DIFUSI
Volume 1, No. 2, Juli 2018

APLIKASI AUDIT MUTU INTERNAL ONLINE STUDI KASUS SPM POLITEKNIK NEGERI BANDUNG

Ade Chandra Nugraha[1)], Nurjannah Syakrani[2)]
[1,2] Jurusan Teknik Komputer, Politeknik Negeri Bandung
e-mail: [1)] chandra@jtk.polban.ac.id [2)] nurjannahsy@gmail.com

Abstrak

Salah satu program dari unit Sistem Penjaminan Mutu (SPM) *di Politeknik Negeri Bandung yang tercantum dalam kalender akademik tahunan adalah Audit Mutu Internal (AMI). Dalam tiga tahun terakhir, program ini telah dilaksanakan secara online menggunakan aplikasi bernama PAMIOL. Penyimpanan data hasil Audit dalam PAMIOL menggunakan link data yang disimpan dalam format Excel maupun Word. Penggunaan dan evaluasi terhadap PAMIOL sebagai media AMI menunjukkan perlunya perbaikan serta disain ulang aplikasi. Berdasarkan analisis PAMIOL dan studi borang program studi (BORANG 3A) BAN PT sebagai salah satu instrumen audit, dibuat rancangan dan implementasi media audit mutu internal online baru bernama AMIOnline untuk perguruan tinggi atau instansi umum. AMIOnline memiliki kemampuan entry pertanyaan audit yang bersifat dinamis, mencakup definisi model pertanyaan, tipe data jawaban, juga formulasi penilaiannya yang dihitung secara otomotis oleh aplikasi ataupun dari entri skor oleh auditor, menghasilkan laporan asesmen, serta pilihan data pendukung untuk laporan AMI. Aplikasi ini juga memiliki fitur merekam data master auditor, auditee dan fitur pemetaan penugasan Auditor dengan pengaturan waktu aktif audit oleh admin. Pengembangan aplikasi AMIOnline ini berdasarkan metodologi pengembangan perangkat lunak waterfall dan menggunakan basis data relasional. Aplikasi AMIOnline sedang dipersiapkan untuk mendukung AMI SPM Polban mulai tahun 2017 dengan kemungkinan modifikasi serta fitur pengembangan template pertanyaan audit lainnya.*

Kata Kunci: *PAMIOL, AMI, model pertanyaan, formulasi penilaian, AMIOnline.*

1. PENDAHULUAN

Politeknik Negeri Bandung (Polban), sebagai salah satu perguruan tinggi, terikat dengan banyak regulasi Nasional, diantaranya Permenristekdikti Nomor 62 tahun 2016 pasal 2 [1], yang menyatakan bahwa tujuan dibangunnya Sistem Penjaminan Mutu (SPM) Pendidikan Tinggi adalah untuk menumbuh kembangkan budaya mutu di Perguruan Tinggi. Dengan demikian SPM bertugas untuk memonitor, mengevaluasi dan mengendalikan standar pelaksanaan Tridarma Perguruan tinggi di Institusi. Permenristekdikti Nomor 44 tahun 2015 mensyaratkan adanya implementasi Standar Nasional Pendidikan Tinggi (SNPT) [2] di setiap Perguruan tinggi. Karenanya POLBAN pun mempersiapkan standar turunan dari setiap 8 standar per darma dalam Tridarma Perguruan Tinggi. Sudah tentu standar-standar tersebut perlu diaudit keberadaannya ataupun implementasinya agar siklus PPEPP (Penetapan, Pelaksanaan, Evaluasi Pengendalian dan Peningkatan) dapat dilaksanakan dengan baik dan terus menerus sehingga terwujud PT yang unggul.

Audit mutu akademik adalah pemerik-saan dan verifikasi secara independen terhadap proses pencapaian mutu akademik sesuai dengan standar akademik yang telah ditetapkan. Audit mutu akademik dibagi menjadi dua, yaitu audit mutu internal (AMI) dan audit mutu eksternal (AME). AME secara nasional dilakukan oleh BAN PT atau lembaga lain dengan cara yang ditetapkan oleh lembaga akreditasi [3].

Dalam rangka menjalankan Permen-ristekdikti No. 32 tahun 2015 [4] tentang Akreditasi Program Studi dan Perguruan Tinggi, Permenristekdikti No. 61 tahun 2016 [5] tentang Pangkalan Data Perguruan Tinggi, maka melalui Surat Keputusan Direktur Politeknik Negeri Bandung No. 0139/K8.R/OT/2011 [6] ditetapkan Piagam Pendirian SPM Polban. Struktur pengelolaan di SPM Polban terdiri atas Ketua, koordinator SPMI (internal), SPME (eksternal) dan PDPM (Pangkalan data pengendali mutu). Dalam menjalankan tugasnya SPM Polban dilengkapi GPM (Gugus Pengendali Mutu) di tingkat Jurusan dan TPM (Tim Pengendali Mutu) ditingkat Program studi.

