

Daur Hidup Pengembangan Sistem untuk Otomasi Perpustakaan

Athanasia Octaviani Puspita Dewi^{1*)}

*Program Studi S-1 Ilmu Perpustakaan, Fakultas Ilmu Budaya, Universitas Diponegoro,
Jl. Prof. Soedharto, SH, Kampus Undip Tembalang, Semarang, Indonesia*

*) Korespondensi: athanasia.dewi.9@gmail.com

Abstract

[System Development Lifecycle of Library Automation] This article describes how to develop library automation system using system development life cycle according to Sommerville which consists of Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing, Operation and Maintenance. This is necessary for librarians to know because most librarians are just adept at using software but have not been so sensitive to software development itself. Hopefully if the librarian understands this, the automated system will run maximally.

Keywords: : *library automation; software; system development life cycle*

Abstrak

Artikel ini menjelaskan cara mengembangkan sistem otomasi perpustakaan menggunakan siklus hidup pengembangan sistem sesuai dengan Sommerville yang terdiri dari Analisis dan Definisi, Desain Sistem dan Perangkat Lunak, Implementasi dan Pengujian Unit, Integrasi dan Pengujian Sistem, Operasi dan Pemeliharaan. Hal ini diperlukan bagi pustakawan untuk mengetahui bagaimana cara mengembangkan sistem informasi karena sebagian besar pustakawan hanya mahir menggunakan perangkat lunak tetapi belum begitu sensitif terhadap pengembangan perangkat lunak itu sendiri. Harapannya adalah jika pustakawan memahami ini, sistem otomatis akan berjalan maksimal.

Kata kunci: otomasi perpustakaan; perangkat lunak; siklus hidup pengembangan sistem

1. Pendahuluan

Saat ini sistem otomasi sudah banyak digunakan di banyak perpustakaan. Bisa dikatakan bahwa perpustakaan yang tidak menggunakan sistem otomasi adalah perpustakaan yang tertinggal. Sistem ini dibuat untuk memudahkan kinerja pustakawan. Dalam penerapannya di perpustakaan, para pustakawan terbiasa hanya menggunakan *software* sistem otomasi dan jarang memperhatikan adanya *updating software*. Padahal pemeliharaan dan pengembangan *software* otomasi penting dilakukan supaya masa hidup *software* lebih lama dan selalu bisa diandalkan.

Dari penjelasan tersebut maka perlu dikaji bagaimana melakukan pengembangan sistem otomasi perpustakaan dengan memperhatikan siklus hidup pengembangan sistem. Tujuan penelitian ini adalah untuk membuka wawasan kepada pustakawan bahwa pengembangan sistem perlu dilakukan demi kelangsungan hidup dari *software* otomasi perpustakaan. Manfaat dari penelitian ini adalah untuk menambah pengetahuan pustakawan dalam mengembangkan sistem otomasi secara maksimal. Metode penelitian yang digunakan adalah studi literatur, dengan menganalisa tulisan-tulisan dari beberapa sumber.

2. Landasan Teori

2.1 Otomasi Perpustakaan

Lasa Hs (2005) menyatakan bahwa otomasi perpustakaan adalah sistem informasi yang di mencakup kegiatan pengumpulan, pengolahan, pengawetan, pelestarian dan penyajian serta penyebaran informasi. Sedangkan Sulisty-Basuki, berpendapat bahwa otomasi perpustakaan merupakan implementasi teknologi informasi untuk kepentingan perpustakaan yang mencakup pengadaan, sampai dengan penyediaan jasa untuk para pemustaka yang datang ke perpustakaan. Dari pendapat dua ahli tersebut, jelas bahwa otomasi perpustakaan berkaitan dengan pengelolaan perpustakaan dengan memanfaatkan teknologi informasi. Biasanya otomasi perpustakaan berupa *software* pengelola perpustakaan seperti Senayan, INLIS Lite, dll.

Tujuan otomasi perpustakaan menurut Saiful (2007: 6) adalah sebagai berikut:

1. Mempermudah integrasi berbagai kegiatan yang ada di perpustakaan.
2. Mempermudah kerjasama dan pembentukan jaringan di perpustakaan.
3. Membantu menghindari duplikasi kegiatan yang terdapat di perpustakaan.
4. Menghindari pekerjaan yang bersifat mengulang dan membosankan.
5. Memperluas jasa perpustakaan.
6. Memberikan peluang untuk mempromosikan jasa perpustakaan dengan mudah.
7. Meningkatkan efisiensi waktu dan pekerjaan pustakawan.

