

Pengembangan Aplikasi Manajemen Pelatihan Laboratorium Software Engineering Di Fakultas Teknik Sistem Komputer

Faiz Risaludin Islami¹⁾, Kodrat Iman Satoto²⁾, Rinta Kridalukmana²⁾
Jurusan Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, Tembalang, Semarang, Indonesia

System computer engineering of Diponegoro University has a software engineering laboratory fonctioned to provide the learning and training activities in order to give the opportunity for the students, to examine and to apply the theory or the research and the verification scientific subject study or a part of certain subject study. The probleme emerge in this training is less the information about the training accepted by the society who willing to attend the training at the software engineering laboratory. In propose, to solve all the problemes appear, the writer make a Management Trainee Development Application System of Software Engineering Laboratory at Faculty of System Computer Engineering gave a complete information about the training and the registration in software engineering laboratory.

Management Trainee Application of Software Engineering Laboratory at Faculty of System Computer Engineering is an application based on website, made by using the program language PHP in scope Framework Code Igniter, and also MySQL for database. The developement methode used the waterfall methode, while designing the application used the UML modele.

The result from this application is the realization an application based on website which is able to manage the trainee data in software engineering laboratory, for example to manage the participant training data, the list of training data, the time schedule training data, admin data, the participant traininh data along the completed training History data.

Keywords: *Laboratory, Management applications, PHP, Framework Code Igniter*

I. PENDAHULUAN

Program studi sistem komputer Universitas Diponegoro memiliki laboratorium *software engineering* yang berfungsi memberikan kegiatan pembelajaran yang bertujuan agar mahasiswa mendapat kesempatan untuk menguji dan mengaplikasikan teori atau penyelidikan dan pembuktian ilmiah matakuliah atau bagian matakuliah tertentu, di samping itu

laboratorium *software engineering* juga memberikan pelatihan bagi mahasiswa dan juga masyarakat pada umumnya. Permasalahan yang timbul dalam pelatihan ini adalah kurangnya informasi yang diterima masyarakat yang ingin mengikuti pelatihan. Disamping itu, masyarakat yang membutuhkan informasi yang lebih jelas tentang pelatihan ini harus datang di laboratorium *software engineering*. Hal ini sangat tidak efisien, karena membutuhkan waktu untuk datang ke laboratorium dan hanya dapat dikunjungi pada saat jam kerja.

Untuk mengatasi semua permasalahan yang ada, dibuat sebuah sistem aplikasi manajemen pelatihan yang memberikan informasi lengkap tentang pelatihan dan pendaftaran di laboratorium *software engineering*. Keuntungan yang diperoleh dengan dikembangkannya sistem ini antara lain, untuk masyarakat tidak perlu datang ke laboratorium hanya untuk mendapatkan informasi dan melakukan pendaftaran tanpa terbatas oleh waktu jam kerja. Karena sistem ini dapat diakses kapan dan di mana saja melalui internet. Sistem ini juga memberikan keuntungan admin dalam mengelola informasi maupun data yang ada pada aplikasi manajemen pelatihan laboratorium *software engineering*.

II. LANDASAN TEORI

A. Sistem Informasi

Sistem informasi dapat didefinisikan sebagai berikut :

- Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
- Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
- Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan

strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Organisasi menggunakan sistem informasi untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka. Bank menggunakan sistem informasi untuk mengolah cek-cek nasabah dan membuat berbagai laporan rekening koran dan transaksi yang terjadi. Perusahaan menggunakan sistem informasi untuk mempertahankan persediaan pada tingkat paling rendah agar konsisten dengan jenis barang yang tersedia.

Sebagian sistem informasi berlandaskan komputer terdapat di dalam suatu organisasi dalam berbagai jenis. Anggota organisasi adalah pemakai informasi yang dihasilkan sistem tersebut termasuk manajer yang bertanggung atas pengalokasian sumber daya untuk pengembangan dan pengoperasian perusahaan.^[3]

B. Software Development Life Cycle

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

- a. Inisiasi (*initiation*)
- b. Pengembangan konsep sistem (*system concept development*)
- c. Perencanaan (*planning*)
- d. Analisis kebutuhan (*requirements analysis*)
- e. Desain (*design*)
- f. Pengembangan (*development*)
- g. Integrasi dan pengujian (*integration and test*)
- h. Implementasi (*implementation*)
- i. Operasi dan pemeliharaan (*operations and maintenance*)
- j. Disposisi (*disposition*)^[1]

C. Model Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

Kenyataannya sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab sebagai berikut:

- a. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
- b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
- c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Sistem manajemen adalah suatu interaksi atau kerjasama untuk melakukan fungsi pengolahan data menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Sehingga peranan Sistem Informasi Manajemen dapat digunakan untuk mencari atau memperoleh informasi, diperlukan adanya data dan unit pengolahan data.^[7]