Untuk melaksanakan proses monitoring dan evaluasi terhadap proses penjaminan mutu di POLBAN, maka SPM POLBAN melaksanakan Audit Mutu Internal (AMI). Proses AMI dilakukan ditingkat pusat maupun jurusan [7] oleh sejumlah SDM terlatih atau bersertifikat yang ditugaskan sebagai auditee, auditor maupun MP (manager program) AMI.

AMI membutuhkan sumberdaya pendukung, salah satunya adalah aplikasi perangkat lunak komputer yang diberi nama PAMIOL (Polban Audit Mutu Internal Online). Aplikasi ini ditujukan untuk uji eksistensi standar (SNPT) bagi semua unit dan versi borang 3A BAN PT untuk khusus Prodi [8]. Selama tiga kali (tiga tahun) penggunaan PAMIOL, ditemukan beberapa kekurangan diantaranya data disimpan pada file Excel dan menggunakan link data, sehingga setiap kali akan dilakukan proses audit maka admin harus mempersiapkan file Excel. Kemudian auditor akan menghitung score secara manual yang dientrikan ke PAMIOL serta mengunggah kembali file jawaban audit. Selain adanya perhitungan manual oleh auditor, beberapa kekurangan aplikasi PAMIOL yang sudah teridentifikasi adalah adanya duplikasi daftar auditor dan audite, serta proses *unlock* lembar kerja oleh satu auditor yang berpengaruh pada auditor pasangannya [9].

Aplikasi baru AMIOnline dibangun untuk mengatasi kelemahan PAMIOL. Pengembangan AMIOnline ditujukan untuk lebih membantu SPM POLBAN dalam pelaksanaan Audit, evaluasi dan penilaian hasil audit. Melalui aplikasi

AMIOnline juga akan dikembangkan basis data yang dapat digunakan untuk proses analisa lebih lanjut dari hasil audit yang sudah rutin dilakukan. Tujuan lainnya agar aplikasi AMI bersifat dinamis dalam hal mengakomodasi kebutuhan database master (auditor, audite, penugasan) maupun basis data transaksional yang makin besar. Model data ini akan diimplementasikan menggunakan *Relasional Database Management System* (RDBMS) yang dapat mengakomodasi variasi model pertanyaan dan jawaban audit serta kemudahan perubahannya (revisi data, menambahkan data, menghapus pertanyaan ataupun jawaban audit) jika diperlukan. Selain itu, adanya formulasi penilaian (rumus terkait entri jawaban atau penilaian bertingkat) yang otomatis akan mereduksi kesalahan perhitungan dari auditor. Fleksibilitas AMIOnline dapat digunakan bagi kalangan lebih luas, tidak hanya AMI perguruan tinggi tetapi instansi umum lainnya. Hal ini dikarenakan adanya fasilitas disain template yang variatif sesuai kebutuhan pengguna [10].

2. METODE

Pengembangan perangkat lunak AMIOnline akan menggunakan pendekatan *waterfall* [11] yang mencakup tiga tahap besar yaitu :

(1) *Requirement analysis and definition* yaitu perumusan dan analisa penyebab masalah dari system yang berjalan. Analisa dilakukan terhadap aplikasi PAMIOL yang berjalan saat ini. Hasil analisa adalah teridentifikasinya kebutuhan (*requirements*) yang merupakan perbaikan dari aplikasi *existing.Requirement* yang terdefinisi akan dipenuhi oleh aplikasi AMIOnline yang akan dibuat. Hasil dari tahap ini digambarkan dalam *tools* : *business process, business rule, dan Software Requirement Specifications* (SRS).

(2) *System and software design*, yaitu tahap yang dilakukan untuk menjelaskan gambaran dari solusi konseptual atas hasil analisa di phase sebelumnya. Tujuan dari tahap ini adalah terbentuknya arsitektur dari aplikasi dan hubungan setiap komponen arsitektur untuk aplikasi yang akan dikembangkan (AMIOnline). Hasil dari tahapan ini dimodelkan menggunakan *tools: Sequence Diagram, Entity Relationship Diagram, Class Diagram*, dan rancangan *interface* atau *mockup* ;

(3) *Implementation and unit testing* berupa realisasi dari setiap rancangan yang telah dibuat menjadi unit program dengan menggunakan bahasa pemrograman PHP. Setiap hasil implementasi yang mengha-silkan unit program langsung dilakukan pengujian terhadap unit program tersebut. Hasil dari tahap ini adalah unit program yang telah diuji.

