

PENGEMBANGAN *LEARNING MANAGEMENT SYSTEM* MENGGUNAKAN FRAMEWORK CODEIGNITER DAN ANGULARJS DI PT. XYZ

Denti Denita Putri

Program Studi Teknik Informatika, Politeknik Negeri Jakarta, Jl. Prof. Dr. G.A Siwabessy, Kampus Baru UI Depok, 16424, Indonesia

E-mail: dentidenitap@gmail.com

Abstract

Information technology is very common in this globalization era. Information technology has been widely used by communities, especially in the company to assist the activities in accordance with applicable company's business processes, one of which is PT. XYZ. PT. XYZ is a company that has a main program that is organizing training and learning, and assessment of employee competence. It uses technology to make education and training activities by applying the Learning Management System (LMS). Currently, the company implements an LMS that accommodates only a portion of the company's business processes. Thus, it needs further development to assist in improving the performance effectiveness and efficiency in accordance with the business processes running in the company. LMS is used to simplify the master data processing needed to support the training and learning, one of them is the material master data. The method to develop the LMS is SDLC method with incremental model. The LMS is implemented using Bootstrap, AngularJS, and Codeigniter framework, also MySQL for the Database Management System (DBMS). The test results show that the LMS system has been run in accordance with the demands and needs of the company making it more effective and efficient in the implementation of learning and training employees.

Keywords: *Learning Management System, SDLC Incremental Model, Bootstrap, AngularJS, Codeigniter*

Abstrak

Teknologi informasi merupakan hal yang sangat umum pada era globalisasi ini. Teknologi informasi telah banyak digunakan oleh masyarakat khususnya di dalam perusahaan untuk membantu kegiatan sesuai dengan proses bisnis yang diterapkan perusahaan, salah satunya adalah PT. XYZ. PT. XYZ merupakan perusahaan yang memiliki program utama yaitu penyelenggaraan pendidikan dan pelatihan serta *assessment* kompetensi pegawai. PT. XYZ memanfaatkan teknologi untuk melakukan penyelenggaraan pendidikan dan pelatihan dengan menerapkan *Learning Management System* (LMS). Saat ini, LMS yang diterapkan perusahaan hanya mengakomodasi sebagian dari proses bisnis perusahaan sehingga perlu adanya pengembangan lebih lanjut untuk membantu dalam meningkatkan efektivitas dan efisiensi kinerja yang dilakukan sesuai dengan proses bisnis yang berjalan di perusahaan. LMS ini digunakan untuk mempermudah dalam pengolahan data-data master yang dibutuhkan dalam menunjang kegiatan pelatihan dan pembelajaran salah satunya adalah data master materi. Metode pengembangan LMS yang digunakan adalah metode SDLC dengan model inkremental. Implementasi LMS menggunakan framework Bootstrap, AngularJS, dan Codeigniter, serta *Database Management System* (DBMS) MySQL. Hasil pengujian sistem menunjukkan bahwa LMS telah berjalan sesuai dengan permintaan dan kebutuhan perusahaan sehingga lebih efektif dan efisien dalam pelaksanaan kegiatan pembelajaran dan pelatihan para pegawainya.

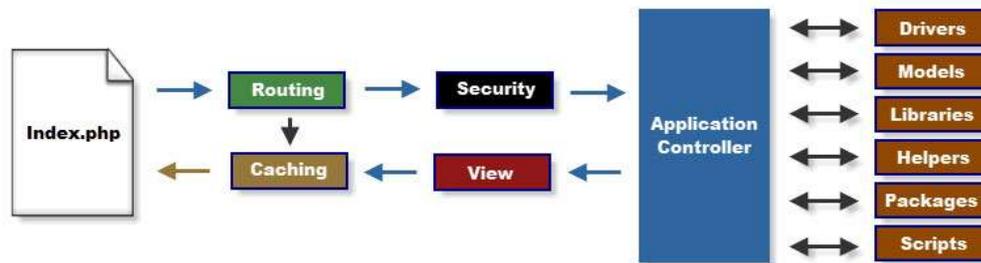
Kata Kunci: *Learning Management System, SDLC Incremental Model, Bootstrap, AngularJS, Codeigniter*

1. Pendahuluan

Teknologi informasi merupakan hal yang sangat umum pada era globalisasi ini. Penggunaan teknologi informasi ditunjang dengan internet saat ini telah banyak dimanfaatkan oleh masyarakat untuk membantu dalam melakukan berbagai kegiatan seperti pengolahan data untuk memperoleh informasi

yang dibutuhkan. Begitu juga penggunaan teknologi informasi di dalam perusahaan yang banyak diaplikasikan untuk membantu kegiatan sesuai dengan proses bisnis yang diterapkan perusahaan, salah satunya PT. XYZ.

PT. XYZ merupakan perusahaan yang memiliki program utama yang ditawarkan yaitu penyelenggaraan pendidikan dan pelatihan, serta *assess-*



Gambar 1. Cara Kerja Framework CI

ment kompetensi pegawai. PT. XYZ memanfaatkan teknologi untuk melakukan penyelenggaraan pendidikan dan pelatihan dengan menerapkan *Learning Management System* (LMS).

LMS merupakan integrasi dari sistem-sistem yang ada di PT. XYZ dan merupakan pengembangan dari LMS sebelumnya. *Learning Management System* (LMS) yang diterapkan oleh PT. XYZ saat ini hanya mengakomodasi sebagian dari proses bisnis perusahaan. Selain itu, pada *Learning Management System* (LMS) yang ada saat ini masih terdapat *error* dan belum memiliki dokumentasi lengkap yang dibutuhkan terkait pengadaan dan penyelenggaraan pelatihan.

Oleh karena itu, perlu adanya pengembangan *Learning Management System* (LMS) untuk membantu dalam meningkatkan efektivitas dan efisiensi kinerja yang dilakukan sesuai dengan proses bisnis yang berjalan di perusahaan. Pengembangan LMS dilakukan dengan adanya penambahan fitur-fitur dan penyempurnaan proses pelaksanaan pelatihan dan pembelajaran.

