

Pengembangan Sistem Repositori Dokumen Akreditasi Institusi Perguruan Tinggi di Universitas Negeri Jakarta

Muhammad Rayhan Haroki¹, Hamidillah Ajie², M. Ficky Duskarnaen³

^{1,2,3} Pendidikan Teknik Informatika dan Komputer Fakultas Teknik
Universitas Negeri Jakarta

¹rayhan_ptik13@mahasiswa.unj.ac.id ²hamidillah@unj.ac.id, ³ duskarnaen@unj.ac.id

Abstrak

Tujuan dari penelitian ini adalah mengembangkan sistem repositori dokumen untuk pengakreditasi institusi perguruan tinggi di lingkungan Universitas Negeri Jakarta. Pengembangan sistem ini dilakukan untuk menyelesaikan masalah pencarian dokumen dalam proses akreditasi di lingkungan Universitas Negeri Jakarta. Metode pengembangan sistem yang digunakan adalah metode Agile model Extreme Programming. Pengembangan sistem dengan menggunakan extreme programming terdiri dari tahap planning, design, coding, dan testing. Kebutuhan sistem didapat melalui user stories. User stories tersebut kemudian diolah menjadi rancangan sistem lalu diimplementasikan pada tahap coding. Setelah melalui tahap coding, sistem harus melalui tahap testing untuk dinyatakan layak digunakan. Pengujian pada sistem ini menggunakan metode black box dengan teknik feature test. Skenario pengujian dibuat berdasarkan fungsi-fungsi yang terdapat pada sistem. Penguji memberi masukan/perintah pada sistem lalu menentukan apakah respons sistem sudah sesuai atau tidak dengan hasil yang diharapkan. Berdasarkan hasil pengujian, sistem yang dikembangkan dalam penelitian ini sudah sesuai dengan hasil yang diharapkan pada setiap skenario pengujian sehingga dapat disimpulkan bahwa sistem ini sudah layak digunakan untuk keperluan akreditasi di Universitas Negeri Jakarta.

Kata kunci : sistem, repositori, agile, extreme programming, black box.

1. Pendahuluan

Berdasarkan Keppres 093/1999 tanggal 4 Agustus 1999, Institut Keguruan dan Ilmu Pendidikan Jakarta berubah menjadi Universitas Negeri Jakarta (UNJ) dan peresmiannya dilaksanakan oleh Presiden Republik Indonesia pada tanggal 31 Agustus 1999 di Istana Negara.

Menurut Pasal 60 ayat 1 Undang-Undang RI Nomor 20 Tahun 2003 tentang sistem pendidikan nasional, akreditasi dilakukan untuk menentukan kelayakan program dan satuan pendidikan pada jalur pendidikan formal dan nonformal setiap jenjang dan jenis pendidikan.

Saat ini untuk lingkup perguruan tinggi, UNJ mendapatkan peringkat akreditasi "A" dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) berdasarkan SK BAN-PT No 763/SK/BAN-PT/AKRED/PT/VII/2015 dan berlaku sampai 10 Juli 2020. BAN-PT merupakan badan yang ditunjuk oleh pemerintah untuk melakukan akreditasi terhadap seluruh perguruan tinggi yang terdaftar di Indonesia termasuk program studi yang ada di dalamnya. Namun untuk tingkat program studi yang ada di UNJ, tidak semua mendapatkan peringkat akreditasi "A". Terdapat 32 program studi dengan peringkat "A", 53 program studi dengan peringkat "B", dan 7 program studi dengan peringkat "C". Dari semua program

studi yang ada, terdapat 17 program studi yang peringkat akreditasinya sudah/akan kedaluwarsa di tahun 2017. Data akreditasi bisa dilihat di situs resmi BAN-PT dengan alamat https://banpt.or.id/direktori/institusi/pencarian_instiusi untuk peringkat akreditasi institusi perguruan tinggi dan untuk peringkat akreditasi program studi https://banpt.or.id/direktori/prodi/pencarian_prodi.

Pelaksanaan akreditasi institusi perguruan tinggi maupun program studi membutuhkan berkas administrasi yang lengkap. Dalam pengisian borang akreditasi, program studi membutuhkan dokumen yang berasal dari fakultas. Selain itu, dalam penilaian akreditasi juga fakultas dilibatkan dalam pengisian borang. Hingga saat ini, UNJ masih menggunakan cara tradisional dalam administrasi berkas termasuk berkas akreditasi. Setiap berkas maupun dokumen yang sudah dicetak akan disimpan di tempat penyimpanan arsip di tingkat universitas maupun masing-masing program studi. Tidak ada tempat penyimpanan berkas di suatu tempat terpadu. Begitu pula dokumen yang berupa file asli dan belum tercetak, tidak tersimpan di tempat penyimpanan terpadu. Sehingga ketika akan diadakan akreditasi, maka administrator program studi maupun perguruan tinggi akan mencari dan mengumpulkan berkas maupun dokumen yang diperlukan.