2.1 Analisis dan identifikasi *Requirement*

Phase analisa dan identifikasi kebutuhan dari perangkat lunak yang akan dikembangkan, dilakukan melalui analisa mendalam terhadap bisnis proses maupun aturan bisnis dari aplikasi PAMIOL yang berjalan di POLBAN. Analisa juga dilakukan terhadap system sejenis yang dijadikan acuan pengembangan, dalam hal ini perilaku penilaian program studi berdasarkan borang akreditasi dari BAN PT akan menjadi media pembanding nya. Dari hasil analisa atas 2 system di atas, maka dilakukan evaluasi.

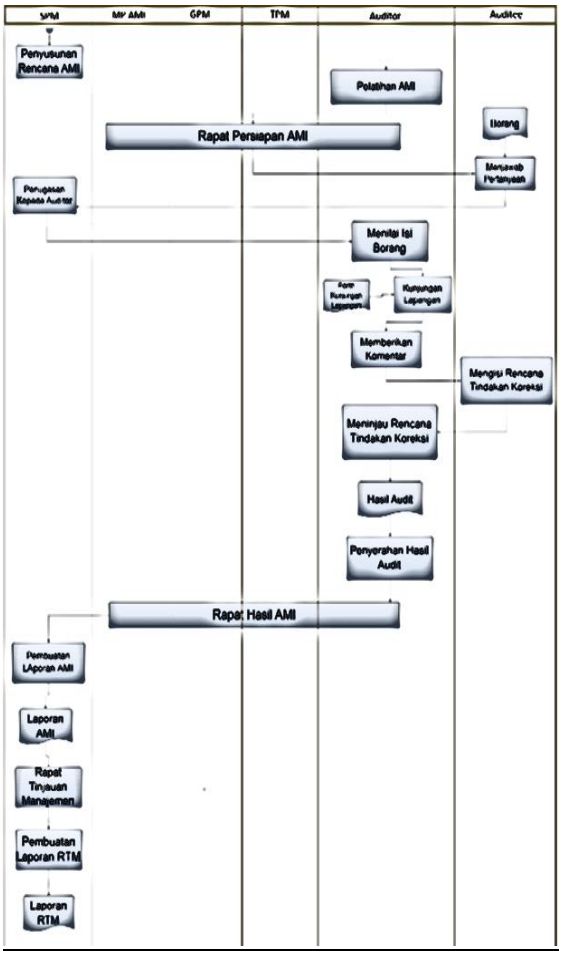
2.1.1. AMI di Polban

Gambar 1 berikut ini merupakan hasil analisis sistem operasional dan prosedur AMI di Polban yang melibatkan SPM, MP AMI, GPM, TPM, Auditee dan Auditor dengan tugas dan fungsinya masing-masing.

Aturan bisnis yang diberlakukan di SPM Polban adalah :

- Calon auditor wajib mengikuti pela-tihan AMI.

- Untuk menjadi auditor wajib dinyatakan lulus dari tes pertama dan tes kedua.
- Jawaban yang diisikan oleh auditee harus merupakan jawaban yang sesuai dengan keadaan nyata.
- Auditor yang dipilih harus berasal dari luar prodi atau unit bagian yang diaudit.



Gambar 1. SOP AMI Polban

- Satu auditee diaudit oleh 2 orang auditor.
- Nilai kualitatif dinilai berdasarkan *expert judgement* auditor.
- Penilaian yang bersifat kuantitatif hasilnya seharusnya benar dan sama untuk semua auditor yang menilai elemen penilaian tersebut.

2.1.2 *Scoring* Borang 3A BAN PT

AMI Polban dengan bantuan PAMIOL mengaudit program studi dengan versi borang 3A BAN PT untuk Diploma. Analisis berdasarkan borang 3A dan Buku VI yang menjadi panduan skor [8], diperoleh tiga tipe jawaban audit berikut ini.

(1). Tekstual

Jenis pertanyaan dengan jawaban deskripsi teks bersifat kualitatif dan membutuhkan *expert judgement* untuk penilaiannya.

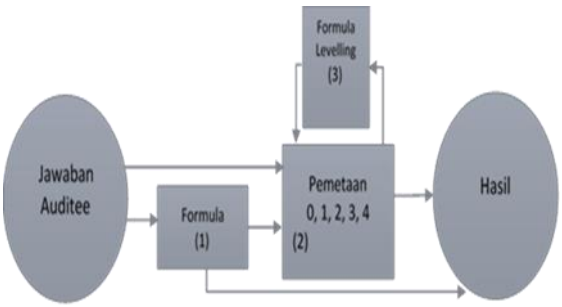
(2). Set

Jenis pertanyaan yang membutuhkan inputan beberapa butir skor dari auditor dengan rentang (0-4), kemudian skor subbutir dikalkulasi dengan satu rumus tertentu untuk mendapatkan skor akhirnya.