Metode pengembangan LMS yang digunakan adalah metode *Systems Development Life Cycle* (SDLC) dengan model inkremental. Implementasi LMS berbasis web ini menggunakan *framework* Bootstrap, AngularJS dan Codeigniter serta *Database Management System* (DBMS) MySQL.

LMS berbasis web ini dikembangkan untuk mempermudah perusahaan dalam mengolah data-data *master* terkait pengadaan kegiatan pelatihan dan pembelajaran seperti data *master* materi dan data *master* lain yang terkait. Selain itu, LMS juga mempermudah perusahaan dalam melakukan pelaksanaan pembelajaran seperti penjadwalan, dan pengaturan kebutuhan-kebutuhan seperti pengaturan kebutuhan lab yang diperlukan untuk proses pembelajaran.

Learning Management System (LMS)

Menurut Amiroh [1], LMS atau *Course Management System* (CMS) juga dikenal sebagai *Virtual Learning Environment* (VLE) merupakan aplikasi perangkat lunak yang digunakan oleh kalangan

pendidik, baik universitas/ perguruan tinggi dan sekolah sebagai media pembelajaran *online* berbasis internet (*e-learning*). Dengan menggunakan LMS, dosen/guru/instruktur dapat mengelola program/kelas dan bertukar informasi dengan siswa. Selain itu, akses terhadap materi pembelajaran yang berlangsung dalam kurun waktu yang telah ditentukan juga dapat dilakukan.

Fitur-fitur yang tersedia dalam LMS untuk institusi pendidikan adalah pengelolaan hak akses pengguna (*user*), pengelolaan *courses*, pengelolaan bahan ajar (*resource*), pengelolaan aktivitas (*activity*), pengelolaan nilai (*grades*), menampilkan nilai (*score*) dan transkrip, serta pengelolaan visualisasi *e-learning*, sehingga bisa diakses dengan *web browser*.

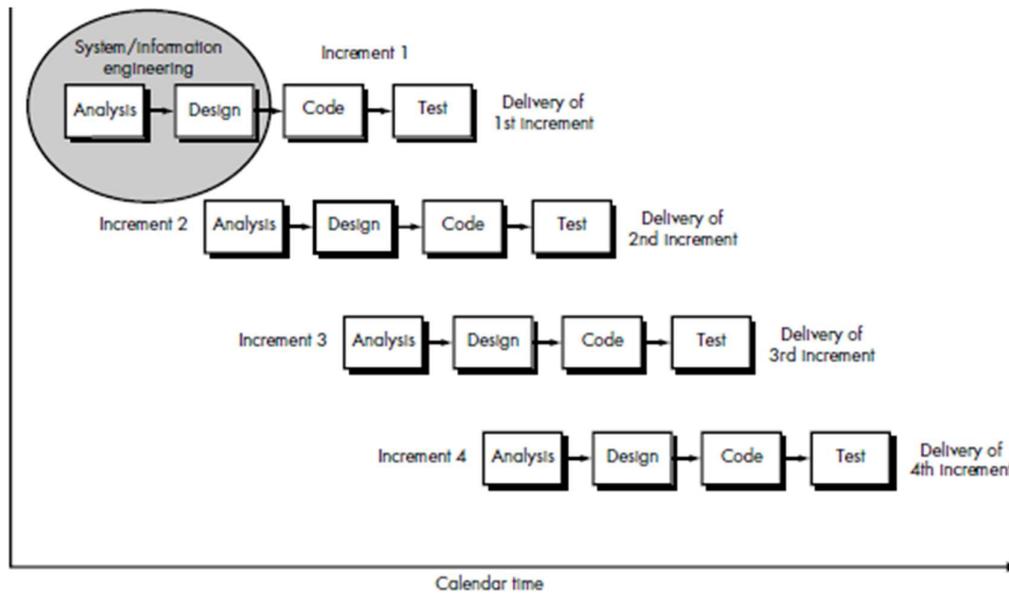
Jadi, *Learning Management System* membuat siswa dan guru/dosen masuk ke dalam ruang “kelas digital” untuk saling berinteraksi (berdiskusi, mengerjakan kuis *online*, dsb) serta mengakses materi-materi pembelajaran dimana saja dan kapan saja selama terkoneksi dengan internet.

Framework Bootstrap

Menurut Zakir [2], Bootstrap merupakan sebuah *framework* yang dapat menyelesaikan permasalahan dalam mendesain *web*. Slogan dari *framework* ini adalah “*sleek, intuitive, and powerful front-end framework for faster and easier web development*”, yang berarti *user* dapat mendesain sebuah website dengan lebih rapi, cepat dan mudah. Selain itu Bootstrap juga *responsive* terhadap banyak *platform*, artinya tampilan halaman website yang menggunakan Bootstrap ini akan tampak tetap rapi, baik versi *mobile* maupun *desktop*.

Framework AngularJS

Angular JS adalah sebuah *front-end frameworks* generasi baru yang mampu membantu meningkatkan produktifitas pengembangan aplikasi serta mempermudah perawatan dan *testing* aplikasi *web* [3]. Angular JS pertama kali dikembangkan oleh Misko Hvery dan Adam Abrons pada tahun



Gambar 2. Proses Model Inkremental

2009. Angular JS mengadopsi arsitektur *Model View Controller* (MVC) dan *Model View View Model* (MVVM) dimana kedua arsitektur ini digunakan untuk mempermudah *maintenance* kode program dan proses *testing*. Angular JS bekerja dengan membaca *tag* HTML yang diberi *custom tag* milik angular, kemudian *framework* Angular JS menerjemahkan *tag* tersebut sebagai sebuah *directive* yang dapat ditautkan pada *input* atau *output* program.

Framework CodeIgniter

Menurut Rahmadiansyah & Irawan [4], CodeIgniter (CI) merupakan sebuah *framework* pemrograman web dengan menggunakan bahasa PHP. *Framework* ini ditulis dengan menggunakan bahasa php versi 4 dan versi 5 oleh Rick Ellislab yang menjadi CEO Ellislab, Inc. dan dipublikasikan dengan lisensi di bawah Apache/BSD *Open Source*. Di dalam CI terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrogram dalam mengembangkan aplikasinya. Berikut merupakan cara kerja *framework* CI [5].