Salah satu contoh kasus yang ada ketika akreditasi dilakukan adalah program studi harus meminta berkas kepada administrator UNJ maupun fakultas jika terdapat berkas yang diperlukan namun tidak tersimpan di arsip program studi itu sendiri. Hal ini tentu akan memperlambat proses persiapan akreditasi karena harus melewati berbagai perizinan untuk mengambil berkas yang diperlukan dari pihak luar. Selain itu, tidak ada sistem pencatatan daftar dokumen yang membuat pencarian dokumen semakin sulit.

Berdasarkan beberapa permasalahan yang ada, maka dibutuhkan sebuah sistem yang dapat menyimpan dokumen maupun berkas yang berkaitan dengan akreditasi secara terpusat dan terpadu. Repositori merupakan sebuah wadah yang menyimpan data dalam bentuk digital. Sistem repositori dapat digunakan untuk menyimpan berkas akreditasi UNJ serta seluruh program studi yang terdapat di UNJ. Untuk memudahkan dalam mengakses sistem tersebut maka sistem yang akan dikembangkan merupakan sistem berbasis web. Sistem ini menghubungkan orang di tempat yang berbeda melalui jaringan khususnya internet. Sistem repositori perlu dikembangkan untuk memudahkan penyimpanan serta pencarian dokumen maupun berkas yang berkaitan dengan administrasi akreditasi.

Tujuan penelitian ini adalah mengembangkan sistem repositori dokumen untuk pengakreditasi institusi perguruan tinggi di lingkungan Universitas Negeri Jakarta. Pengembangan sistem ini dilakukan untuk menyelesaikan masalah pencarian dokumen dalam proses akreditasi di lingkungan Universitas Negeri Jakarta.

2. Dasar Teori

2.1. Akreditasi Institusi Perguruan Tinggi

Menurut BAN-PT (2011) dalam Buku 3: Pedoman Penyusunan Borang, akreditasi institusi perguruan tinggi adalah proses penilaian terhadap institusi secara keseluruhan untuk mengetahui komitmen institusi terhadap penyelenggaraan akademik dan manajemen institusi, yang didasarkan pada standar akreditasi yang telah ditetapkan. Akreditasi dilakukan oleh BAN-PT (Badan Akreditasi Nasional Perguruan Tinggi) terhadap semua perguruan tinggi di Indonesia.

Evaluasi terhadap mutu kinerja institusi perguruan tinggi dilakukan melalui asesmen terhadap borang akreditasi dan laporan evaluasi diri institusi perguruan tinggi oleh tim asesor yang terdiri atas berbagai keahlian terkait yang berpengalaman dan memahami hakikat penyelenggaraan perguruan tinggi, baik dalam bidang akademik maupun bidang manajemen.

Evaluasi-diri merupakan upaya institusi perguruan tinggi untuk mengetahui gambaran mengenai kinerja dan keadaan dirinya melalui pengkajian dan analisis yang dilakukan oleh institusi perguruan tinggi sendiri. Pengkajian dan analisis itu

dapat dilaksanakan dengan memanfaatkan pakar sejawat dari luar perguruan tinggi, sehingga evaluasi-diri dapat dilaksanakan secara obyektif. Hasil evaluasi diri digunakan untuk memperbaiki mutu kinerja dan produk institusi perguruan tinggi.

2.2. Sistem

Menurut Sutarnan (2009) dalam bukunya yang berjudul Pengantar Teknologi Informasi, sistem adalah kumpulan elemen yang saling berinteraksi dalam suatu tujuan utama. Sedangkan menurut Jogiyanto (2010), dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen.

2.3. Pengembangan Sistem

Sistem yang ada saat ini merupakan hasil dari sebuah proses yang disebut dengan pengembangan sistem. Menurut Jogiyanto (2010) dalam bukunya Analisis dan Desain Sistem Informasi, menjelaskan bahwa pengembangan sistem dapat berarti menyusun sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau untuk memperbaiki sistem yang sudah ada sedangkan metodologi pengembangan sistem adalah metode prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan untuk mengembangkan suatu sistem informasi.

2.4. Repositori

Menurut Kamus Besar Bahasa Indonesia Daring, repositori adalah suatu tempat penyimpanan. Definisi lain dari repositori adalah suatu wadah atau tempat penyimpanan dari berbagai data atau *file* digital. Sebuah repositori dibuat oleh universitas tertentu atau lembaga penelitian lainnya dikenal sebagai sebuah repositori institusi. Hal ini dapat dimaksudkan untuk mengumpulkan dan melestarikan data digital. Hal ini juga dapat berisi lembaga perpustakaan digital, koleksi cetak dan dokumen naskah, arsip publik dan material grafis serta *software open source* yang berasal dari lembaga atau di tempat lain yang digunakan suatu lembaga untuk digunakan di ruang lingkungannya, dan umumnya tersedia bagi siapa saja. Hal ini juga dapat berisi output administrasi lembaga, seperti laporan, direktori, dan dokumentasi arsip lokal atau lainnya yang diperlukan untuk mendukung kebutuhan lembaga tersebut Prasetyo (2016).