(3). Numerik

Jenis pertanyaan dengan jawaban data kuantitatif dan membutuhkan cara penghitungan skor berdasarkan formula atau pemetaan seperti alur diagram pada Gambar 2. Proses kalkulasi skor membutuhkan istilah berikut:

- Formula : Rumus untuk menghitung skor jawaban dari inputan auditee.
- Pemetaan : inputan auditee yang dipetakan kepada (0, 1, 2, 3, 4).
- Formula berjenjang: Rumus yang dipakai untuk mengkalkulasi jawaban dari inputan auditee setelah melewati proses pemetaan.



Gambar 2. Diagram Skor Numerik Berjenjang

Berdasarkan diagram pada Gambar 2, terdapat lima tipe kalkulasi yaitu menghitung data inputan auditee dengan menggunakan :

(1) suatu formula dan menjadi skor akhir,

(2) pemetaan dan menjadi skor akhir,
(3) formula pada awalnya kemudian dipetakan sebagai skor akhir,
(4) pemetaan kemudian menggunakan formula menjadi skor akhir, atau
(5) kombinasi formula, pemetaan, formula atau pemetaan, formula, pemetaan kemudian menjadi skor akhir.

2.1.3 Kekurangan PAMIOL

Analisis PAMIOL sebagai aplikasi untuk AMI dicantumkan secara ringkas menjadi lima kekurangan seperti dalam Tabel 1.

Tabel 1. Analisis Aplikasi PAMIOL

Kekurangan	Deskripsi Permasalahan yang Timbul
Penyimpanan beberapa data dengan link data	Menyulitkan proses pemilahan data.
Data jawaban disimpan dalam bentuk blob	Menyulitkan proses pemilahan dan analisa data dan data yang disimpan sangat besar.
Beberapa data di menu master tidak dapat di-hapus	Banyak data yang tidak terpakai di dalam *database*. Kemudian user yang sudah tidak aktif lagi tetap bisa login sehingga membahaya-kan keamanan aplikasi.
Penghitungan nilai numerik masih dilakukan diluar aplikasi	Proses penilaian menjadi lama dan rawan terjadi kesalahan input nilai ke dalam aplikasi.
Struktur model data yang kurang sesuai	Kurang efisien karena ada beberapa field yang diulang.

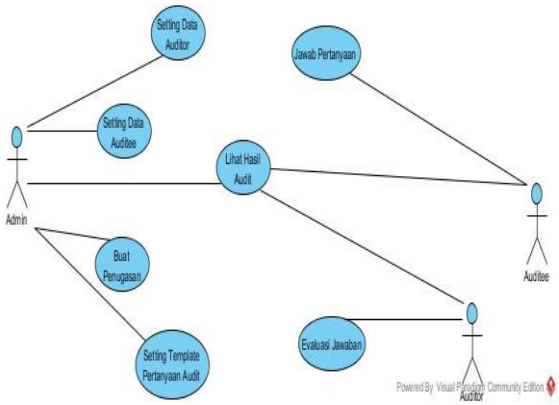
Khusus kekurangan dari sisi model data dari aplikasi PAMIOL dapat diuraikan lebih rinci pada Tabel 2 berikut ini.

Tabel 2. Analisis Model Data

ID Evaluasi	EV-05
Analisis Terkait	Struktur model data yang kurang sesuai.
Deskripsi	
Terlalu banyak redundansi data pada struktur data yang diimplementasikan. Model data pamiol hanya berisi kumpulan entity yang beberapa fieldnya diulang di entity lain.	
Kelebihan	
Waktu perancangan lebih sedikit karena model data yang digunakan pamiol hanya kumpulan entity.	
Kekurangan	
Kurang efisien karena ada beberapa field yang diulang di entity lain. Selain itu pengolahan query pada saat pembangunan aplikasi menjadi sulit.	
Solusi	
Merancang *database* baru dengan struktur model data yang lebih terstruktur.	

2.1.4 Evaluasi

Hasil evaluasi dari butir-butir analisis sistem AMI, dirangkum dalam *use case* diagram [12] sesuai Gambar 3.



Gambar 3. *Use case* AMI Polban

2.2 Disain

Rancangan pengembangan fitur dan data untuk mengatasi kekurangan PAMIOL yaitu pada aplikasi AMIOnline diberikan di Tabel 3 berikut ini.