Konsep pada *framework* CodeIgniter yaitu MVC (*Model View Controller*). MVC adalah sebuah pola rancangan yang memungkinkan para pengembang memisahkan kode program kedalam tiga kelompok, yaitu *model*, *view* dan *controller* [5].

Model, bagian ini berisi kode yang digunakan untuk koneksi dan mengakses *database*. *View*, berisi kode-kode HTML dan PHP yang digunakan untuk menampilkan informasi ke layar *browser*. Biasanya kode-kode yang berhubungan dengan for-

mat tampilan, misalnya huruf, form, warna dan lain-lain. *Controller*, berisi kode-kode *script* yang menjalankan fungsi aturan bisnis aplikasi dan menjadi perantara antara *Model* dan *View* serta seluruh sumber daya yang dibutuhkan untuk memproses permintaan layanan HTTP untuk ditampilkan dalam halaman *web* [4].

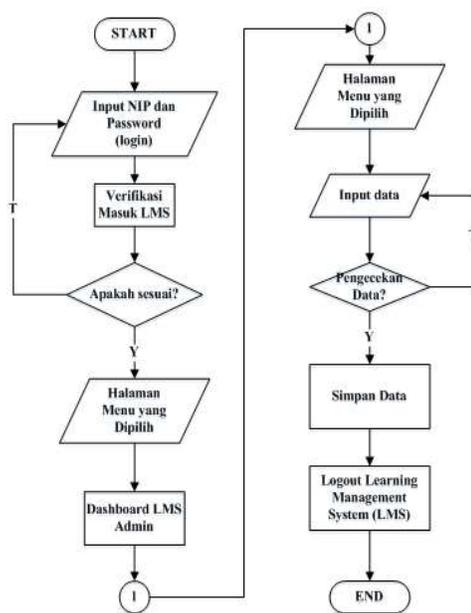
MySQL

Menurut Supono & Putratama [6], MySQL adalah sistem manajemen *database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system* (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, andal dan mudah digunakan.

2. Metode

Dalam membangun suatu aplikasi atau sistem tidak akan terlepas dari tahapan-tahapan dalam pembuatannya. Tahapan dalam pembuatan ini berbeda-beda, sesuai dengan model pengembangan yang dipilih.

Menurut Al Fatta [7], beberapa ahli membagi proses-proses pengembangan sistem, ke dalam sejumlah urutan yang berbeda-beda. Tetapi tetap mengacu pada proses-proses standar yaitu analisis, desain, implementasi dan pemeliharaan. Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang dikenal dengan *Systems Development Life Cycle* (SDLC) yang meru-



Gambar 3. *Flowchart Learning Management System (LMS)*

pakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain. SDLC meliputi fase-fase sebagai berikut: Identifikasi dan seleksi proyek, Inisiasi dan perencanaan proyek, Analisis, Desain, Implementasi, dan Pemeliharaan.

Pada pengembangan *Learning Management System (LMS)*, model yang digunakan dalam metode SDLC adalah Model Inkremental. Model inkremental menghasilkan serangkaian penambahan, yang disebut inkremen, yang memberikan fungsionalitas yang semakin baik kepada pelanggan saat masing-masing inkremen itu disampaikan.

Model penambahan sedikit demi sedikit (inkremental) menggabungkan elemen-elemen aliran proses *linier* dan *paralel*. Ketika model inkremental digunakan, hasil pada tahap pertama seringkali berupa produk inti (*core product*). Itu berarti spesifikasi kebutuhan dasar telah ada, tetapi fitur-fitur pelengkap/tambahan (beberapa diantaranya diketahui, beberapa lainnya tidak) tetap belum terselesaikan. Produk inti ini digunakan oleh pelanggan (atau dalam bentuk evaluasi). Sebagai hasil dari penggunaan dan/atau evaluasi, suatu rencana pengembangan untuk inkrementasi berikutnya dikembangkan. Perencanaan ini berisi perencanaan modifikasi produk untuk menghasilkan produk yang lebih sesuai dengan kebutuhan pengguna dan juga berisi perencanaan untuk mengembangkan fitur-fitur dan fungsionalitas-fungsionalitas tambahan [8].

Proses inkremental berfokus pada pengiriman produk yang bersifat operasional pada setiap tahapan. Proses ini diulang mengikuti pengiriman produk yang dihasilkan dari tiap-tiap tahap hingga produk yang bersifat lengkap (produk *final*) dihasilkan di bagian akhir proses pengembangan perangkat lunak. Dalam hal ini, model proses inkremental berfokus pada pengiriman produk operasional pada setiap tahap. Produk versi awal seringkali merupakan versi yang relatif lebih sederhana dari produk *final*, namun produk-produk ini langsung dapat digunakan untuk melayani kebutuhan pengguna dan juga dapat digunakan untuk menyediakan kerangka kerja untuk dilakukannya evaluasi selanjutnya oleh pengguna.