Dari tujuan diciptakannya otomasi perpustakaan tersebut dapat disimpulkan bahwa, otomasi perpustakaan diciptakan untuk mengefisiensikan pekerjaan pustakawan, supaya kegiatan-kegiatan yang terdapat di perpustakaan tidak terduplikasi, hal ini bertujuan untuk menyajikan pelayanan yang maksimal untuk pemustakanya.

Menurut Supriyanto dan Muhsin (2008 : 38-42) dalam sebuah sistem otomasi perpustakaan terdapat beberapa unsur yang saling berkaitan. Unsur-unsur tersebut sebagai berikut:

1. Pengguna (*User*)

Pengguna merupakan unsur utama dalam sebuah sistem otomasi perpustakaan. Pengguna merupakan seseorang yang menggunakan dan memanfaatkan dalam menelusur untuk mendapatkan informasi yang dibutuhkan.

2. Perangkat Keras (*Hardware*)

Perangkat keras adalah peralatan yang dapat disentuh dan dapat menerima inputan berupa data, kemudian diolah dan nantinya akan berubah menjadi sebuah informasi.

3. Perangkat Lunak (*Software*)

Perangkat lunak berupa perintah-perintah yang bertugas memberitahu perangkat keras untuk melakukan tugas sesuai dengan inputan dari pengguna. Di dalam perintah-perintah tersebut terdapat metode atau prosedur yang berfungsi mengoperasikan perangkat keras agar sesuai dengan

permintaan pengguna. yang memberitahu perangkat keras untuk melakukan suatu tugas sesuai perintah.

4. Jaringan (*Network*)

Tujuan utama adanya jaringan di dalam otomasi perpustakaan adalah untuk memudahkan pengguna dalam melakukan pertukaran data.

5. Data

Dalam perpustakaan data dapat berupa angka, huruf, suara, gambar, dan simbol-simbol. Dalam perpustakaan data ini dapat berupa identitas sebuah buku atau bibliografi, data anggota, data pustakawan, dan lain sebagainya.

6. Manual atau Panduan Operasional

Prosedur adalah penjelasan bagaimana meng*install*, menyesuaikan, menjalankan suatu perangkat keras atau perangkat lunak.

Menurut Kadir (2003: 423) mengadakan sistem informasi dapat dilakukan dengan berbagai cara, antara lain:

1. Membuat Sendiri

Sebuah organisasi yang besar biasanya memenuhi kebutuhan sistem informasi dengan cara membuat sendiri. Dengan begitu maka hasil dari sistem informasi ini akan sesuai dengan kebutuhan pengguna sistem. Akan tetapi sistem informasi dengan cara membuat sendiri biasanya akan memakan waktu yang lama karena pembangunan sistem akan dimulai dari nol. Kesuksesan pengembangan sistem informasi dengan cara membuat sendiri terletak pada kemampuan karyawan teknologi informasi dalam instansi tersebut dan juga dukungan kesiapan dari para calon pemakai sistem yang terlibat selama pengembangan sistem ini berlangsung.

2. Membeli Perangkat Lunak Paket

Membeli perangkat lunak sama dalam artian di sini adalah membeli program siap pakai. Biasanya pembelian secara paket seperti ini tidak memungkinkan perusahaan pembeli untuk melakukan modifikasi sendiri sistem yang sudah terpasang, sehingga pemeliharaan akan bergantung kepada vendor. Akan tetapi kadang pengembang dari internal instansi dapat mengatur beberapa variabel tertentu yang terdapat pada paket. Memilih membeli perangkat lunak yang sudah terpaket akan menghemat waktu pengembangan sistem informasi, dan biasanya paket sudah dilakukan pengujian atau *testing* terhadap *bug*.

3. Melakukan *Outsourcing*

Outsourcing adalah menyerahkan pengembangan sistem informasi kepada pihak ketiga. Akan tetapi biasanya pihak ketiga juga sekaligus menangani operasi sistem dan bahkan terlibat dalam penyediaan perangkat kerasnya. Cara mengembangkan sistem seperti ini menguntungkan karena biasanya pihak ketiga yang bekerja sama adalah mereka yang memiliki banyak pengalaman pada sistem yang sama sehingga waktu atau durasi untuk pengembangan sistem menjadi lebih pendek.