Dapat disimpulkan bahwa definisi repositori yang sesuai dalam pengembangan sistem repositori adalah sebuah tempat atau wadah untuk menyimpan data berbentuk *file* digital.

2.5. Agile Software Development

Menurut Pressman (2010), *Agile Software Development* adalah sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, dimana persyaratan dan solusi

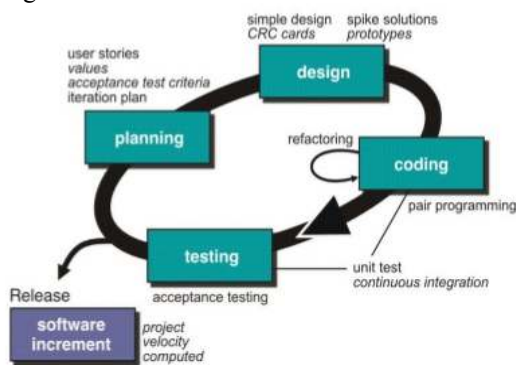
berkembang melalui kolaborasi antar tim yang terorganisir. Istilah ini diciptakan pada tahun 2001 ketika *Agile Manifesto* dirumuskan.

Metode *Agile* umumnya mempromosikan disiplin proses manajemen proyek yang mendorong inspeksi dan adaptasi; filosofi kepemimpinan yang mendorong kerja sama dalam tim, pengorganisasian dan akuntabilitas; praktek rekayasa yang memungkinkan pengiriman perangkat lunak berkualitas tinggi dengan cepat; dan pendekatan bisnis yang sejalan dengan pengembangan kebutuhan pelanggan dan tujuan perusahaan.

2.6. eXtreme Programming (XP)

Extreme Programming menurut Gelinis (2008) adalah pendekatan disiplin pengembangan perangkat lunak yang menekankan keterlibatan pelanggan dan mempromosikan kerja sama tim. Menurut Pressman (2010), *Extreme Programming (XP)* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi.

Dapat disimpulkan bahwa *extreme programming* adalah suatu metode pengembangan perangkat lunak yang menekankan kepada adaptasi terhadap kebutuhan pengguna serta kerja sama tim yang kuat.



Gambar 2.1. Alur Model *Extreme Programming*

Tahapan-tahapan dari *Extreme Programming* yang terdapat pada gambar 2.1 terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *design* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing*. Tahapan ini bisa berulang jika hasil pengujian menunjukkan adanya fungsi yang belum tercapai sesuai dengan kebutuhan sistem.

Unsur-unsur lain dari *Extreme Programming* meliputi *paired programming* pada tahapan *coding*, *unit testing* pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara

programmer dan pelanggan ketika terjadi perubahan kebutuhan pelanggan seiring berlalunya waktu berlalu.

Metode ini membawa unsur-unsur yang menguntungkan dari praktik rekayasa perangkat lunak tradisional ke tingkat “ekstrem”, sehingga metode ini dinamai *Extreme Programming*. Unsur-unsur yang menjadi karakteristik metodologi adalah kesederhanaan, komunikasi, umpan balik, dan keberanian.

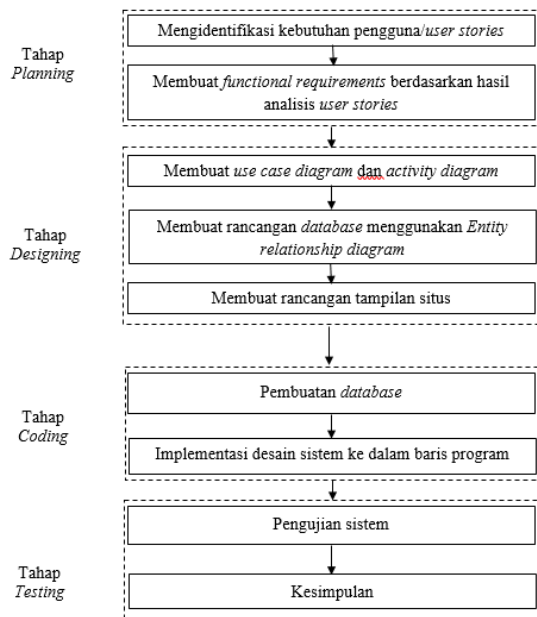
2.7. Object-Oriented Programming

Object-Oriented Programming (OOP) atau pemrograman berorientasi obyek (PBO) adalah gaya pemrograman yang berfokus pada penggunaan obyek untuk merancang dan membangun aplikasi. (Yaiser, 2011). Menurut Dwi (2013) OOP merupakan teknik membuat suatu program berdasarkan obyek dan apa yang bisa dilakukan obyek tersebut. OOP terdiri dari obyek-obyek yang berinteraksi satu sama lain untuk menyelesaikan sebuah tugas. Kode-kode di-breakdown agar lebih mudah di-manage. Breakdown berdasarkan obyek-obyek yang ada pada program tersebut. Dianjurkan diimplementasikan untuk program dengan berbagai ukuran karena lebih mudah untuk men-debug. Jadi, OOP adalah suatu teknik atau gaya pemrograman yang berorientasi kepada obyek yang ada di dalam sistem.