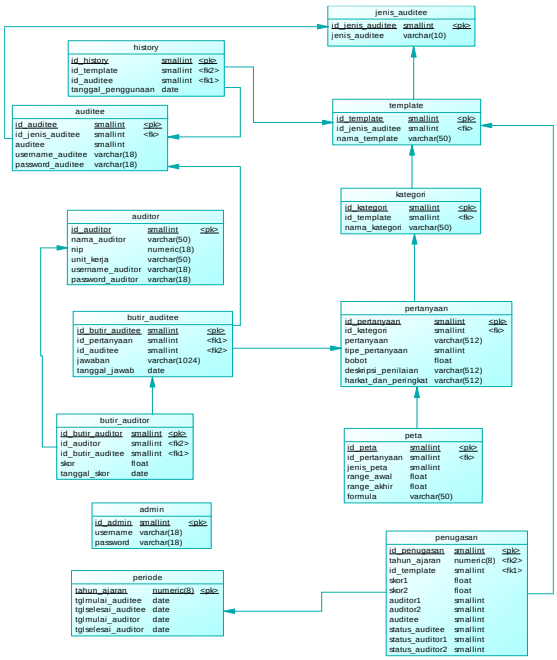
Tabel 3. Fitur yang dikembangkan

Nama Fitur	Deskripsi
Setting Data	Fitur ini berfungsi untuk

Auditor	mencatat data auditor yang ada.
Setting Data Auditee	Fitur ini berfungsi untuk mencatat data auditee yang ada.
Setting Template Pertanyaan Audit	Fitur ini berfungsi untuk membuat template pertanyaan audit yang nantinya dipakai oleh auditee dan auditor.
Setting Penugasan	Fitur ini berfungsi untuk mencatat data penugasan audit.
Jawab Pertanyaan	Fitur ini berfungsi untuk meng-*entry* jawaban auditee.
Evaluasi Jawaban	Fitur ini berfungsi untuk meng-*entry* nilai kualitatif dan mengkalkulasikan nilai kuantitatif. Fitur ini dibuat untuk membantu auditor dalam hal penilaian audit.
Lihat Hasil Audit	Fitur ini berfungsi untuk melihat hasil audit yang telah dilakukan. Fitur ini dibuat berdasarkan tujuan no 2 yaitu untuk penelusuran ulang hasil audit.

2.2.1 Disain Basis Data

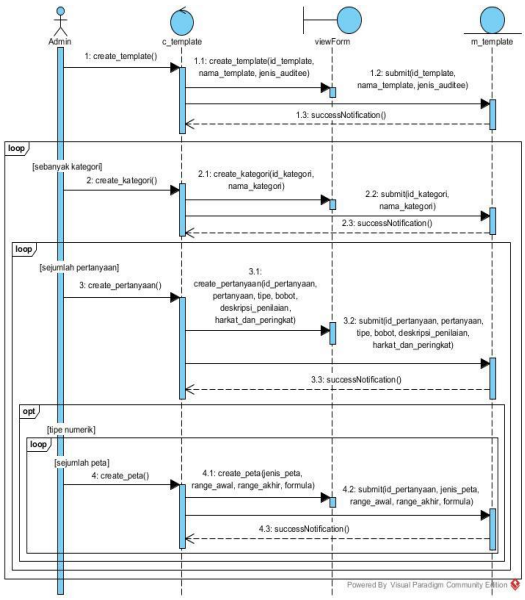
Rancangan *Physical Data Model* (model data fisik, PDM) AMIOnline berdasarkan RDBMS diberikan berupa diagram pada Gambar 4, sebagai berikut.



Gambar 4. PDM AMIOnline

2.2.2 Disain Perilaku Aplikasi

Perilaku dari aplikasi yang akan dikembangkan, dapat dimodelkan menggunakan *Sequence Diagram* (SeqD). Dalam SeqD dijelaskan interaksi antar objek [12] yang terjadi dalam aplikasi untuk memenuhi *UseCase* yang terdefinisi dari hasil analisa di sub bab sebelumnya. Sebagai contoh diberikan SeqD untuk perilaku penambahan template pertanyaan audit (*AddTemplate*). Proses-nya adalah admin masuk ke halaman tambah template dan mengisikan data template di ruang yang tersedia untuk membuat template baru, kemudian menambahkan kategori dan pertanyaan. SeqD *AddTemplate* dan deskripsi dari tahapan penambahan *Template* pertanyaan audit diuraikan pada Gambar 4 dan Tabel 4 di halaman berikut.



Gambar 5. SeqD *AddTemplate*

Tabel 4. Deskripsi SeqD *AddTemplate*

ID *Sequence Diagram*	SD-02
Nama *Sequence Diagram*	Add template
ID *Use Case*	UC-ADM-02
No. Requirement	REQ RP-01, REQ CP-01, REQ CP-02, REQ CP-03, REQ CP-04, REQ DP-01
Deskripsi	Menggambarkan alur proses menambahkan template, kategori dan pertanyaan
Logika proses	