D. ERD

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.^[1]

E. UML

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.^[4]

F. Use Case

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.^[4]

G. Sequence diagram

Sequence diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup

objek dan pesan yang dikirimkan dan diterima antar objek.^[4]

H. Class diagram

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.^[4]

I. PHP

PHP merupakan singkatan dari *Hypertext Preprocessor* yang merupakan bahasa berbentuk skrip yang ditempatkan pada *server* dan diproses di *server*. Hasilnya kemudian dikirimkan ke *browser* klien.^[5]

J. Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML. Sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi klien, yang artinya di sisi *browser* bukan di sisi *server*.^[2]

K. CSS

CSS didesain untuk memisahkan konten dokumen dengan presentasi dokumen, termasuk elemen seperti *layout*, warna dan pilihan huruf (*font*). Pemisahan ini akan memberikan keuntungan dalam hal aksesibilitas konten, meningkatkan fleksibilitas dan memungkinkan beberapa halaman untuk berbagai format yang sama serta mampu mengurangi kompleksitas dan pengulangan dalam penulisan atau struktur konten.^[6]

L. Code Igniter

Code Igniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*model view controller*) untuk membangun *website* dinamis. Dengan menggunakan PHP Code Igniter akan memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.^[10]

M. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user* dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi *GNU General Public Licence* (GPL). Tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.^[9]

III. PERANCANGAN SISTEM

A. Tahapan Perancangan Sistem

Tahapan perancangan sistem memiliki tujuan untuk men-desain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi organisasi atau perusahaan. *Software Development Live Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Model yang digunakan dalam penelitian tugas akhir ini adalah model *waterfall*. Model *waterfall* terdiri dari 5 tahap yaitu Analisis Kebutuhan, Desain, Pengodean, Pengujian dan Pemeliharaan.^[8]

B. Analisis Kebutuhan

Tahapan analisis kebutuhan dilakukan untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat seperti apa yang diinginkan *user* dan kemudian mentransformasikan ke dalam sebuah deskripsi yang jelas dan lengkap.

1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan fungsionalitas atau layanan yang harus diberikan oleh sistem. Kebutuhan ini akan bergantung pada jenis perangkat lunak yang sedang dikembangkan, pengguna yang diharapkan menggunakan perangkat lunak tersebut dan jenis sistem yang akan digunakan. Kebutuhan fungsional dari aplikasi ini adalah:

1. Mengelola data *admin*
2. Membuka dan menutup jalannya pelatihan pada sistem
3. Menambahkan dan menghapus pelatihan
4. Menampilkan *history* pelatihan yang sudah berlangsung
5. Mengelola data peserta pelatihan.

2. Kebutuhan non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang tidak langsung berhubungan dengan spesifik yang disediakan oleh sistem. Kebutuhan ini berhubungan dengan properti sistem yang muncul belakangan, seperti keandalan, waktu tanggap dan penempatan pada media penyimpanan. Batasan sistem mengenai kemampuan piranti masukan dan keluaran yang dipakai pada antarmuka sistem juga termasuk dalam kebutuhan non-fungsional. Kebutuhan non-fungsional aplikasi ini antara lain:

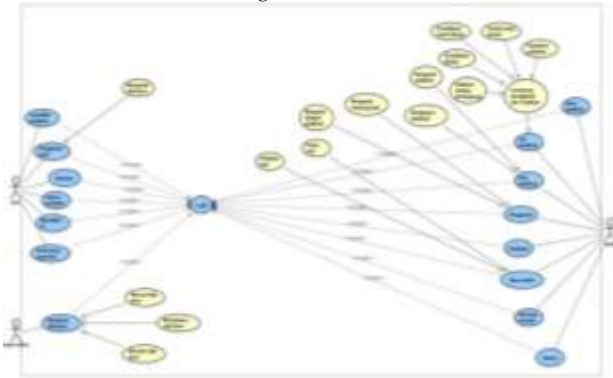
1. Operasional
Aplikasi ini dapat berjalan lancar pada perangkat dekstop, seperti Pc (Personal computer) dan Laptop.
2. Aplikasi membutuhkan koneksi internet untuk mengubungkan data dan server.

C. Desain

Tahap perancangan / desain perangkat lunak merupakan proses multi langkah dan berfokus pada beberapa atribut perangkat lunak. Proses ini berdasarkan dari analisa sebelumnya sehingga menciptakan sebuah rancangan yang sesuai dengan kebutuhan dari pihak terkait. Dalam perancangan perangkat lunak ini, digunakan *Entity Relationship Diagram (ERD)* untuk pemodelan basis data dan *UML* untuk pemodelan perangkat lunak.