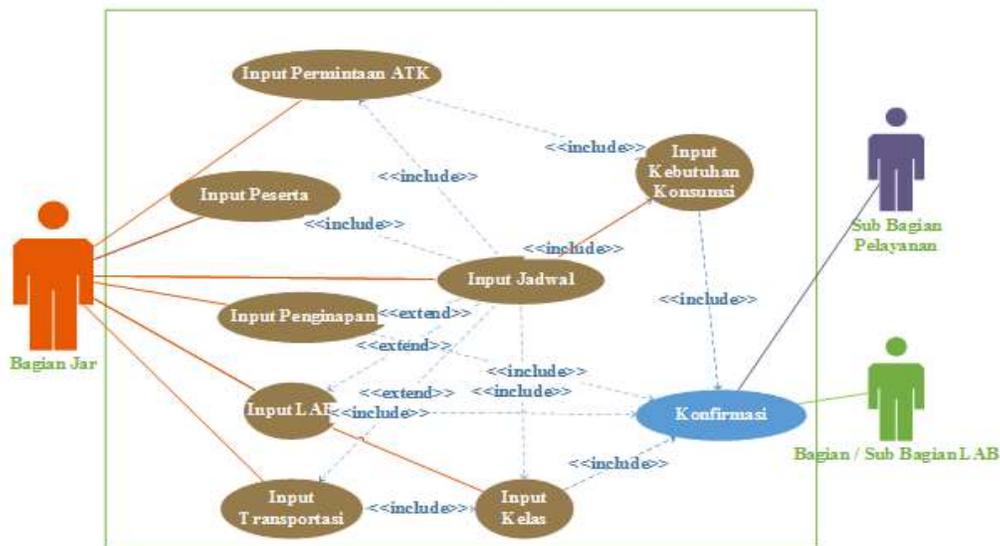
Pembangunan perangkat lunak dengan model inkremental sangat berguna ketika *staff* (pekerja) tidak tersedia untuk implementasi lengkap dengan batas waktu bisnis yang telah ditetapkan untuk proyek tersebut. Produk tahap yang lebih awal dapat diimplementasikan menggunakan jumlah orang yang lebih sedikit. Jika produk inti (*core product*) dapat diterima oleh pelanggan/pengguna, maka penambahan *staff* (jika diperlukan) dapat dilakukan untuk menyelesaikan produk pada tahap-tahap berikutnya. Selain itu, inkremental dapat direncanakan dengan mempertimbangkan risiko-risiko teknis tertentu.

3. Hasil dan Analisis

Proses bisnis LMS yang ada di PT. XYZ merupakan dasar yang dijadikan acuan untuk melakukan implementasi LMS. Pertama, *login* dilakukan ke dalam LMS dengan memasukkan *username* dan *password* yang telah ditentukan, kemudian sistem akan melakukan verifikasi ke *database*. Pada halaman *dashboard (index)* LMS, terdapat banyak menu pada bagian sebelah kiri (*sidebar*) yang dapat dipilih. Ketika menu dipilih, sistem akan melakukan pengecekan hak akses menu.

Menu-menu pada LMS dikelompokkan menjadi tiga bagian yaitu menu *master*, menu penyelenggaraan pembelajaran dan menu pengaturan. Menu-menu tersebut dapat ditambahkan, diubah ataupun dihapus apabila memiliki hak akses sebagai *superadmin*. Pada halaman utama menu (modul) yang dipilih, sistem akan menampilkan data-data modul terkait. Aksi yang dapat dilakukan adalah penambahan, pengubahan dan hapus data yang terdapat pada setiap modul baik untuk data *master* maupun data *detail* penjadwalan lab.

Analisa dan perancangan *Learning Management System (LMS)* akan dijelaskan dengan menggunakan bagan-bagan atau diagram seperti *Flowchart*, *Use case Diagram*, *Class Diagram*, dan *Entity Relationship Diagram (ERD)*.



Gambar 4. Use Case Diagram Learning Management System (LMS)

Flowchart atau disebut juga dengan diagram alir digunakan pada perancangan LMS untuk menggambarkan alur di dalam program secara logika [9]. Berikut merupakan *flowchart Learning Management System (LMS)* di PT. XYZ.

Gambar 3 merupakan *flowchart Learning Management System* secara keseluruhan. *Flowchart* mewakili setiap modul yang dikerjakan karena pada setiap modulnya memiliki alur yang sama. Hal pertama yang dilakukan adalah melakukan *login* dengan *username* dan *password* yang terdaftar pada *database*. Jika *username* dan *password* sesuai, sistem akan memproses halaman *dashboard* LMS. Sedangkan jika *username* dan *password* tidak sesuai maka sistem akan menampilkan pesan *error* dan tetap meminta untuk memasukkan *username* dan *password* yang sesuai.

Ketika memilih menu, sistem akan mengecek akses menu tersebut. Jika memiliki akses menu yang dipilih tersebut, maka sistem akan menampilkan halaman menu yang dipilih. Sebaliknya jika tidak dapat mengakses menu dan muncul pesan *error*, kemungkinan hak akses menu yang dapat diakses belum ditambahkan. Pada setiap halaman menu, dapat memilih aksi yang akan dilakukan yaitu berupa tambah data (*add*), *edit* data (*update*) dan hapus data (*delete*). Jika memilih aksi *add* atau *edit*, maka sistem akan menampilkan *form* untuk melakukan kedua aksi tersebut dan dapat menginput data atau mengedit data yang ditampilkan. Apabila aksi yang dilakukan berhasil, sistem akan menampilkan kembali halaman menu utama dengan notifikasi berhasil *add* atau *edit* data. Apabila tidak berhasil, sistem akan menampilkan validasi

error pada *form*. *Logout* dapat dilakukan apabila tidak memilih aksi maupun menu lainnya.

Use case diagram digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan LMS yang dikembangkan. Berikut adalah *use case diagram* LMS di PT. XYZ. Gambar 4 merupakan *use case diagram* keseluruhan LMS di PT. XYZ. Mulai dari interaksi input data-data *master* hingga dilakukan konfirmasi.

Proses perancangan selanjutnya digambarkan dengan *class diagram* untuk menggambarkan struktur LMS dari pendefinisian kelas-kelas pembuatan LMS. *Class Diagram* pada *Learning Management System (LMS)* dikelompokkan menjadi 5 bagian yaitu *Master 1*, *Master 2*, *Master 3*, *WO* dan *Penjadwalan*. Gambar 4 merupakan *class diagram* untuk tabel *master 1*.

Perancangan *database* pada *Learning Management System (LMS)* digambarkan menggunakan *Entity Relationship Diagram (ERD)*. *ERD* LMS juga dikelompokkan menjadi 5 bagian yaitu *Master 1*, *Master 2*, *Master 3*, *WO* dan *Penjadwalan*.

Setelah adanya perancangan dengan *ERD*, kamus data dibutuhkan untuk menggambarkan rincian setiap entitas. Dalam LMS dibuat satu *database* yang terdiri dari 67 tabel. Namun dalam pembahasan ini, hanya akan dijelaskan 2 tabel terkait modul *master* materi.

Pada pengembangan LMS khususnya modul *master* materi melibatkan dua tabel yaitu tabel *master* materi dan tabel *master* jenis materi seperti yang digambarkan pada kedua tabel diatas.