3. Metodologi

Penelitian dilakukan dengan menggunakan pendekatan metode pengembangan perangkat lunak model *eXtreme Programming (XP)*. Proses pengembangan sistem ini dilakukan dengan menggunakan *notebook* Asus N46VZ sebagai alat dan hasil wawancara sebagai bahan penelitian.

Pada bagian ini dijelaskan tentang komponen dan alat yang digunakan selama penelitian atau perancangan dan bagaimana cara/ langkah-langkah melakukan penelitian atau perancangan yang telah dilakukan tersebut. Gambaran mengenai cara melakukan penelitian atau perancangan ada kalanya akan lebih baik jika direpresentasikan dalam bentuk diagram alir. Penomoran gambar mengikuti penomoran Bab seperti contoh Gambar 3.1 Contoh Diagram.



Gambar 3.2. Diagram Alir Penelitian

Gambar 3.1 menunjukkan diagram alir penelitian. XP dipilih sebagai dasar pengembangan sistem karena cocok untuk menghasilkan perangkat lunak berkualitas baik dalam waktu singkat dan tim yang kecil. XP memiliki 4 tahap utama yaitu *planning* (perencanaan), *design* (perancangan), *coding* (penulisan baris kode program/ implementasi), dan *testing* (pengujian).

Tahap pertama adalah *planning* atau perencanaan. Sebelum melakukan perencanaan dalam pengembangan sistem, dilakukan pengumpulan data berupa *user stories*. *User stories* didapat melalui wawancara dengan *user* yang mengetahui permasalahan yang ada. Setelah mengetahui masalah yang ada maka *user stories* akan dianalisis untuk menentukan kebutuhan fungsional sistem, kebutuhan pengguna, dan fitur utama dari sistem yang akan dikembangkan. Selain *user stories*, studi literatur juga dibutuhkan untuk mengetahui kebutuhan sistem.

Tahap kedua adalah *design* atau perancangan sistem. Perancangan pada XP tetap mengedepankan prinsip *Keep it Simple (KIS)*. Rancangan sistem yang dibuat merupakan representasi dari sistem guna mempermudah pengembang dalam membangun sistem dan didapatkan dari hasil analisis *user stories*. Rancangan dari sistem pada penelitian ini digambarkan dengan UML (*Unified Modelling Language*). UML memiliki beberapa jenis diagram. Pengembangan sistem repositori ini akan menggunakan *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

Diagram pertama yang akan dibuat adalah *Use Case Diagram*. Pembuatan *use case diagram* didasarkan pada *user stories* yang telah didapatkan sebelumnya. *User stories* diterjemahkan menjadi daftar kebutuhan fungsional, kemudian setiap kebutuhan fungsional dijadikan sebuah *case*.

Use case diagram memiliki tiga komponen utama yaitu *actor* sebagai pengguna sistem, lingkup sistem, dan *use case*. *Use case* merupakan gambaran fungsionalitas dari suatu sistem sehingga *actor* atau pengguna sistem paham mengenai kegunaan sistem yang akan dibuat.

Diagram kedua yang akan dibuat adalah *Activity Diagram*. *Activity diagram* menggambarkan *workflow* aktivitas dari suatu sistem atau bagian dari sistem melalui aliran kontrol dalam mengeksekusi sebuah fungsi/*use case*.

Diagram selanjutnya yang akan dibuat adalah *Class Diagram*. *Class diagram* dibuat dalam pengembangan sistem berorientasi obyek untuk menggambarkan class dari suatu sistem serta atribut dan fungsi apa saja yang bisa dilakukan oleh *class* tersebut.

Use case diagram, *activity diagram*, dan *class diagram* merupakan penggambaran dari alir kerja sistem. Tahap perancangan sistem selanjutnya adalah merancang database. Struktur rancangan database digambarkan dengan *Entity Relationship Diagram (ERD)*. ERD digunakan untuk menggambarkan struktur *logical database* dalam bentuk diagram. ERD menyediakan cara yang sederhana dan mudah untuk memahami berbagai komponen dalam desain *database*. Setelah melakukan perancangan sistem, langkah berikutnya adalah merancang tampilan antarmuka situs. Perancangan tampilan antarmuka situs dilakukan berdasarkan kebutuhan sistem yang digambarkan melalui beberapa diagram sebelumnya. Halaman situs serta konten di dalamnya dirancang dengan membuat *layout wireframe* dan *mockups*.

Tahap pengembangan berikutnya adalah *Coding*. Tahap ini merupakan proses implementasi dari perancangan sistem yang telah dibuat. Langkah pertama ialah pembuatan database dalam sistem ini menggunakan MySQL. Struktur *database* serta struktur *table* dan relasi di dalamnya mengacu pada ERD.