1. UML

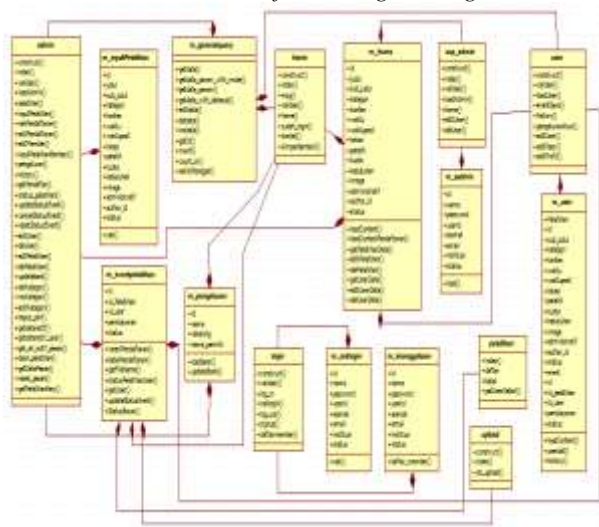
a. Usecase Diagram



Gambar 2 Diagram Usecase

b. Class Diagram

Class diagram atau diagram kelas menggambarkan hubungan yang ada pada kelas-kelas pada aplikasi. Berikut merupakan diagram kelas aplikasi Manajemen Pelatihan Laoratorium *software engineering*

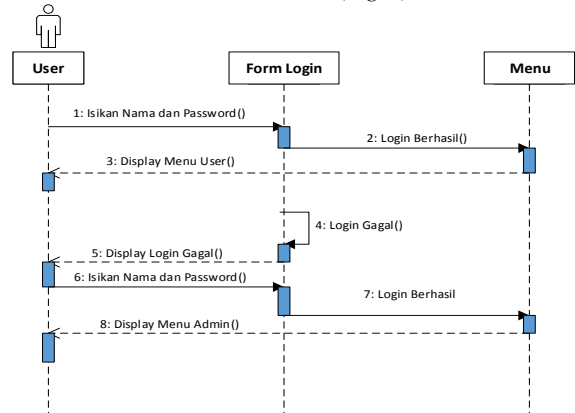


Gambar 3 Diagram Kelas Aplikasi Manajemen Pelatihan Laboratorium *software engineering*

c. Sequence Diagram

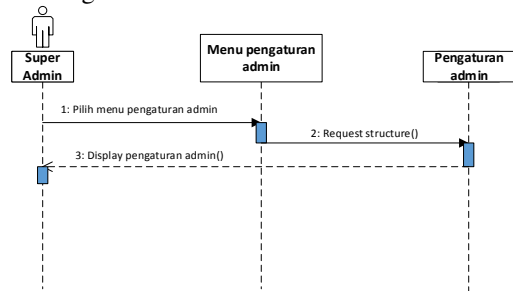
Berkut ini adalah gambar-gambar Diagram *Sequence* dari Pengembangan Aplikasi Manajemen Laboratorium *Software Engineering*.

Gambar 4 merupakan diagram sekuensial untuk masuk ke sistem (*login*).



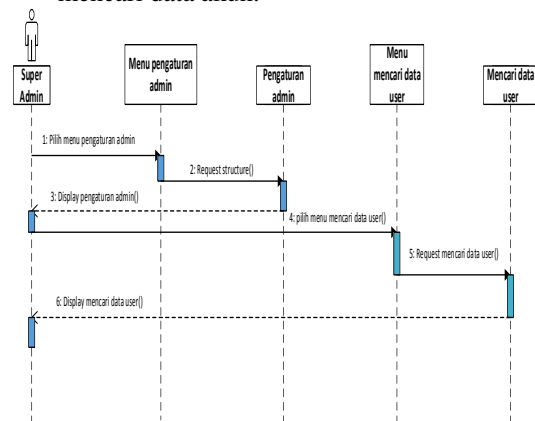
Gambar 4 Diagram sekuensial masuk ke sistem (*login*)

Gambar 5 merupakan diagram sekuensial dari mengelola data akun



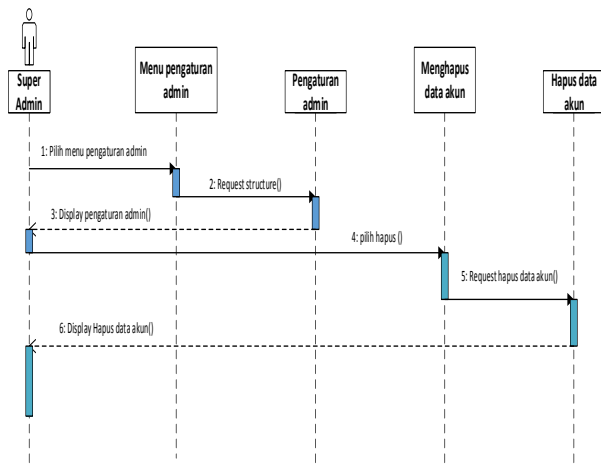
Gambar 5 Diagram sekuensial dari mengelola data akun

Gambar 6 merupakan diagram sekuensial dari mencari data akun.