Perancangan tampilan sistem akan digambarkan melalui desain *mockup* khususnya modul *mas-*

TABEL 1
KAMUS DATA TABEL MASTER MATERI

No.	Nama	Type	Length	Keterangan
1.	id_materi	int	11	Primary Key
2.	id_jenis_materi	int	11	Foreign Key
3.	kode_materi	varchar	50	Not Null
4.	materi	varchar	255	Not Null
5.	durasi	tinyint	4	Not Null
6.	nilai_minimum	decimal	4,2	Not Null
7.	dibuat_oleh	int	11	Not Null
8.	waktu_buat	timestamp	-	Not Null
9.	diedit_oleh	int	11	Not Null
10.	waktu_edit	timestamp	-	Not Null
11.	status	enum	'0','1'	Not Null

ter materi. Desain *mockup* tersebut akan mewakili modul *master* lainnya dikarenakan setiap modul pada LMS ini memiliki kesamaan pada setiap rancangannya. Lampiran Gambar I merupakan rancangan halaman utama pada modul *master* materi. Pada halaman ini, terdapat daftar (*list*) materi untuk kegiatan pembelajaran atau pelatihan. Selain itu, halaman ini digunakan sebagai penghubung untuk melakukan aksi *Add*, *Edit* dan *Delete*.

Lampiran Gambar II merupakan rancangan halaman *add* pada modul *master* materi yang terdiri dari kolom-kolom (*field*) inputan. Secara umum, rancangan halaman *add* pada halaman modul lainnya baik modul *master* maupun modul *detail* adalah sama. Perbedaannya hanya terletak pada kolom-kolom (*field*) inputan yang harus diisi sesuai dengan tabel-tabel yang terlibat dengan modul tersebut.

Lampiran Gambar III merupakan rancangan halaman *edit* modul *master* materi. Form yang ditampilkan pada halaman *edit* sama dengan form yang ditampilkan pada halaman *add*, namun perbedaannya hanya terletak pada kolom (*field*) inputannya. Sistem akan menampilkan data sebelum dilakukan perubahan pada setiap kolom (*field*) inputan seperti pada gambar diatas. Rancangan halaman *edit* pada modul lain juga memiliki kesamaan dengan rancangan halaman *edit* pada modul *master* materi. Lampiran Gambar IV menunjukkan rancangan *alert* pemberitahuan apabila akan dilakukan hapus (*delete*) data materi.

Implementasi *Learning Management System* (LMS) modul *master* materi pada PT.XYZ dilakukan dengan menggunakan *framework* Bootstrap, *framework* CodeIgniter dan *framework* Angular JS. Adapun tujuan implementasi adalah untuk mengetahui apakah perangkat lunak yang dibangun dapat berjalan dan memiliki kualitas yang baik serta memungkinkan dilakukannya pengembangan lebih lanjut.

Implementasi LMS dilakukan dengan melakukan *coding* program di mulai dari proses implementasi *database* yang telah dirancang sebelumnya, hingga proses implementasi sistem baik

TABEL 2
KAMUS DATA TABEL MASTER JENIS MATERI

No.	Nama	Type	Length	Keterangan
1.	id_jenis_materi	int	11	Primary Key
2.	kode_jenis_materi	varchar	50	Not Null
3.	jenis_materi	varchar	50	Not Null
4.	dibuat_oleh	int	11	Not Null
5.	waktu_buat	timestamp	-	Not Null
6.	diedit_oleh	int	11	Not Null
7.	waktu_edit	timestamp	-	Not Null
8.	status	enum	'0','1'	Not Null

front-end maupun *back-end* LMS. Setelah tahap implementasi, proses pengujian dilakukan untuk mengetahui kualitas kinerja dari LMS dengan menggunakan pengujian *black box*.

Implementasi sistem merupakan hasil dari perancangan sistem yang telah dibuat sebelumnya. Pada bagian ini hanya dijabarkan hasil implementasi *Learning Management System* (LMS) yaitu modul *master* materi dikarenakan hasil implementasi untuk modul *master* lainnya sama dengan modul tersebut. Jadi penjelasan modul tersebut akan mewakili modul-modul lainnya. Lampiran Gambar V menunjukkan tampilan halaman utama Modul *master*.

Lampiran Gambar VI menampilkan daftar materi yang memiliki status aktif atau belum dihapus. Data yang ditampilkan berupa kode materi, jenis materi, materi, durasi, nilai minimum dan status. Selain menampilkan daftar materi, user juga dapat membuat (menambahkan), mengubah (*edit*), dan menghapus data materi.

Lampiran Gambar VII merupakan form untuk menambahkan data materi. Setiap kolom pada form harus diisi, mulai dari kode materi, jenis materi, materi, durasi dan nilai minimum. Pada kolom jenis materi, user dapat memilih jenis materi yang tersedia. Sedangkan untuk durasi dan nilai minimum yang diinputkan berupa angka (*numeric*). Setelah proses *input* data dilakukan, sistem akan menampilkan kembali halaman *index* dengan pemberitahuan penambahan data berhasil.

Pengujian merupakan salah satu tahapan yang ada di dalam siklus hidup pengembangan sistem. Tujuan utamanya untuk mengetahui apakah sistem yang dibangun telah sesuai dengan kebutuhan *user*. Menurut Rizky [10], Secara teoritis, *testing* dapat dilakukan dengan berbagai jenis tipe dan teknik. Namun secara garis besar, terdapat dua jenis tipe *testing* yang paling umum digunakan di dalam lingkup rekayasa perangkat lunak. Dua jenis tersebut adalah *white box testing* dan *black box testing*.

Rencana pengujian perangkat lunak pada *Learning Management System* (LMS) menggunakan *black box testing*. *Black box testing* berfokus

pada persyaratan fungsional perangkat lunak yang dibuat. Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Pada *black box testing*, perangkat lunak tersebut akan dieksekusi kemudian dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar *listing* programnya [10]. Rencana pengujian sistem dapat dilihat pada Lampiran Tabel I.