Selain itu, biaya untuk melakukan pengembangan sistem akan lebih murah karena banyak perusahaan kecil dan bersifat lokal yang bergerak dalam bidang pengembangan sistem informasi. Akan tetapi juga terdapat beberapa hal yang perlu dipertimbangkan jika memilih pengembangan sistem dengan cara *outsourcing*, seperti data yang dimiliki oleh perusahaan akan diketahui pihak luar instansi, dan karena sistem tidak dibangun sendiri maka akan menjadi tergantung kepada pihak ketiga jika sewaktu-waktu ada *upgrade* sistem informasi.

2.2 Siklus Hidup Pengembangan Sistem

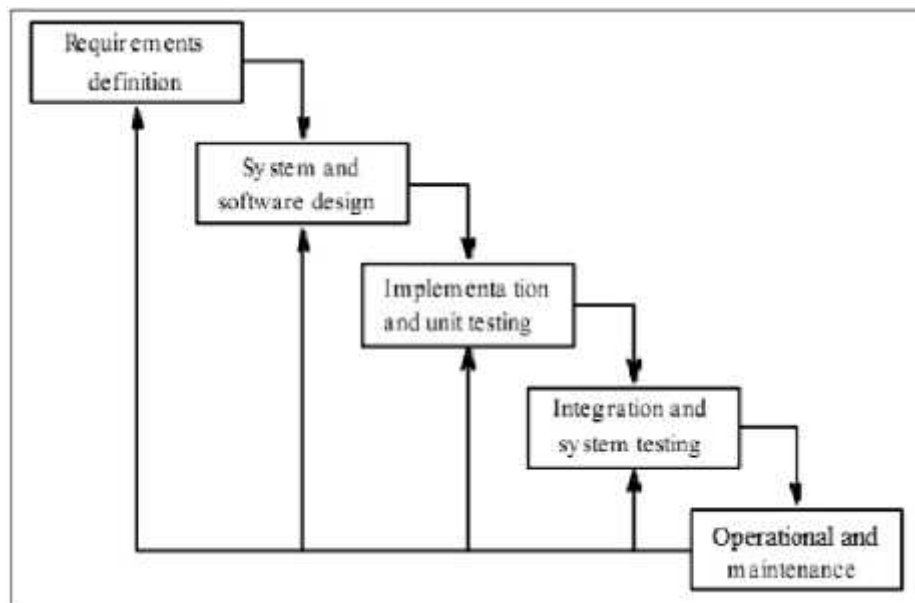
Untuk mengembangkan sistem informasi pada sebuah instansi, maka digunakanlah metodologi pengembangan sistem. Hoffer dkk dalam Kadir (2003: 398) menjelaskan bahwa yang dimaksud dengan metodologi di sini adalah proses standar yang diikuti oleh organisasi atau instansi untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa kebutuhan, mendesain, mengimplementasikan, dan memelihara sistem informasi. Dalam siklus hidup pengembangan *system* atau sering disebut dengan SDLC (*System Development Life Cycle*) terdapat beberapa metode yang bisa dipakai, seperti metode *Waterfall*, RAD, *Prototype*, dll. Beberapa tahapan dalam SDLC menurut para ahli dapat dilihat pada tabel 1.

Tabel 1. Tahapan Dalam SDLC (Kadir: 2003, 398)

Sumber	Tahapan-Tahapan dalam SDLC
Alter (1992)	Inisiasi, pengembangan, implementasi, dan operasi dan pemeliharaan.
Fabbri dan Schwab (1992)	Studi kelayakan, rencana awal, analisis sistem, desain sistem, dan implementasi sistem.
Hoffer George, dan Valacich (1998)	Identifikasi dan seleksi proyek, inisiasi dan perencanaan proyek, analisis, perancangan logis, perancangan fisik, implementasi, dan pemeliharaan.
McLeod (1998)	Perencanaan, analisis, perancangan, dan implementasi.
Turban, McLean, dan Wetherbe (1999)	Inisiasi proyek, analisis sistem dan studi kelayakan, analisis dan perancangan logis, akuisisi atau pengembangan, implementasi, operasi, evaluasi pasca audit, dan pemeliharaan.
Zwass (1998)	Studi kelayakan, analisis kebutuhan, perancangan logis, perancangan fisik, pengkodean dan pengujian, konversi, dan kajian pasca-implementasi

Pada tabel 1 dapat dilihat beberapa perbandingan tahapan-tahapan SDLC (*System Development Life Cycle*). Tahapan-tahapan tersebut kurang lebih hampir sama, yaitu meliputi analisis sistem, desain sistem, implementasi sistem, pengoperasian sistem dan pemeliharaan.