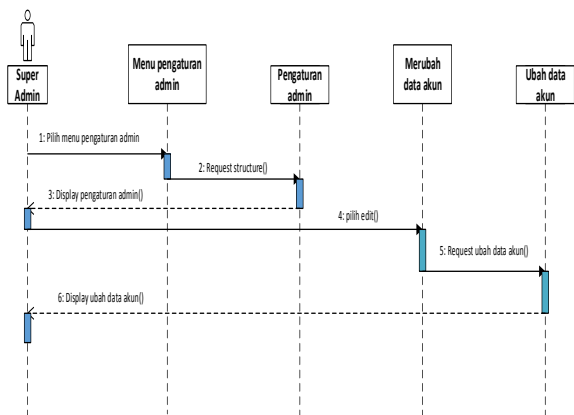
Gambar 6 Diagram sekuensial dari mencari data akun

Gambar 7 merupakan diagram sekuensial dari menghapus data *user*.



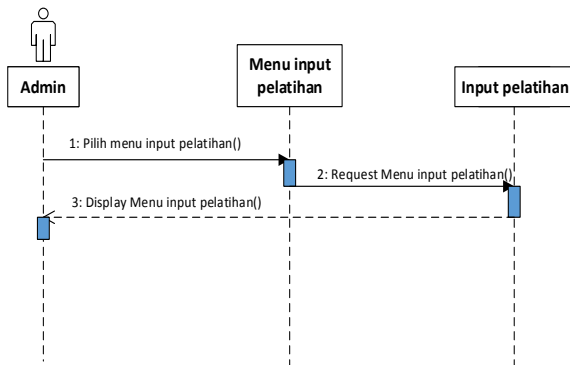
Gambar 7 Diagram sekuensial dari Menghapus data user

Gambar 8 merupakan diagram sekuensial dari merubah data akun.

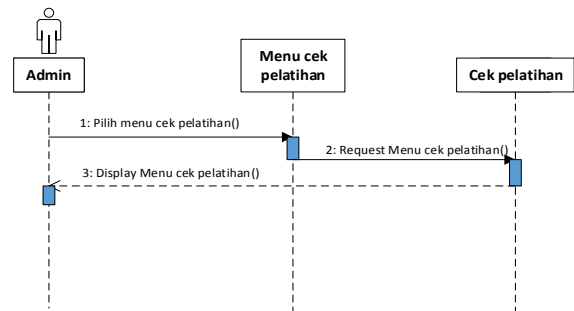


Gambar 8 Diagram sekuensial dari merubah data akun

Gambar 9 merupakan diagram sekuensial dari Input Pelatihan

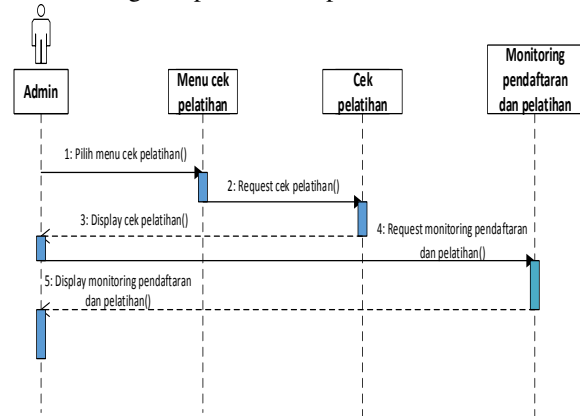


Gambar 9 Diagram sekuensial dari input pelatihan
Gambar 10 merupakan diagram sekuensial dari Cek Pelatihan



Gambar 10 Diagram sekuensial dari cek pelatihan

Gambar 11 merupakan diagram sekuensial dari monitoring dan pendaftaran pelatihan



Gambar 11 Diagram sekuensial dari monitoring dan pendaftaran pelatihan

IV. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi Basis Data

Tahapan ini mengimplementasikan hasil perancangan basis pada sistem yang telah dibuat sebelumnya.