Setelah melaksanakan rencana pengujian maka dilanjutkan dengan kasus dan hasil pengujian. Hal ini seperti terdapat pada Lampiran Tabel II. Dari hasil pengujian tersebut, diketahui bahwa pengembangan LMS pada modul *master* materi yang dikembangkan dapat berjalan sesuai dengan permintaan dan kebutuhan perusahaan. Kinerja perusahaan menjadi lebih efektif dan efisien dengan kemudahan pengolahan data-data *master* yang dibutuhkan dalam kegiatan pembelajaran dan pelatihan. Selain pengujian *black box*, dilakukan juga *User Acceptance Test* (UAT) untuk pengujian dari sisi *user* di PT. XYZ.

4. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan penelitian yang dilakukan diatas dalam hal ini pengembangan *Learning Management System* (LMS) adalah bahwa analisa data terkait perencanaan dan perancangan sistem sangat berguna untuk mendukung pengembangan LMS. LMS berjalan sesuai dengan permintaan dan kebutuhan perusahaan dalam pelaksanaan kegiatan pembelajaran dan pelatihan para pegawainya. LMS mempermudah dalam pengelolaan data-data *master* dan data penjadwalan lab sehingga kinerja perusahaan lebih efektif dan efisien dalam menyelenggarakan kegiatan pembelajaran.

Pada penelitian selanjutnya, pengembangan LMS dapat menggunakan *framework* lain, selain *framework* Codeigniter dan AngularJS. Dapat pula disertai dengan penambahan fitur-fitur baru sesuai

dengan proses bisnis PT. XYZ. Penggunaan metode pengembangan sistem lainnya selain metode SDLC model inkremental juga dapat menjadi alternatif penelitian.

Referensi

- [1] Amiroh, *Membangun E-Learning dengan LMS Moodle*, Sidoarjo, Genta Group Production, 2012.
- [2] Zakir, A., "Rancang Bangun Responsive Web Layout dengan Menggunakan Bootstrap Framework," *Jurnal Nasional Informatika dan Teknologi Jaringan*, p. 7. 2016.
- [3] Haviv, A. Q., *MEAN Web Development*. Birmingham B2 2PB UK, Packt Publishing Ltd, 2014.
- [4] Rahmadiansyah, D. & Irawan, D., "Implementasi Metode Model View Controller Menggunakan Framework Code Igniter dalam Pengembangan Aplikasi Manajemen Depo Petikemas pada Unit Usaha Belawan Logistics Center," *Seminar Nasional Teknologi Informasi dan Komunikasi (SNASTIKOM)*, pp. 1-6. 2012.
- [5] Riyanto, *Membuat Sendiri Aplikasi E-Commerce dengan PHP dan MySQL Menggunakan CodeIgniter dan JQuery*, Yogyakarta, ANDI, 2011.
- [6] Supono & Putratama, V., *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*, Yogyakarta, Deepublish, 2016.
- [7] Al Fatta, H., *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*, Yogyakarta, ANDI, 2007.
- [8] Pressman, R. S., *Software Engineering*, New York, McGraw-Hill, 2001.
- [9] Sitorus, L., *Algoritma dan Pemrograman*, Yogyakarta, ANDI, 2015.
- [10] Rizky, S., *Konsep Dasar Rekayasa Perangkat Lunak*, Jakarta, PT.Prestasi Pustakarya, 2011.

Lampiran

LAMPIRAN TABEL I
RENCANA PENGUJIAN MODUL *MASTER* MATERI

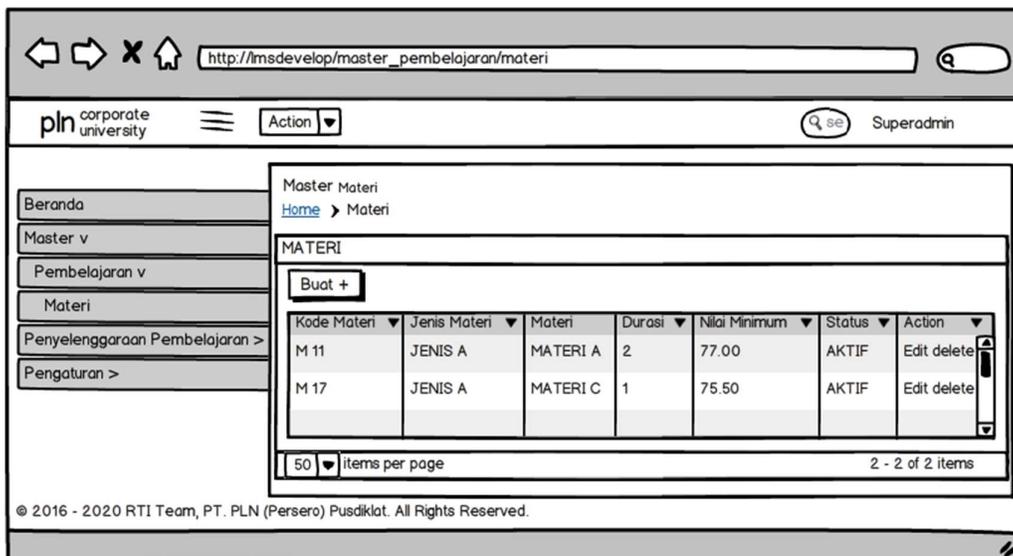
Halaman Pengujian	Detail Pengujian	Jenis Pengujian
Tambah <i>Master</i> Materi	Data <i>master</i> materi akan tersimpan ke <i>database</i> dengan status 1 (Aktif) setelah <i>admin</i> meng-klik button <i>Add</i> , lalu sistem kembali ke halaman modul <i>master</i> materi dengan notifikasi berhasil <i>add</i> data.	<i>Black Box</i>
<i>Edit Master</i> Materi	Mengubah data <i>master</i> materi yang tersimpan di <i>database</i> setelah <i>admin</i> meng-klik button <i>Edit</i> , lalu sistem kembali ke halaman modul <i>master</i> materi dengan notifikasi berhasil <i>edit</i> data.	
Hapus <i>Master</i> Materi	Menghapus data <i>master</i> materi hanya pada halaman modul <i>master</i> materi (bukan pada <i>database</i>), dan mengubah status data materi menjadi 0 (Tidak Aktif) di <i>database</i> .	
Pencarian Data <i>Master</i> Materi	Melakukan pencarian data pada halaman modul <i>master</i> materi sesuai dengan kata kunci yang diinputkan di setiap kolomnya.	

