil penelitian maka dapat disimpulkan bahwa:

- Aplikasi *dektop* memiliki keterbatasan dalam pengaksesan perangkat lunak *Source code library*, salah satunya adalah keterbatasan koneksi. Untuk mengatasi keterbatasan koneksi tersebut, basis data lokal dapat dimanfaatkan untuk menyimpan data *Source code library online* ke dalam perangkat penyimpanan lokal, sehingga pada saat aplikasi *dektop* tidak dapat terhubung dengan *Internet* perangkat lunak *Source code library* masih dapat digunakan dengan mengakses data lokal.
- Berdasarkan hasil perbandingan yang dilakukan, *framework* yang digunakan untuk mengembangkan perangkat lunak untuk *platform* yang spesifik seperti *Windows* mempunyai performa yang lebih baik dibandingkan *framework* yang dapat digunakan untuk mengembangkan berbagai macam perangkat. Meski demikian, *.NET framework* yang dipilih pada penelitian ini dapat digunakan untuk mengembangkan aplikasi multi platform tanpa

terlalu banyak kehilangan performa aplikasi yang dikembangkan.

- c. *Source code library* yang dibangun oleh Overzone *Software* memiliki konfigurasi yang cukup rumit untuk aplikasi dan modul *service*-nya, Konfigurasi ini dapat menyulitkan pada lingkungan sistem yang belum dikenal dengan baik seperti pada windows yang tidak memberikan spesifikasi teknis secara komplit. Kesalahan konfigurasi seperti pada .NET Framwework memiliki resiko *Source code library* tidak dapat digunakan sama sekali.

Sedangkan saran untuk penelitian selanjutnya adalah :

- a. *Source code library* Overzone *Software* memiliki API dan web service yang tidak cukup lengkap. Kurangnya modul ini tidak memberikan kesempatan bagi pengembang atau pihak ketiga dan peneliti untuk mengintegrasikan *Source code library* dengan sistem yang lain.
- b. Sinkronisasi data besar pada aplikasi *dektop* memiliki resiko kegagalan yang lebih besar. Pada penelitian ini masih menggunakan *single sinchronization process*, yang berarti jika di tengah jalan proses sinkronisasi terdapat gangguan, maka proses sinkronisasi harus dimulai kembali dari awal. Untuk mengatasi hal tersebut, perlu dikembangkan multi stage synchronization yang memungkinkan aplikasi melakukan sinkronisasi dalam beberapa tahap.

5. Ucapan Terima Kasih

Penelitian ini didukung oleh Kementrian Riset dan Pendidikan Tinggi (Ristekdikti) dalam bentuk dana hibah skim penelitian dosen pemula (PDP). Kami berterima kasih kepada teknis dan anggota yang terlibat secara tidak langsung dalam penelitian ini. Terima kasih juga tidak lupa juga kepada LPPM AMIK Indonesia yang telah memberikan dukungan penelitian dalam bentuk penyediaan sarana laboratorium dan konsultasi teknis.

6. Daftar Pustaka

Bansal, K. A. 2014. Introduction to Programming Languages, CRC Press Taylor & Francis Group. United States of America.

Crookshanks, E. 2015. Practical Enterprise *Software*

Development Techniques, Apress Publisher. Groningen. The Netherlands.

Gitakarma, M. S. dan Tjahyani, L. P. A. S. 2011. Modifikasi Claroline dengan Metode Pembelajaran Computer-Supported Collaborative Learning (CSCL) Berbasis Konstruktivisme, Prosiding Seminar Nasional Pendidikan Teknik Informatika, Singaraja: 20 September 2011. 1-15.

Goel, M. 2016. Editorial review: *Source code library*. www.source-code-library.software.informer.com. 10 Mei 2016 (21:20).

Goldman, M and Miller, R. 2009. Codetrail: Connecting Source Code and Web Resources. Journal of Visual Languages & Computing. 20(4): 223-235.

Kort, D. W. 2016. DevOps on the Microsoft Stack, Ordina Microsoft Solutions Apress Publisher. Groningen. The Netherlands.

McMillan, C et al. 2010. Recommending Source Code Examples via API Call Usages and Documentation, Proceedings of the 2nd International Workshop on Recommendation Systems for *Software* Engineering. Association for Computing Machinery. 2-8 Mei 2010. 21-25.

Noviansyah, E. 2008. Aplikasi *Website* Museum Nasional menggunakan Macromedia Dreamweaver MX. STIK. Jakarta.

OverZone, S. 2016. *Source code library* Documentation. www.highdots.com. 10 Mei 2016 (09:00).

Sebesta, W, R. 2012. Concepts of Programming Languages. 10nd ed. Pearson Education, Inc., Publishing as Addison-Wesley, United States of America.

Thummalapenta, S and Xie, T. 2007. PARSEWeb: A *Programmer* Assistant for Reusing Open Source Code on the Web, Proceedings of the twenty-second IEEE/ACM international conference on Automated *software* engineering. Association for Computing Machinery. 4-9 Novembr 2007. 204-213.

Wedhasmara, A. 2009. Memasyarakatkan dan Memperoleh Concern Akan *Software* Quality, Sebagai

Faktor Pendorong Penerapan CMMI atau CMM-SW, Jurnal Sistem Informasi (JSI). 1(3): 100-112.

Yahya, S., Ahmad E. A. dan Jalil, K. A., 2010, The Definition and Characteristics of Ubiquitous Learning: A Discussion, Int. Journal of Education and Development using Information and Communication Technology. 1(6): 117-127.