Gambar 12 merupakan contoh implementasi pada tabel Bank.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(20)	latin1_swedish_ci		No	None	
3	rekening	varchar(30)	latin1_swedish_ci		No	None	
4	nama_pemilik	varchar(50)	latin1_swedish_ci		No	None	

Gambar 12 Implementasi tabel bank

Implementasi tabel yang lain ditunjukkan oleh gambar 13 sampai dengan gambar 18.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(100)			No	None	AUTO_INCREMENT
2	id_pelatihan	varchar(5)	latin1_swedish_ci		No	None	
3	id_user	varchar(5)	latin1_swedish_ci		No	None	
4	pembayaran	varchar(100)	latin1_swedish_ci		Yes	none	
5	status	varchar(20)	latin1_swedish_ci		No	normal	

Gambar 13 Implementasi tabel event

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(5)			No	None	AUTO_INCREMENT
2	konten	varchar(50)	latin1_swedish_ci		Yes	NULL	
3	id_user	varchar(5)	latin1_swedish_ci		Yes	NULL	
4	waktu	varchar(20)	latin1_swedish_ci		Yes	NULL	
5	read_status	int(11)			No	0	

Gambar 14 Implementasi tabel History

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(20)			No	None	AUTO_INCREMENT
2	kategori	varchar(30)	latin1_swedish_ci		No	None	

Gambar 15 Implementasi tabel Kategori

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(100)			No	None	AUTO_INCREMENT
2	judul	varchar(80)	latin1_swedish_ci		No	None	
3	sub_judul	varchar(80)	latin1_swedish_ci		Yes	NULL	
4	kategori	varchar(30)	latin1_swedish_ci		Yes	NULL	
5	konten	varchar(250)	latin1_swedish_ci		No	None	
6	waktu	varchar(30)	latin1_swedish_ci		No	None	
7	waktu_post	varchar(20)	latin1_swedish_ci		Yes	NULL	
8	lokasi	varchar(100)	latin1_swedish_ci		No	None	
9	pelatih	varchar(30)	latin1_swedish_ci		No	None	
10	kuota	int(3)			Yes	NULL	
11	kebutuhan	varchar(50)	latin1_swedish_ci		Yes	NULL	
12	image	varchar(40)	latin1_swedish_ci		Yes	NULL	
13	administratif	varchar(15)	latin1_swedish_ci		Yes	0	
14	author_id	varchar(5)	latin1_swedish_ci		Yes	NULL	
15	status	varchar(11)	latin1_swedish_ci		Yes	tidak	

Gambar 16 Implementasi table pelatihan

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(100)			No	None	AUTO_INCREMENT
2	pesan	longtext	latin1_swedish_ci		Yes	NULL	
3	nama	varchar(50)	latin1_swedish_ci		Yes	NULL	
4	email	varchar(40)	latin1_swedish_ci		Yes	NULL	
5	telep	varchar(15)	latin1_swedish_ci		Yes	NULL	
6	waktu	varchar(20)	latin1_swedish_ci		Yes	NULL	
7	read_status	int(2)			No	0	

Gambar 17 Implementasi tabel pesan_umum

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(30)	latin1_swedish_ci		No	None	
3	password	varchar(32)	latin1_swedish_ci		No	None	
4	userid	varchar(15)	latin1_swedish_ci		No	None	
5	alamat	varchar(50)	latin1_swedish_ci		Yes	NULL	
6	email	varchar(40)	latin1_swedish_ci		Yes	NULL	
7	instansi	varchar(30)	latin1_swedish_ci		Yes	NULL	
8	status	varchar(10)	latin1_swedish_ci		No	normal	

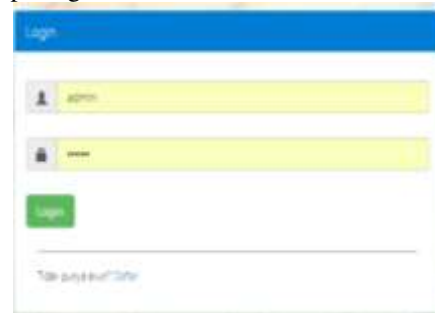
Gambar 18 Implementasi tabel user

B. Implementasi Antarmuka

Berikut ini merupakan antarmuka yang dimiliki Aplikasi Manajemen Pelatihan Laboratorium:

1. Halaman Login

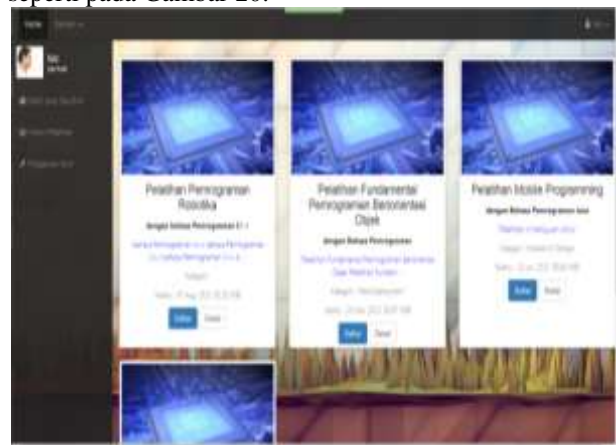
Berisi *form* untuk melakukan *login* ke aplikasi dengan memasukkan *username* dan *password*, tampilan seperti pada gambar 19.