LAMPIRAN TABEL II
HASIL PENGUJIAN MODUL MASTER MATERI PADA DATA NORMAL

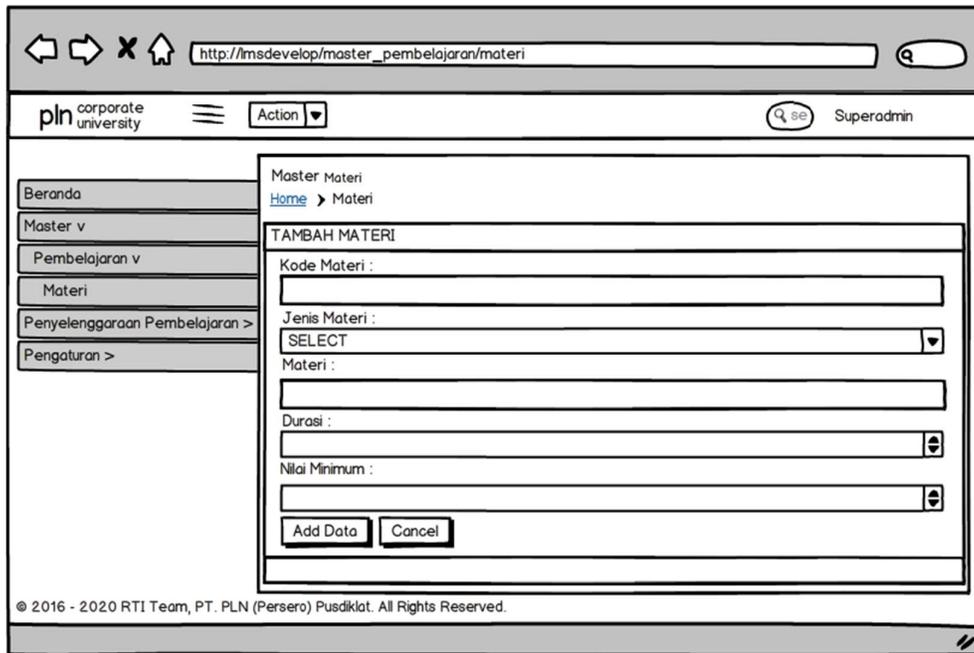
Data Masukan	Harapan	Pengamatan	Kesimpulan
Mengisi semua data pada form <i>add</i> materi	Data materi tersimpan ke <i>database</i> dan tampil pada halaman modul <i>master</i> materi.	Data materi tersimpan ke <i>database</i> dengan status 1 (Aktif) dan ditampilkan pada halaman modul <i>master</i> materi, serta tampil notifikasi berhasil tambah data.	[√] diterima [] ditolak
Klik tombol <i>Add</i>			
Mengubah satu atau lebih data materi pada form <i>edit</i>	Data materi pada halaman modul <i>master</i> materi dan data materi pada <i>database</i> berhasil di <i>edit</i> .	Data materi pada halaman modul <i>master</i> materi dan data materi pada <i>database</i> berubah, serta tampil notifikasi berhasil <i>edit</i> data.	[√] diterima, [] ditolak
Klik tombol <i>Edit</i>			
Memilih salah satu data materi	Data materi pada halaman modul <i>master</i> materi terhapus dan data materi pada <i>database</i> tidak terhapus.	Data materi pada halaman modul <i>master</i> materi terhapus dan data materi pada <i>database</i> tidak terhapus tetapi berubah status menjadi 0 (Tidak Aktif), serta muncul notifikasi berhasil hapus data.	[√] diterima, [] ditolak
Klik tombol <i>Delete</i>			
Memasukkan kata kunci pencarian "MATERI A"	Menampilkan data materi sesuai dengan kata kunci yang diinputkan.	Menampilkan data "MATERI A" sesuai dengan kata kunci yang dimasukkan.	[√] diterima, [] ditolak

LAMPIRAN TABEL III
HASIL PENGUJIAN MODUL MASTER MATERI PADA DATA TIDAK NORMAL

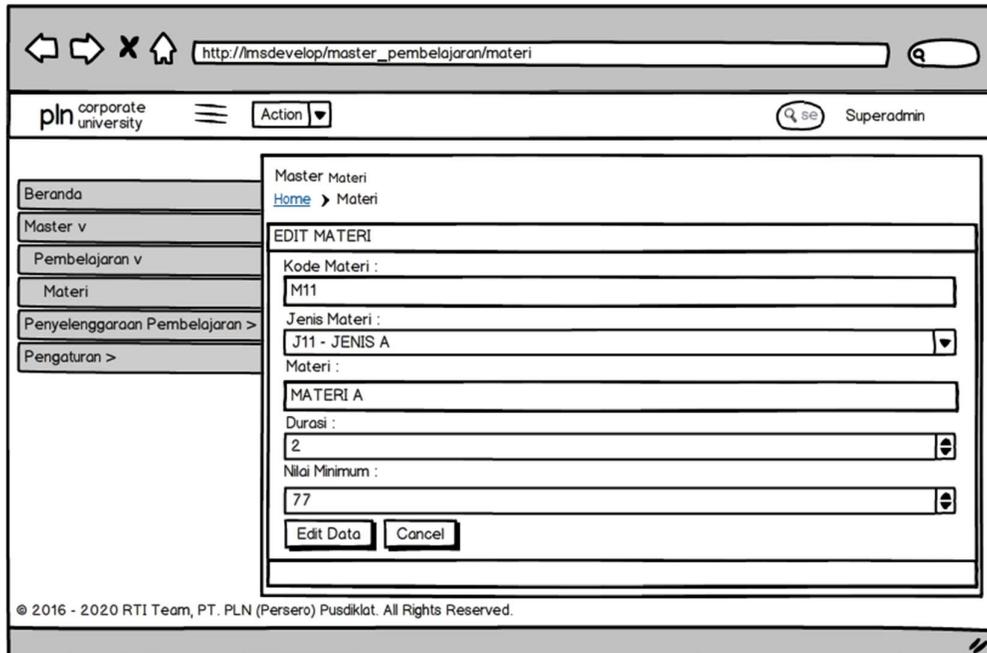
Data Masukan	Harapan	Pengamatan	Kesimpulan
Field pada form tambah data ada yang tidak diisi atau kosong.	Menampilkan pesan validasi bahwa data materi harus diisi.	Menampilkan pesan validasi "data materi harus diisi" pada bagian atau field yang kosong dan sistem tetap menampilkan halaman <i>add</i> data <i>master</i> materi.	[√] diterima, [] ditolak
Data materi sudah ada di <i>database</i> dengan status 1 (Aktif)	Menampilkan pesan validasi bahwa data materi sudah ada.	Menampilkan pesan validasi "data materi sudah ada" pada bagian atau <i>field</i> yang diinputkan dan sistem tetap menampilkan halaman <i>add</i> data <i>master</i> materi.	[√] diterima, [] ditolak
Tidak melakukan perubahan data apapun.	Menampilkan pesan validasi bahwa tidak ada perubahan.	Menampilkan pesan validasi "data tidak ada perubahan" dan sistem tetap menampilkan halaman <i>edit</i> materi.	[√] diterima, [] ditolak
Klik tombol <i>Edit</i> .			