Menurut Pressman (2010: 39) model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*.



Gambar 1. Metode *Waterfall* (Sommerville, 2001)

Berikut ini adalah tahapan siklus hidup sistem dengan menggunakan metode *waterfall*:

1. *Requirements Analysis and Definition*

Tahapan ini muncul karena adanya permintaan terhadap sistem baru. Tahapan ini dimulai dari mengumpulkan kebutuhan dari *user*, kemudian menganalisis dan mendefinisikan kebutuhan yang harus dipenuhi oleh sistem atau *software* yang akan dibuat. *Software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. Tahap ini sering disebut dengan *Project Definition*. Dalam mengumpulkan analisis kebutuhan ini, pembuat sistem bisa melakukan dengan berbagai macam cara, antara lain seperti melakukan wawancara, riset terhadap sistem yang sedang berjalan, observasi lapangan, menyebar kuisisioner, pengamatan terhadap sistem yang serupa, dan membuat prototipenya (Kadir, 2003: 404).

2. *System and Software Design*

Membuat desain sistem dan desain *software* yang akan dibuat. Dalam tahap ini kebutuhan-kebutuhan yang sudah dikumpulkan diubah menjadi bentuk *blueprint software* sebelum mengubahnya ke dalam bentuk kode-kode, desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Proses ini harus didokumentasikan sebagai konfigurasi dari *software*.

3. *Implementation and Unit Testing*

Mengimplementasikan dengan menggunakan kode-kode yang dimengerti oleh komputer dengan menggunakan bahasa pemrograman yang ditentukan, dalam keadaan ini masih dibagi menjadi modul-modul kecil, serta diuji tiap unitnya.

4. *Integration and System Testing*

Agar dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain yang sudah dibuat harus diubah menjadi bentuk yang dapat dimengerti oleh mesin, yaitu diubah dengan bahasa pemrograman melalui proses yang bernama *coding*. Dari modul-modul kecil tadi, akan dijadikan menjadi satu sistem utuh (diintegrasikan) yang kemudian akan dilakukan *testing* secara keseluruhan.

5. *Operation and Maintenance*

Tahapan berikutnya adalah pengoperasian sistem, yang dimaksud dengan *operation* adalah uji coba semua fungsi-fungsi *software* agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

Maintenance dalam suatu sistem otomasi sangat diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya seperti itu.

4. Hasil dan Pembahasan

Setelah dipaparkan mengenai teori otomasi perpustakaan dan siklus hidup pengembangan sistem, maka dari sini dapat diambil beberapa kaitannya. Otomasi perpustakaan merupakan *software* yang digunakan untuk mengelola administrasi perpustakaan. Sedangkan sebuah *software* itu sendiri dibangun dengan memperhatikan langkah-langkah yang terdapat pada siklus hidup pengembangan sistem.

Beberapa hal yang perlu diperhatikan oleh pustakawan selaku pengguna otomasi perpustakaan terkait dengan siklus hidup pengembangan sistem informasi perpustakaan adalah:

1. *Requirement Analysis and Definition*

Pada tahap ini yang bisa dilakukan oleh pustakawan adalah mencatat kemampuan apa yang bisa dilakukan oleh sistem nantinya. Mencatat perangkat keras yang mungkin akan digunakan jika sistem sudah jadi juga bisa dilakukan dalam tahap ini, siapa calon pengguna sistem, kemampuan handal apa yang ingin dimiliki daripada sistem otomasi perpustakaan yang lama. Dapat juga dilakukan pemilihan dari mana sistem otomasi ini, seperti yang dipaparkan pada 2.1 apakah membuat sendiri, membeli perangkat lunak paket, atau melakukan *outsourcing*.

2. *System and Software Design*

Di tahapan ini pustakawan akan bekerja sama dengan pihak pembuat sistem, kira-kira desain seperti apa yang dirancang dari hasil analisa kebutuhan tadi, sebelum menuju ke proses pembuatan sistem.