Gambar 19 Halaman Login

2. Halaman User

Berisi fitur untuk mendaftar pelatihan, tampilan seperti pada Gambar 20.



Gambar 20 Halaman Pengguna

3. Halaman Admin

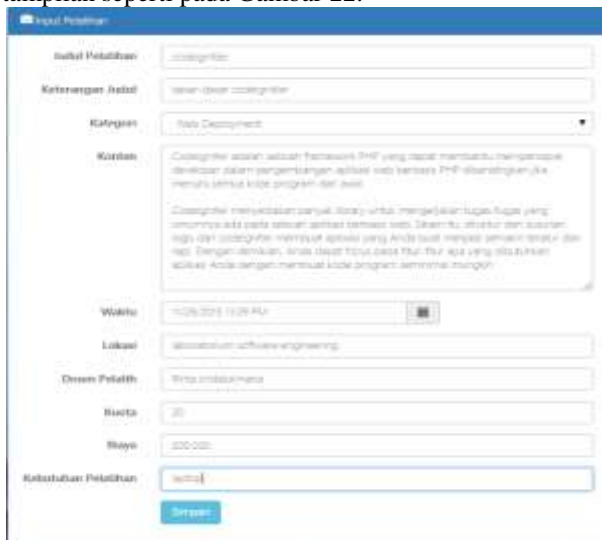
Berisi fitur untuk melihat pesan *user* dan pelatihan baru, tampilan seperti pada Gambar 21.



Gambar 21 Halaman Admin

4. Halaman *Input* pelatihan

Berisi fitur untuk menambahkan pelatihan baru, tampilan seperti pada Gambar 22.



Gambar 22 Halaman *Input* Pelatihan

5. Halaman *Cek* Pelatihan

Berisi fitur untuk mengelola pendaftaran dan waktu pelatihan, tampilan seperti pada Gambar 23.



Gambar 23 Halaman *Cek* Pelatihan

6. Halaman *Edit* Pelatihan

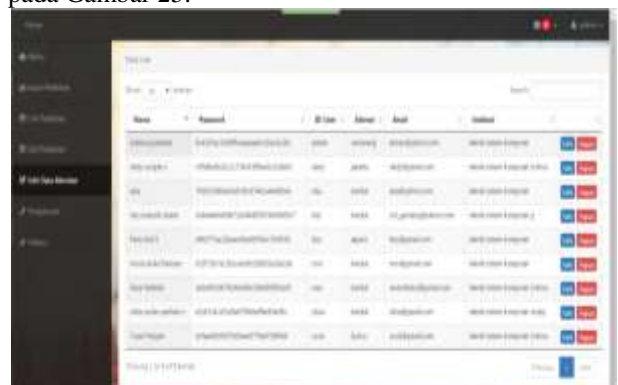
Berisi fitur untuk mengubah atau menghapus pelatihan, tampilan seperti pada Gambar 24.



Gambar 24 Halaman *Edit* Pelatihan

7. Halaman *Edit* Data *Member*

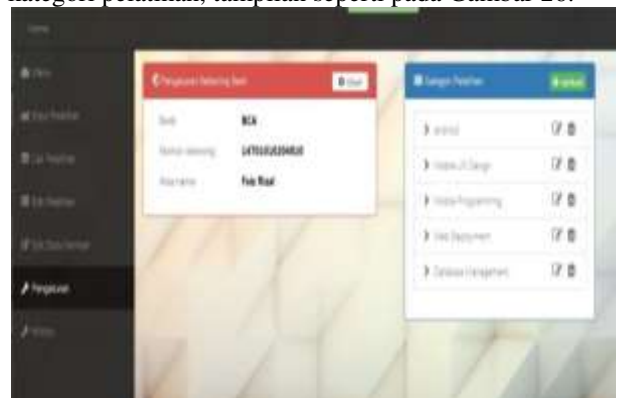
Berisi fitur untuk mengelola *user*, tampilan seperti pada Gambar 25.



Gambar 25 Halaman *Edit* Data *Member*

8. Halaman *Menu* Pengaturan *Admin*

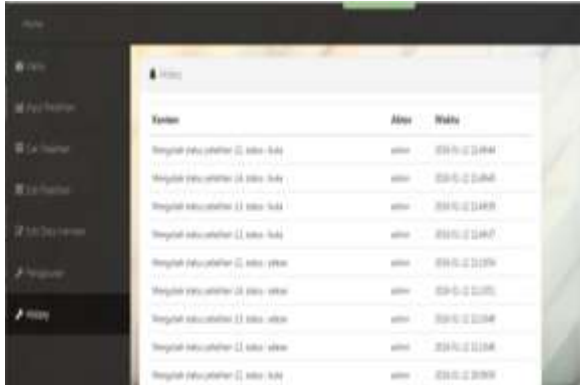
Berisi fitur untuk mengelola nomor rekening bank dan kategori pelatihan, tampilan seperti pada Gambar 26.