Langkah berikutnya dalam tahap ini adalah penulisan baris program atau *coding*. Langkah ini meliputi *coding* untuk membuat tampilan halaman situs berdasarkan *layout wireframe* dan *mockups* serta memasukkan fungsi-fungsi sistem ke dalam halaman yang dibuat berdasarkan kebutuhan sistem. Tahap penulisan kode program menggunakan framework PHP Code Igniter agar lebih mudah dan cepat. Setelah itu, sistem yang sudah dibuat dihubungkan dengan *database*.

Pada tahap *coding* juga disisipkan tahap *refactoring*. *Refactoring* adalah proses mengubah sistem perangkat lunak sedemikian rupa tanpa mempengaruhi fungsional sistem dalam memperbaiki struktur internalnya. Hal ini dilakukan untuk meminimalkan kemungkinan adanya *bugs*.

Tahap terakhir adalah tahap *Testing*. Sistem yang telah dibangun diuji coba dengan menggunakan metode *blackbox* untuk menunjukkan apakah kebutuhan awal *user* atau *user stories* sudah dipenuhi.

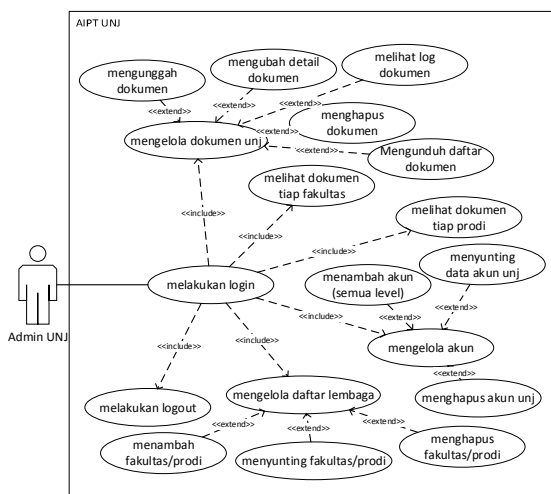
Sistem yang sudah sesuai dengan kebutuhan dalam pengujian *blackbox* dapat disimpulkan telah layak untuk dirilis. Namun jika hasil pengujian belum sesuai maka tahapan pengembangan sistem diulang kembali.

3.1 Tahap *Planning*

Tahap perencanaan merupakan tahap untuk mendapatkan kebutuhan sistem. Tahap ini dimulai dengan mendengarkan user stories. User stories didapat dengan mewawancarai koordinator program studi Pendidikan Teknik Informatika dan Komputer yaitu Dr. Yuliatri Sastrawijaya, M.Pd dan koordinator program studi Pendidikan Fisika, Dr. Esmar Budi, M.T. Hasil wawancara diuraikan pada Lampiran 3. Studi literatur dari buku pedoman akreditasi juga dilakukan untuk menyusun daftar kebutuhan fungsional dari sistem yang dikembangkan.

3.2 Tahap *Design*

Rancangan sistem yang dibuat untuk pertama kali adalah *use case diagram*. Diagram ini dibuat berdasarkan kebutuhan sistem.



Gambar 3.2 Use Case Diagram Admin UNJ

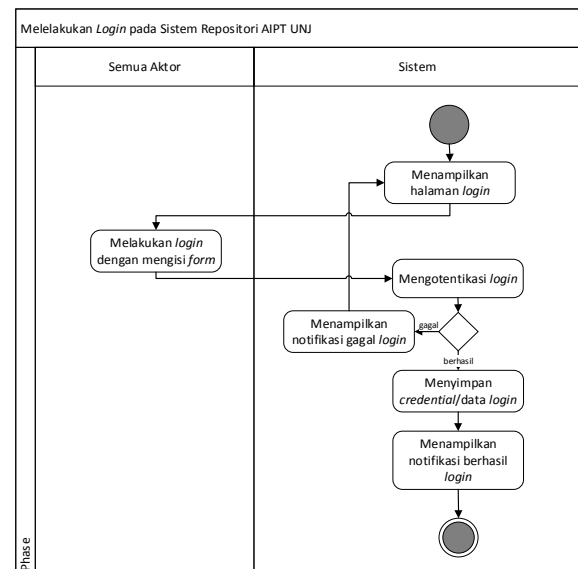
Gambar 3.2 merupakan *use case diagram* Admin UNJ di dalam sistem repositori. Setiap lingkaran di dalam sistem disebut dengan *case*. Setiap *case* menggambarkan aktivitas apa saja yang dapat dilakukan oleh aktor di dalam sebuah sistem yang digambarkan dengan persegi yang disebut dengan *boundary*.

Aktivitas pertama yang dilakukan oleh Admin UNJ serta aktor lainnya adalah melakukan *login*. Setelah *login*, barulah Admin UNJ dapat melakukan aktivitas yang lain seperti mengelola dokumen UNJ, melihat dokumen fakultas maupun prodi, mengelola daftar lembaga, mengelola akun, dan keluar dari sistem dengan melakukan *logout*. Ketergantungan suatu aktivitas terhadap aktivitas lainnya digambarkan dengan anak panah terputus-putus dengan label *include*. Pada aktivitas mengelola dokumen, mengelola daftar lembaga, dan mengelola

akun, Admin UNJ dapat melakukan aktivitas lain seperti menambah akun dan mengunggah dokumen yang bersifat opsional dan digambarkan dengan anak panah terputus-putus dengan label *extend*.