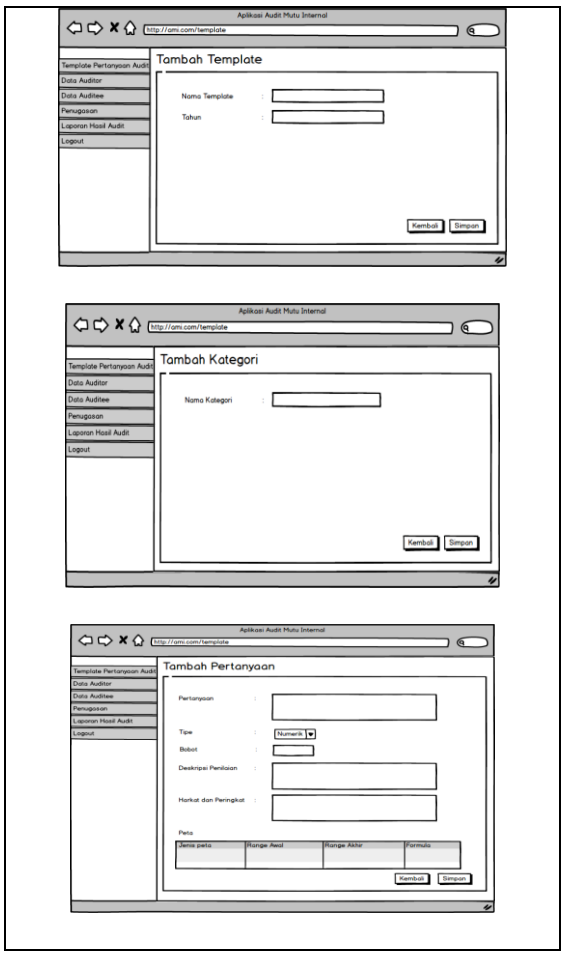
```
BEGIN

        Read(keyboard) vid_template
        Read(keyboard) vnama_template
        Read(keyboard) vtahun
        //memasukkan data inputan ke dalam database
        INSERT INTO template (id_template, nama_template, tahun)
        VALUES ('vid_template', 'vnama_template', 'vtahun')

LOOP sebanyak kategori
        Read(keyboard) vid_kategori
        Read(keyboard) vnama_kategori
        //memasukkan data inputan ke dalam database
        INSERT INTO template (id_kategori, nama_kategori)
        VALUES ('vid_kategori', 'vnama_kategori')

        LOOP sebanyak pertanyaan
                Read(keyboard) vid_pertanyaan'
                Read(keyboard) vpertanyaan
                Read(keyboard) vtipe
                Read(keyboard) vbobot
                Read(keyboard) vdeskripsi_penilaian
                Read(keyboard) vharkat_dan_peringkat
        //memasukkan data inputan ke dalam database
        INSERT INTO template (id_pertanyaan, pertanyaan, tipe, bobot, deskripsi_penilaian, harkat_dan_peringkat)
        VALUES ('vid_pertanyaan', 'vpertanyaan', 'vtipe', 'vbobot', 'vdeskripsi_penilaian', 'vharkat_dan_peringkat')
        ENDLOOP
ENDLOOP

END
```

2.2.3 Disain Antar Muka

Untuk mengimplementasikan design perilaku di sub bab sebelumnya, maka dapat dimodelkan sebuah rancangan antar muka atau *user interface* (UI) yang akan dijadikan media komunikasi antara pengguna (*User*) dengan Aplikasi yang akan dikembangkan.

Dari perilaku pembuatan data pertanyaan audit baru yang diuraikan di sub bab sebelumnya, teridentifikasi bahwa agar user (*admin* AMIOnline) dapat meng entry kan pertanyaan di aplikasi yang akan dikembangkan, maka dibutuhkan 3 buah UI. Ketiga buah UI yang akan dikembangkan, memiliki fungsi untuk menampilkan form tambah template, mengentry kategori pertanyaan, serta UI untuk entry pertanyaan. Ketiga UI tersebut dimodelkan dalam Gambar 6 di halaman berikut.

Semua *Usecase* yang terdefinisi dalam hasil evaluasi PAMIOL di gambar 3, harus dilakukan pemodelan data, pemodelan perilaku maupun pemodelan *User Interfece* seperti yang dilakukan untuk *AddTemplate*.



Gambar 6. Tiga UI Form *AddTemplate*

3. HASIL DAN PEMBAHASAN

AMIOnline telah dibangun dengan baik menggunakan framework *Code*

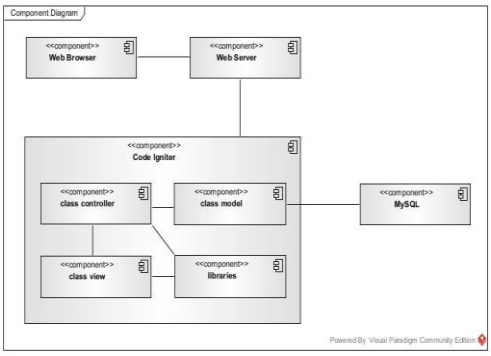
Igniter yang memerlukan pengaturan dan modifi-kasi untuk konfigurasi aplikasi, dan database.

Dari rancangan yang telah dibuat, dilakukan implementasi berupa :

- struktur komponen aplikasi ditunjukkan dengan *Component Diagram*,
- infrastruktur sistem ditunjukkan dengan *deployment diagram*,
- peta aplikasi yang ditunjukkan dengan peta web.