3. *Implementation and Unit Testing*

Pada tahap implementasi dan pengujian unit, jika pembuat sistem otomasi perpustakaan adalah pihak luar, maka yang dapat dilakukan pustakawan adalah menyerahkannya kepada pihak *outsourcing*. Hal yang sama juga akan berlaku jika pembuat sistem adalah pihak internal perusahaan. Kecuali jika pustakawan juga memiliki kemampuan dalam mengubah desain sistem menjadi kode-kode, maka pustakawan bisa turut serta dalam tahapan ini.

4. *Integration and System Testing*

Seperti tahap sebelumnya, di tahap ini pustakawan yang tidak memiliki keahlian dalam hal pemrograman tidak bisa ikut andil di dalamnya.

5. *Operation and Maintenance*

Pada tahapan ini *programmer* bisa bekerja sama dengan pustakawan untuk mengecek apakah fungsi sistem otomasi perpustakaan yang sudah jadi tersebut tidak terdapat *error* atau *bug*. Dalam tahapan ini juga akan dilakukan pengecekan apakah fungsi-fungsi sudah sesuai seperti pada tahapan analisa kebutuhan sistem. Kemudian untuk tahap *maintenance*, pustakawan bisa memikirkan ke depannya nanti sistem otomasi perpustakaan ini akan dilakukan *updating* di menjadi seperti apa. Hal yang biasanya diabaikan oleh para pustakawan adalah pada tahap terakhir yaitu tahap *operation and maintenance*. Ketika perpustakaan menggunakan suatu *software*, maka mereka akan fokus ke *software* tersebut akan tetapi tidak memperhatikan bahwa sebuah *software* memiliki siklus hidup. Sehingga pustakawan akan mengabaikan tahap *maintenance*. Jika tahap ini diabaikan, maka yang terjadi adalah penggunaan *software* yang cenderung ketinggalan jaman atau usang karena tidak diupdate. *Updating software* penting dilakukan jika ternyata selama dalam pengoperasiannya terjadi *bug* atau *error* yang dijumpai. Jika terjadi hal yang demikian maka *maintenance* perlu dilakukan.

Pengembangan *software* tidak hanya sebatas pengupdatean *software*. Jika dilihat dari siklus hidup pengembangan sistem (Gambar 1), maka pengembangan *software* dapat dilakukan kembali ke proses *Requirement Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing, Operation and Maintenance* secara berurutan, jika hal ini dilakukan terus menerus, maka akan tercipta pengembangan sistem otomasi yang handal dan dapat mengefisienkan kinerja pustakawan pada khususnya.

5. Simpulan

Siklus hidup pengembangan sistem terdiri dari *Requirement Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing, Operation and Maintenance*. Beberapa tahap ini juga dapat diulang kembali ke tahap yang lain, sesuai dengan anak panah yang ada pada gambar siklus hidup. Dengan adanya siklus hidup pengembangan sistem ini, maka diharapkan pustakawan mau belajar untuk memahami bahwa *software* harus selalu dipantau, tidak hanya digunakan setiap harinya saja. Bila *software* mengalami *error*, maka harus segera dilakukan perbaikan supaya penggunaan sistem otomasi dan pelayanan dalam perpustakaan semakin maksimal.

Daftar Pustaka

Kadir, Abdul. 2003. *Pengenalan Sistem Informasi*. Yogyakarta: Andi.

Kadir, Abdul dan Terra CH. Triwahyuni. 2003. *Pengenalan Teknologi Informasi*. Yogyakarta: Andi.

- Lasa HS. 2005. Manajemen Perpustakaan. Yogyakarta : Gama Media.
- Saiful, Huda I. (2007), *Automasi Perpustakaan*, Makalah Tugas Pelatihan Jardiknas Magelang.
- Sommerville, Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak). Jakarta: Erlangga.
- Sulistyo-Basuki. 1993. Pengantar Ilmu Perpustakaan. Jakarta: Gramedia Utama.
- Supriyanto, Wahyu dan Ahmad Muhsin, 2008. *Teknologi Informasi Perpustakaan*. Yogyakarta: Kanisius.
- Pressman, R.S. 2010. Software Engineering : a practitioner's approach, McGraw-Hill. New York.