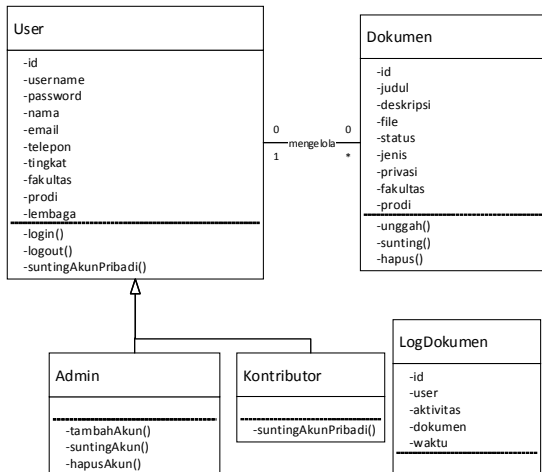
Aktivitas kelola dokumen terbatas hanya bisa dilakukan pada dokumen yang sesuai dengan tingkatan akun tersebut. Contohnya Admin UNJ hanya dapat mengelola dokumen UNJ dan seterusnya. Aktivitas menambah akun dapat dilakukan oleh admin pada semua tingkat untuk menambah akun dengan tingkat yang sama dan tingkat di bawahnya. Admin UNJ dapat membuat akun untuk tingkat universitas, fakultas, dan program studi. Tetapi untuk aktivitas menyunting dan menghapus akun hanya bisa dilakukan oleh admin untuk akun dengan tingkat yang sama. Aktivitas mengelola daftar lembaga hanya bisa dilakukan oleh Admin UNJ.

Setelah membuat *use case diagram*, langkah selanjutnya adalah membuat *activity diagram*. *Activity diagram* menjabarkan alur kerja dan interaksi antara aktor dan sistem dalam sebuah case/aktivitas. Terdapat beberapa aktivitas yang dilakukan oleh banyak aktor, oleh karena itu terdapat beberapa *activity diagram* yang dibuat untuk beberapa aktor sekaligus.



Gambar 3.3 Activity Diagram Login

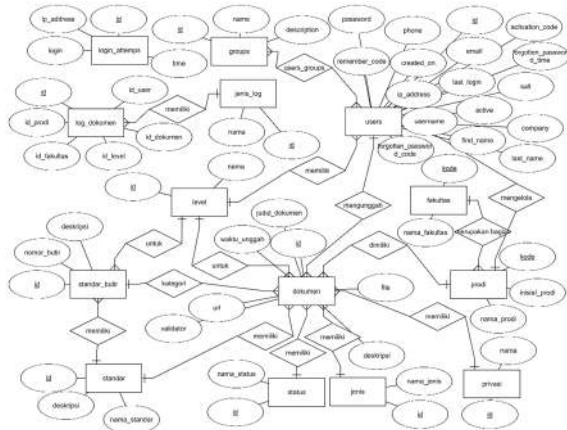
Gambar 3.3 menggambarkan aktivitas *login* yang dapat dilakukan oleh semua aktor atau bisa disebut dengan *user*. Ketika *user* membuka situs maka sistem akan menampilkan halaman *login*. Setelah *user* mengisi data dan menekan tombol *login*, maka sistem akan mengotentikasi data tersebut. Jika otentikasi berhasil, maka sistem akan menyimpan data *login* user lalu menampilkan notifikasi berhasil *login*. Jika otentikasi gagal, maka sistem akan menampilkan notifikasi gagal dan mengembalikan halaman *form login*.



Gambar 3.4 Class Diagram

Gambar 3.4 merupakan *class diagram* dari sistem yang dikembangkan. *Class diagram* yang dibuat terdiri dari beberapa *class* utama dalam sistem yaitu *class User*, *Dokumen*, dan *LogDokumen*. *Class Admin* dan *kontributor* merupakan sub-*class* dari *User*. *Class* terdiri dari atribut dan *method* atau fungsi yang bisa dilakukan dari *class* tersebut. *User* dapat mengelola dokumen tergantung tingkat serta hak aksesnya.

Berdasarkan hasil analisis kebutuhan dan diagram sebelumnya, maka perancangan database sistem yang dikembangkan digambarkan pada Gambar 3.5.



Gambar 3.5 Entity Relationship Diagram

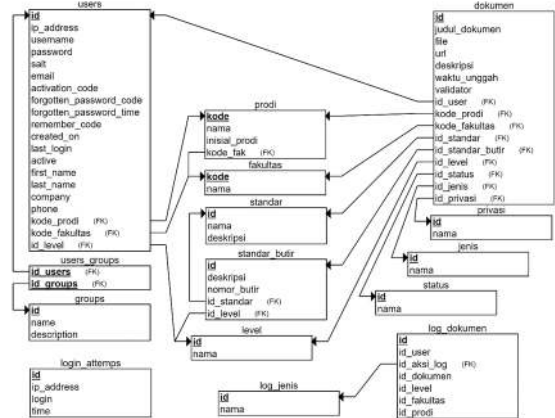
4. Hasil dan Analisis

4.1 Tahap Coding

Tahap coding merupakan tahap implementasi dari perancangan sistem. Sebelum melanjutkan ke tahap ini, penyedia layanan web/web server harus tersedia. Perangkat notebook Asus N46VZ dapat dijadikan web server dengan meng-install aplikasi XAMPP. Aplikasi XAMPP mencakup web server Apache dan Database Management System (DBMS) MySQL yang dipakai pada sistem ini.