3.1 Struktur Komponen Sistem

Component Diagram (Diagram Komponen, DK) dibawah ini digunakan untuk menunjukan keterkaitan antar komponen dari aplikasi. Komponen terdiri dari *web browser*, *web server*, *database server* dan *framework Code Igniter*, diilustrasikan pada Gambar 7 dan Tabel 5.



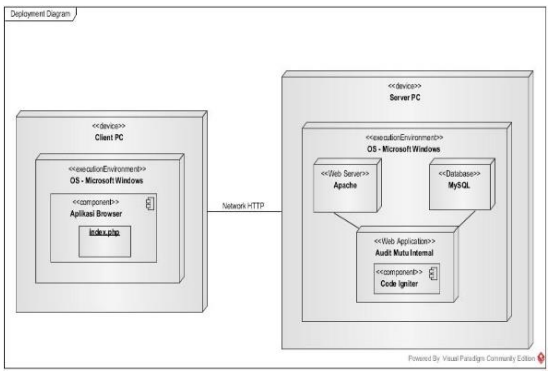
Gambar 7. DK Struktur Komponen Aplikasi

Tabel 5. Deskripsi Diagram Komponen

Nama Komponen	Deskripsi
Komponen *web browser*	Merupakan komponen yang memungkinkan pengguna untuk mengakses aplikasi ini.
Komponen *web server*	Merupakan komponen yang memungkinkan aplikasi dijalankan pada web browser.
Komponen *controller*	Merupakan komponen yang berisi class-class untuk menghubungkan antara logika pemrograman aplikasi dengan *user interface*.
Komponen *model*	Merupakan komponen yang berisi class-class untuk menghubungkan ke dalam *database*.
Komponen *view*	Merupakan komponen yang menghubungkan pengguna dengan aplikasi *(user interface).*
Komponen *database*	Merupakan komponen yang digunakan untuk menyimpan data aplikasi.
Komponen *libraries*	Merupakan komponen yang berisi fungsi-fungsi dari *framework CodeIgniter*.

3.2 Infrastruktur Sistem

Infrastruktur dari aplikasi yang dikembangkan dapat digambarkan melalui *Deployment Diagram* (DD) untuk yang menyatakan keterhubungan node *Client* dan server aplikasi [12]. Apa saja yang harus ada di sisi *client* dan yang harus ada di sisi *server application*, dideskripsi dalam uraian di Tabel 6. Hal ini menggambarkan infrastruktur fisik dan kedudukan sistem terhadap komponen-komponen pendu-kungnya.



Gambar 8. DD Infrastruktur sistem

Tabel 6. Deskripsi *Deployment Diagram*

Node	Deskripsi
Audit Mutu Internal	*Database* aplikasi Audit Mutu Internal.
Framework CodeIgniter	*Framework* dari aplikasi Audit Mutu Internal.
MySQL	*Database management system*, yang digunakan untuk penyimpanan data aplikasi.
Apache	*Database server* yang menyimpan dan mengirim halaman *database* dengan berupa informasi yang ada didalamnya menggunakan *protocol* HTTP.
Microsoft Windows	Sistem operasi dari aplikasi Audit Mutu Internal.
Aplikasi	Akses untuk melakukan

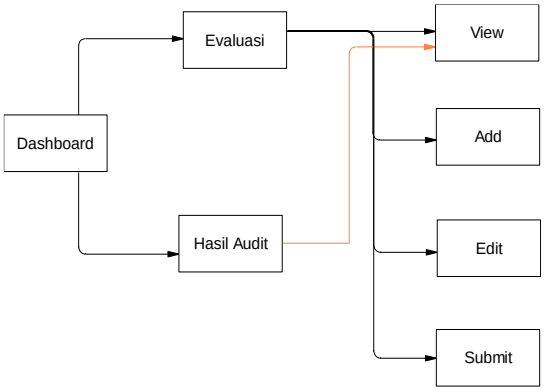
Node	Deskripsi
Browser	*request* dan respon dari PC
Server PC	*Device* yang menjadi tempat pemrosesan *request* dari *client* terhadap aplikasi Audit Mutu Internal.
Client PC	*Device* yang menjadi tempat mengirim *request* ke *server*.

3.3 Peta Aplikasi

Peta web atau struktur menu dari aplikasi yang dibuat, dibagi atas hak akses tiap pengguna. Tiga kelompok pengguna AMIOnline dibagi menjadi admin, auditee dan auditor. Sebagai contoh diberikan peta web auditor sebagaimana deskripsi pada Tabel 7 dan Gambar 9.