Gambar 26 Halaman *Menu* Pengaturan *Admin*

9. Halaman *History* *Admin*

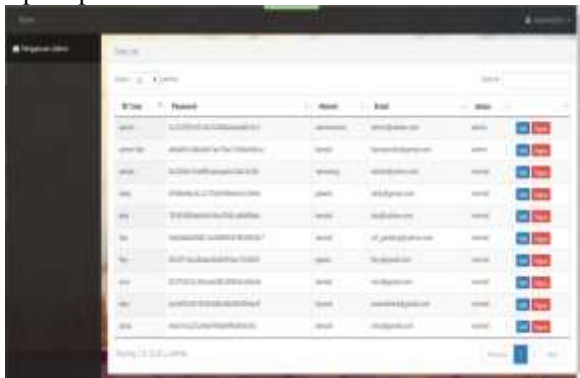
Berisi fitur untuk melihat semua aktifitas yang dilakukan *admin*, tampilan seperti pada Gambar 27.



Gambar 27 Halaman History Admin

10. Halaman Superadmin

Berisi fitur untuk mengelola admin dan user, tampilan seperti pada Gambar 28.



Gambar 28 Halaman Superadmin

C. Pengujian

Proses pengujian dilakukan menggunakan metode *black-box* berdasarkan fungsi dari aplikasi yang tertera pada diagram *use case*.

Tabel-tabel di bawah ini merupakan ringkasan dari pengujian yang telah dilakukan. Pertama adalah Tabel 1 yang merupakan ringkasan pengujian yang berhubungan dengan pengelolaan data User.

Tabel 1 Pengujian *Blackbox*

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
Pengujian login	Memasukkan <i>username</i> dan <i>password</i> yang digunakan untuk login	Muncul halaman utama <i>user</i>	Berhasil
Pengujian mencari data <i>user</i>	Memasukkan nama <i>user</i> yang akan dicari di <i>menu</i> daftar <i>user</i>	Muncul nama <i>user</i> yang akan dicari di <i>menu</i> daftar <i>user</i>	Berhasil
Pengujian ubah data <i>user</i>	Memilih data pengguna yang tersedia dan mengubah informasinya	Muncul pesan data pengguna berhasil diperbarui dan data yang baru muncul dalam	Berhasil

Pengujian hapus data <i>user</i>	Memilih data pengguna tertentu dan menghapus data tersebut dengan mengklik tombol hapus	daftar Muncul pesan data pengguna berhasil dihapus dan data yang dihapus menghilang dari daftar	Berhasil
----------------------------------	---	--	----------

Tabel 2 Pengujian *Blackbox* (lanjutan)

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
Pengujian login	Memasukkan <i>username</i> dan <i>password</i> yang digunakan untuk login	Muncul halaman utama <i>user</i>	Berhasil
Pengujian menu <i>admin</i>	Memilih pesan <i>user</i> dan pelatihan baru yang ada pada <i>menu admin</i>	Maka akan muncul <i>menu</i> pesan <i>user</i> dan <i>menu</i> pelatihan baru	Berhasil
Pengujian input pelatihan	Memilih input pelatihan maka akan muncul form input pelatihan, setelah selesai menginput klik tombol <i>save</i>	Muncul pesan input berhasil, dan muncul pelatihan baru	Berhasil
Pengujian cek pelatihan	Memilih tombol pendaftar maka akan muncul jumlah pendaftar dan mengelola status pelatihan	Pendaftar pelatihan telah terdaftar dan status pelatihan bisa dirubah.	Berhasil
Pengujian edit pelatihan	Memilih pelatihan yang akan diedit dan dihapus	Maka akan muncul form edit pelatihan, isi semua form dan klik tombol <i>save</i>	Berhasil
Pengujian Edit data member	Memilih <i>user</i> yang akan diedit atau dihapus di <i>menu</i> data <i>user</i>	maka akan muncul <i>menu</i> edit <i>user</i> dan ketik simpan setelah selesai, dan tombol hapus untuk hapus <i>user</i>	Berhasil
Pengujian Pengaturan	Mengubah kategori pelatihan dan mengubah rekening <i>bank</i>	Kategori pelatihan berhasil diubah dan rekening <i>bank</i>	Berhasil
Pengujian History	Memilih History	Maka akan muncul history seluruh kegiatan apa saja yang dilakukan oleh <i>admin</i>	Berhasil
Pengujian notifikasi	Memilih tombol notifikasi	Maka akan muncul <i>menu</i> seluruh pesan <i>user</i>	Berhasil