Setelah web server tersedia, langkah selanjutnya adalah implementasi rancangan database yang dibuat menggunakan DBMS MySQL. Database sistem ini

memiliki relasi antar tabel di dalamnya yang ditunjukkan pada gambar 4.1.



Gambar 4.1. Relasi Database

Langkah selanjutnya setelah database dibuat adalah pembuatan situs dengan menulis kode program. Penulisan kode program di sisi *server (backend)* dilakukan dengan menggunakan bahasa pemrograman PHP dan *framework* Code Igniter. Bagian antar muka situs (*frontend*) dibuat dengan menggunakan HTML, CSS dan jQuery.

Penulisan kode program untuk sistem ini dibuat sesuai dengan hasil rancangan sistem. Kebutuhan-kebutuhan sistem diterjemahkan ke dalam algoritma tertentu yang ditulis dalam kode program. Fungsi *login*, *logout*, dan kelola akun dibuat dengan menggunakan *library* yang tersedia untuk *framework* Code Igniter yaitu IonAuth. *Library* ini dapat diunduh secara gratis. Selain menyediakan banyak *file* yang berisi kode program, *library* ini juga menyediakan tabel-tabel database untuk menjalankan fungsinya. Terdapat empat tabel bawaan dari *library* ini yaitu tabel *users*, *users_groups*, *groups*, dan *login_attempts*. Beberapa penyesuaian pada tabel *users* dilakukan untuk memenuhi kebutuhan sistem. Kode program yang sudah disediakan IonAuth juga disesuaikan dengan kebutuhan sistem.

Fungsi *login* dan *logout* menggunakan *library* IonAuth dengan sedikit penyesuaian dalam kode programnya. Halaman *login* dibuat dengan menggunakan HTML dan CSS sedangkan fungsi *logout* tidak memiliki halaman khusus, hanya notifikasi berhasil *logout*.

Fungsi kelola akun terdapat banyak penyesuaian dalam kode program IonAuth. Proses menambah akun pada sistem ini membutuhkan identitas lembaga dari *user* yang akan dibuatkan akun untuk dimasukkan ke dalam tabel *users*, tetapi kolom identitas lembaga *user* tersebut tidak tersedia secara *default* pada IonAuth. Karena itu, baik pada kode program dan *database* dilakukan penyesuaian untuk menyertakan identitas lembaga *user*. Fungsi-fungsi yang ada pada kode program IonAuth disalin lalu ditulis pada *file* baru dengan beberapa perubahan. Lalu tabel *users* pada *database* ditambahkan kolom *id_level*, *kode_fakultas*, dan *kode_prodi*. Sedangkan

untuk fungsi lain seperti kelola dokumen, kelola lembaga, dan log dokumen ditulis dari awal.

Tampilan situs dibuat dengan menggunakan HTML. *Framework* CSS Bootstrap dan Materialize digunakan untuk menambah estetika tampilan situs. Selain itu terdapat beberapa *library* yang digunakan yaitu AdminLTE pada halaman admin dan Datatables untuk menampilkan tabel. Bahasa pemrograman Javascript dan JQuery juga digunakan untuk mengolah tampilan. Pembuatan tampilan situs ini didasarkan pada rancangan tampilan yang sudah dibuat pada tahap *designing*.

Proses *refactoring* juga dilakukan pada tahap *coding* khususnya saat penulisan kode program. Proses ini dilakukan untuk mengurangi kemungkinan *bugs* yang ditemukan pada tahap pengujian. Pada tahap ini juga terdapat beberapa penambahan kebutuhan sistem sehingga terjadi perubahan pada daftar kebutuhan fungsional, perancangan sistem dan struktur *database*.

4.2 Hasil Tahap Coding

Tahap *coding* menghasilkan sebuah produk situs web utuh bernama “Sistem Repositori Akreditasi Institusi Perguruan Tinggi Universitas Negeri Jakarta”. Produk ini harus dinyatakan lulus tahap *testing* agar layak digunakan. Gambar 4.2 merupakan *screenshot* dari hasil pembuatan salah satu halaman situs.



Gambar 4.2 Halaman Dashboard Admin

4.3 Hasil Pengujian

Hasil pengujian dalam penelitian ini merupakan hasil dari tahap *testing*. Tabel 4.1 merupakan hasil dari pengujian Sistem Repositori Dokumen Akreditasi Institusi Perguruan Tinggi Universitas Negeri Jakarta untuk beberapa skenario.