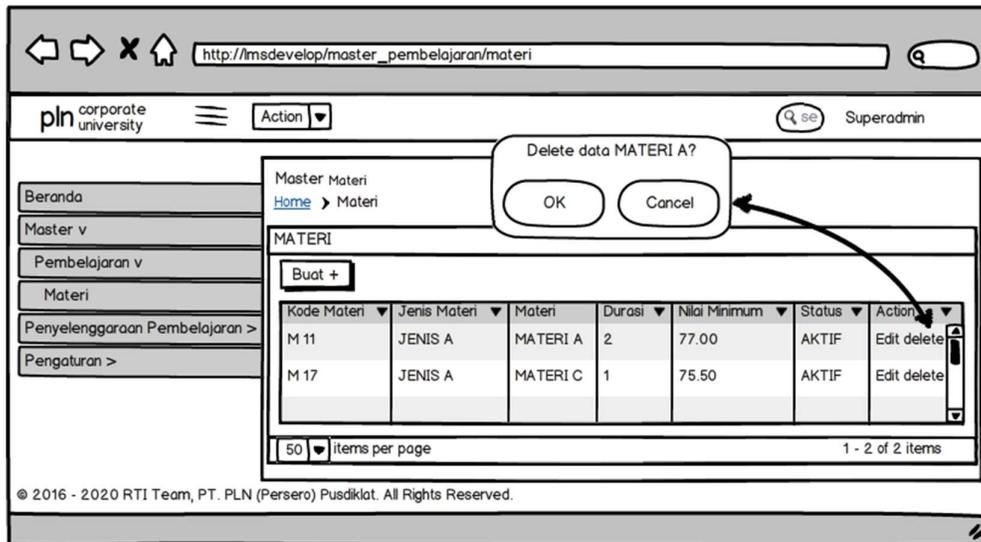
Lampiran Gambar I. Rancangan Halaman Utama Modul Master Materi



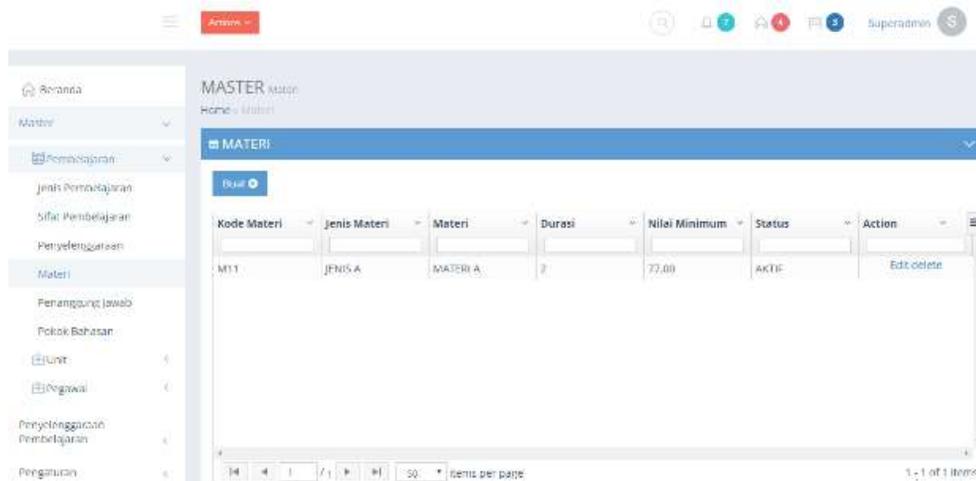
Lampiran Gambar II. Rancangan Halaman Add Modul Master Materi



Lampiran Gambar III. Rancangan Halaman Edit Modul Master Materi



Lampiran Gambar IV. Rancangan Alert Delete Modul Master Materi



Lampiran Gambar V. Tampilan Halaman Utama (Index) Modul Master Materi

The screenshot shows the 'TAMBAH MATERI' form in the MASTER Materi module. The form includes the following fields:

- Kode Materi:
- Jenis Materi:
- Materi:
- Durasi:
- Nilai Minimum:

Buttons:

Lampiran Gambar VI. Tampilan Halaman Buat (Add) Data Modul Master Materi

The screenshot shows the MASTER Materi index page with a success message: "Tambah Materi MATERI C Berhasil". Below the message is a table with the following data:

Kode Materi	Jenis Materi	Materi	Durasi	Nilai Minimum	Status	Action
M11	JENIS A	MATERI A	2	77.00	AKTIF	Edit delete
M17	JENIS A	MATERI C	2	75.50	AKTIF	Edit delete

Lampiran Gambar VII. Tampilan Index Materi Setelah Add Data Materi