Tabel 7. Deskripsi Struktur Menu Auditor

Nama Menu	Deskripsi
Evaluasi	Halaman yang berisi data template pertanyaan audit yang sebelumnya telah dijawab oleh auditee. Pada halaman ini auditor dapat mengentry nilai, mengedit nilai, melihat nilai, dan submit nilai.
Hasil Audit	Halaman yang berisi daftar data hasil audit yang telah dilakukan beserta skor akhirnya. Pada halaman ini auditor dapat melihat isi hasil audit.
View	View adalah tombol untuk mengarahkan pada halaman yang digunakan untuk melihat detail dari suatu konten.
Add	Add adalah tombol untuk mengarahkan pada halaman yang digunakan untuk melakukan insert suatu konten.
Edit	Edit adalah tombol untuk mengarahkan pada halaman yang digunakan untuk memperbaiki/mengubah suatu konten.
Submit	Submit adalah tombol untuk mengarahkan pada halaman awal auditor. Setelah menekan tombol submit maka auditor tidak dapat mengedit skor lagi.



Gambar 9. Struktur Menu Auditor

Aplikasi AMIOnline telah diimplementasikan dan dipersiapkan untuk AMI SPM Polban mulai 2017. Dari uji coba dan pembahasan, aplikasi AMIOnline mampu :

- mencatat data auditor, auditee, penugasan, jawaban, dan skor yang nantinya dipergunakan dalam proses pengisian dan penilaian audit,
- membantu auditee dalam mencatat jawaban dan menyimpannya ke dalam *database*,
- membantu auditor untuk melakukan perhitungan nilai hasil jawaban butir audit yang bertipe numerik. Untuk jawaban bertipe tekstual dan set aplikasi mampu mengolah hasil penilaian auditor berdasarkan *expert judgement*,
- membantu staff pendukung SPM dalam meng-*entry* data template pertanyaan atau borang beserta cara perhitungannya yang kemudian digunakan untuk proses audit,
- menghitung total skor audit secara otomatis,
- menampilkan data jawaban auditee beserta skornya untuk ditelusuri ulang secara *historical* setelah batas waktu penilaian oleh auditor terlampaui.

Selanjutnya beberapa hal yang masih diperlukan untuk pengembangan aplikasi AMIOnline adalah :

- Pembuatan fitur untuk penyimpanan data pendukung atau lampiran sebagai artifak dari jawaban auditee. Penamaan file data pendukung

sebaiknya dibuat format khusus sehingga dapat mewakili sumber file tersebut.

- Pembuatan fitur untuk mendukung pelaporan seperti laporan penugasan auditee dan auditor, laporan hasil/nilai audit, laporan pelaksanaan audit dan laporan lain yang bersifat grafis untuk memudahkan monitoring pelaksanaan audit.
- Aplikasi dapat dikembangkan dengan kemampuan menganalisa hasil audit sehingga memudahkan proses rapat tinjauan manajemen (RTM) yang akan dilaksanakan.

4. KESIMPULAN

Perangkat lunak audit mutu AMIOnline yang dikembangkan dapat digunakan untuk merekam borang atau template pertanyaan audit. Sifatnya yang dinamis memungkin-kan untuk mengakomodasi perubahan instrumen pertanyaan pada borang bila diperlukan. Penerapan RDBMS membuat aplikasi ini menampung data jawaban auditee sehingga dapat ditelusuri atau digunakan kembali pada saat dibutuhkan. Dengan variasi model-model pertanyaan audit yang didefinisikan pada template maka aplikasi ini dapat digunakan untuk membatu proses audit mutu di lingkungan perguruan tinggi atau instansi lainnya.

5. REFERENSI

[1]. Permenristekdikti No. 62 , 2016.
[2]. Permenristekdikti No. 44, 2015.
[3]. D. J. P. Tinggi., 2006 "*Audit Akade-mik dalam Perguruan Tinggi*",Jakarta.
[4]. Permenristekdikti No. 32, 2015.
[5]. Permenristekdikti No. 61, 2016.
[6]. Piagam SPM, SK Dir Polban No. 0139/R8.R/OT/2011.
[7]. Dokumen Mutu SPM Polban, 2015.
[8]. BAN-PT, (2010) Buku IV *Panduan Pengisian Instrumen*, Buku VI *Pedoman Penilaian Borang*.
[9]. Laporan Akhir Tahun SPM, 2015
[10].Ade Chandra Nugraha, Nurjannah Syakrani , 2016, *Aplikasi AMI Online, studi kasus Politeknik Negeri Bandung*, Laporan UPPM.
[11]. I. Sommerville, 2011, *Software Engineering Ninth Edition*, Boston, Massachusetts: Pearson Education, Inc.
[12]. Larman, C, (2004). *Applying UML and Patterns*, Addison Wesley Professional.

Ucapan Terima Kasih

Terima kasih sebesar-besarnya penulis ucapkan kepada UPPM Polban atas pendanaan hibah PkM 2016; SPM Polban sebagai mitra PkM; Kontributor : An Nisa, Chandriawan dan Rizki; serta reviwer