Tabel 4.1 Sebagian Hasil Pengujian Fungsional

No	Fungsi	Skenario	Hasil Yang Diharapkan	Hasil
1.	Login	User melakukan login dengan data yang benar	Sistem menampilkan notifikasi berhasil login dan menampilkan halaman utama	Sesuai
2.	Login	User melakukan login dengan data yang tidak benar	Sistem menampilkan halaman login dengan notifikasi kesalahan gagal login	Sesuai

No	Fungsi	Skenario	Hasil Yang Diharapkan	Hasil
3.	Logout	User memilih logout	Sistem menampilkan notifikasi berhasil logout lalu menampilkan halaman login	Sesuai
4.	Lihat Dokumen	User melakukan pencarian dokumen dengan memilih pilihan lembaga dan standar dokumen yang tersedia	Sistem menampilkan hasil pencarian daftar dokumen sesuai dengan pilihan user	Sesuai

Pengujian sistem ini sudah meliputi keseluruhan fungsi yang dibutuhkan termasuk kebutuhan yang ditambahkan pada saat pengembangan sistem. Berdasarkan hasil pengujian, produk yang diuji sudah memenuhi keseluruhan kebutuhan sistem. Tidak terjadi kesalahan maupun ketidaksesuaian pada saat pengujian dilakukan sesuai dengan instrumen yang tersedia.

Hasil penelitian berupa produk yang telah melewati tahap pengujian dan dinyatakan layak dapat digunakan untuk proses pengumpulan dokumen akreditasi di lingkungan Universitas Negeri Jakarta untuk jenjang sarjana. Produk ini juga dapat dikembangkan agar bisa digunakan untuk seluruh jenjang maupun lembaga yang ada di Universitas Negeri Jakarta.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Sistem repositori dokumen akreditasi dikembangkan melalui tahapan yang ada di dalam metode pengembangan *Agile* model *Extreme Programming* yaitu *planning*, *designing*, *coding*, dan *testing*. User yang dapat menggunakan sistem ini adalah admin dari tingkat program studi, fakultas, dan universitas. Fitur-fitur yang terdapat pada sistem ini disesuaikan dengan kebutuhan pengguna dan juga didasarkan pada buku pedoman akreditasi yang dibuat oleh BAN-PT.

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan metode *black-box* untuk setiap fungsi, sistem yang dikembangkan dalam penelitian ini sudah sesuai dengan hasil yang diharapkan pada setiap skenario pengujian. Sehingga dapat disimpulkan bahwa pengembangan sistem repositori dokumen akreditasi institusi perguruan tinggi di Universitas Negeri Jakarta dapat dilakukan dengan menggunakan metode pengembangan *Agile* model *Extreme Programming* untuk menyelesaikan permasalahan dalam pencarian dokumen yang berkaitan dengan akreditasi tingkat perguruan tinggi dan program studi tingkat sarjana di lingkungan Universitas Negeri Jakarta.

5.2 Saran

Agar berjalan dengan baik, sistem membutuhkan seorang admin yang paham mengenai permasalahan akreditasi. Beberapa pengembangan yang dapat dilakukan lebih lanjut yaitu sistem dapat mengetahui dokumen mana yang sudah lengkap dan belum lengkap. Hal ini juga berkaitan dengan kebutuhan lain yang tentu akan berubah dan berkembang seiring waktu. Sehingga pengembangan sistem perlu dilakukan secara terus-menerus.

Daftar Pustaka:

- Badan Pengembangan dan Pembinaan Bahasa, (2016), *Kamus Besar Bahasa Indonesia Daring*, <http://kbbi.kemdikbud.go.id/entri/repositori>, Diakses 22 Agustus 2017.
- BAN-PT, (2011), *Buku 3: Pedoman Penyusunan Borang*, Jakarta, BAN-PT, pp. 2
- Dwi, A., dkk, (2013), *Modul Praktikum Pemrograman Berorientasi Obyek*, Bandung, Institut Teknologi Telkom, pp. 31
- Gelinas, U.J., dkk., (2008), *Business Processes and Information Technology*, Hong Kong, The Open University of Hong Kong, pp. 219
- Jogiyanto, H.M., (2010), *Analisis dan Desain Sistem Informasi*, Edisi IV, Yogyakarta, Andi Offset, pp. 54-59
- Prasetyo, W.D, (2016), *Aplikasi Repositori Software Berbasis Web di Universitas Negeri Jakarta* [skripsi], Jakarta, Fakultas Teknik Universitas Negeri Jakarta, pp. 9
- Pressman, R.S, (2010), *Software Engineering: a Practitioner's Approach*, 7th edition. New York: McGraw-Hill, pp. 67-73
- Sutarman, (2009), *Pengantar Teknologi Informasi*, Yogyakarta, Bumi Aksara, pp. 5
- Yaiser, Michelle. (2011). *Object-oriented programming concepts: Objects and classes*. <http://www.adobe.com/devnet/actionscript/learning/ooop-concepts/objects-and-classes.html>, Diakses 22 Agustus 2017.