Tabel 3 Pengujian *Blackbox* (lanjutan)

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
Pengujian login	Memasukkan <i>username</i> dan <i>password</i> yang digunakan untuk login	Muncul halaman utama <i>user</i>	Berhasil
Pengujian mendaftar Id	Memilih tombol daftar maka akan muncul form daftar, isi semua dan klik daftar	Muncul pesan Pendaftaran berhasil	Berhasil
Pengujian memilih pelatihan	Memilih Pelatihan kemudian klik daftar, dan upload bukti pembayaran pelatihan	Muncul pesan menunggu konfirmasi pada pelatihan yang didaftar	Berhasil
Pengujian bantuan	Memilih bantuan dan akan muncul tombol hubungi <i>admin</i> .	Muncul form pesan hubungi <i>admin</i> , isikan kan semua dan tekan tombol kirim	Berhasil
Pengujian event yang saya ikuti	Memilih event yang saya ikuti	Maka akan muncul pelatihan yang diikuti oleh <i>user</i>	Berhasil
Pengujian history Pelatihan	Memilih history pelatihan	Maka akan muncul pelatihan-pelatihan yang pernah di ikuti oleh <i>user</i> dan sudah selesai	Berhasil
Pengujian pengaturan Akun	Mengubah pengaturan Id dan profil,	Maka akan muncul edit profil dan klik simpan, dan pengaturan Id setelah selesai klik simpan	Berhasil

V. PENUTUP

Kesimpulan dan saran dari hasil penelitian dan pembahasan adalah sebagai berikut.

A. Kesimpulan

Dari hasil pengujian dan analisis Pengembangan aplikasi manajemen pelatihan laboratorium *software engineering* di fakultas teknik sistem komputer, dapat disimpulkan beberapa hal sebagai berikut:

1. Berdasarkan hasil pengujian aplikasi menggunakan metode *Black-box*, seluruh fungsi menu yang ada dalam aplikasi telah berhasil sesuai dengan fungsinya.

2. Aplikasi ini mempunyai 3 *user* yaitu, super admin, admin dan member
3. Aplikasi ini memiliki fitur notifikasi pada admin yang berfungsi memberikan pemberitahuan apabila ada keluhan atau pengaduan dari *user* tentang pelatihan.
4. Super admin berwenang untuk mengelola data akun.
5. Admin di sini bertugas mengelola pendaftar pelatihan dan mengatur jalannya pelatihan.
6. Aplikasi ini dapat melakukan fungsi pendaftaran pelatihan dan konfirmasi pembayaran secara *online*.

B. Saran

Terdapat saran dari hasil penelitian yaitu sebagai berikut :

1. Pengembangan aplikasi manajemen pelatihan laboratorium *software engineering* di fakultas teknik sistem komputer dapat dikembangkan ke versi *mobile*.
2. Menambahkan notifikasi pada *user* sebagai informasi pendaftaran pelatihan, apakah pendaftaran pelatihan *user* sudah diterima atau ditolak admin.

DAFTAR PUSTAKA

- [1] A.S, Rosa dan Shalahuddin, M., *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Modula, Bandung, 2011.
- [2] Adi, A.P dan Sanjay, R., *Web Makin Dahsyat dengan JQuery*, Kompas Gramedia. Semarang, 2012.
- [3] Bin Ladjamudin, Al-Bahra, *Analisis dan Desain Sistem Informasi*, Graha Ilmu, Yogyakarta, 2005.
- [4] Gata, Windu dan Gata, Grace., *Sukses Membangun Aplikasi Penjualan dengan Java*, PT Elex Media Komputindo, Jakarta, 2013.
- [5] Hirin A.M dan Virgi., *Cepat Mahir Pemrograman Web dengan PHP dan MySQL*, Prestasi Pustakaraya, Jakarta, 2011.
- [6] Huda, Miftahul., *Membuat Aplikasi Database dengan Java, MySQL dan Netbeans*, PT. Elex Media Komputindo, Jakarta, 2010.
- [7] Sutanta, Edhy, *Sistem Informasi Manajemen*, Graha Ilmu, Yogyakarta, 2003.
- [8] R. Tantra, *Manajemen Proyek Sistem Informasi*, Yogyakarta: ANDI, 2012.
- [9] Wahyono, Teguh., *Practice Guide PHP OnWindows*, PT. Elex Media Komputindo, Jakarta, 2009.
- [10] Wiswakarma, Komang., *9 Langkah Menjadi Master Framework CodeIgniter.*, Loko Media, Yogyakarta, 2010.