

Penerapan Metode Prototyping dalam Pengembangan Sistem Audit Internal Tingkat Program Studi (Studi Kasus: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung)

¹Gofariyanti Dharmaningtyas dan ²Tristiyanto

^{1,2} Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, Jl. Prof. Dr. Ir. Sumantri Bojonegoro No.1, Gedong Meneng, Rajabasa, Kota Bandar Lampung, 35141, Indonesia
e-mail: ¹gofariyanti@gmail.com, ²tristiyanto.1981@fmipa.unila.ac.id

Abstract — PerBAN-PT No. 5 concerning APS Instruments, Accreditation is an assessment activity to determine the eligibility of a Study Program. The purpose of accreditation is to be able to determine the eligibility of a Study Program based on criteria that refer to the National Higher Education Standards and to guarantee the quality of the Study Program externally both in the academic and non-academic fields. Accreditation status is a benchmark for the performance of the study program concerned and describes the quality, efficiency, and relevance of an organized study program. The person in charge of assessing is called an auditor; the auditor needs information about the study program obtained from completing the accreditation assessment instrument by first making a study program self-evaluation document. In practice, staff must collect data along with evidence to the auditor to carry out an assessment in accordance with the standards set by BAN-PT, but study program staff in data collection still often experience data being scattered from one another, because the data collected is not small. This also sometimes results in less than optimal data collection to the auditor so that it can reduce the assessment. This problem can be overcome by making an Audit Information System so that data collection can be more organized. This information system was built using the prototype method with three stages, namely Listen to Customer, Build/Revise Mock Up, and Test Drive Mock-Up. This system uses the Laravel framework while for coding this system uses the PHP programming language. The testing method used is the Black Box Testing method with the Equivalence Partitioning type. The result of this study is a Web-Based Internal Audit Information System that can facilitate the internal auditing of a study program.

Keywords: Black-box testing; Information system; Internal audit; Prototyping; Study Program.

1. PENDAHULUAN

Program studi adalah kesatuan rencana belajar sebagai pedoman penyelenggaraan pendidikan akademik atau profesional yang diselenggarakan atas dasar suatu kurikulum serta ditujukan agar mahasiswa dapat menguasai pengetahuan, keterampilan, dan sikap sesuai dengan sasaran kurikulum. Akreditasi adalah kegiatan penilaian untuk menentukan kelayakan suatu Program Studi. Tujuan dari akreditasi adalah agar dapat menentukan kelayakan Program Studi berdasarkan kriteria yang mengacu pada Standar Nasional Perguruan Tinggi dan menjamin mutu Program Studi secara eksternal baik di bidang akademik maupun non-akademik. Proses akreditasi program studi dapat dikatakan sebagai suatu proses evaluasi terhadap program studi. Evaluasi adalah penilaian terhadap sesuatu, yang dalam hal akreditasi ini adalah kinerja program studi atau universitas. Kinerja adalah gambaran mengenai tingkat pencapaian pelaksanaan suatu kegiatan/program/kebijakan dan mewujudkan sasaran, tujuan, misi, dan visi organisasi yang tertuang dalam perumusan skema strategis (strategic planning) suatu organisasi [1][2].

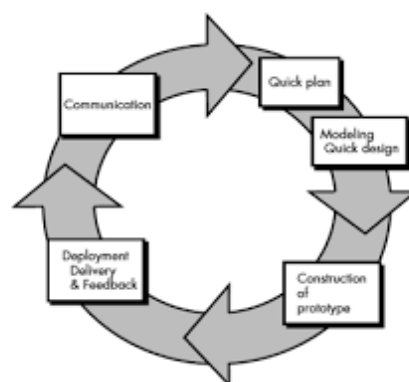
Proses akreditasi terdiri dari tiga tahapan yaitu evaluasi data dan informasi, penetapan status akreditasi, dan pemantauan status akreditasi. Akreditasi diselenggarakan dengan menugaskan asesor yang ahli di bidangnya untuk melakukan penilaian terhadap borang atau instrumen penilaian akreditasi yang diajukan oleh program studi. Di era ini, penerapan teknologi dalam menunjang kegiatan akreditasi juga semakin sering diterapkan, salah satunya dengan menggunakan sistem informasi. Sebuah sistem informasi dapat didefinisikan secara teknis sebagai sekumpulan komponen yang saling terkait yang mengumpulkan (atau mengambil), memproses,

menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam suatu organisasi [3].

Terdapat beberapa penelitian terkait penerapan sistem informasi pada proses akreditasi. Penelitian [4] mengangkat topik simulasi akreditasi program studi berbasis web dan disesuaikan dengan iaps versi 4.0. Hasilnya adalah disimpulkan bahwa aplikasi simulasi akreditasi program studi berbasis web dapat dibuat dengan menggunakan perhitungan dari *file excel* simulasi akreditasi program studi dan dengan mengikuti arahan penilaian akreditasi dari *file* lampiran 1 Instrumen Akreditasi Program Studi 4.0. Penelitian lainnya dilakukan oleh [5] yang membahas tentang penyesuaian kondisi internal sesuai standar dari BAN-PT dengan dibangunnya sistem persiapan akreditasi menggunakan sistem *dashboard* yang mengintegrasikan keseluruhan isi data pendukung sehingga didapat informasi mengenai bagian-bagian yang kurang dan yang sudah memenuhi standar. Penelitian juga dilakukan oleh [6] dengan judul Sistem Informasi Visitasi Akreditasi Program Studi Teknik Informatika. Sistem ini dibangun dengan menggunakan metode *Waterfall* dan menghasilkan sistem informasi yang dapat berjalan dengan baik dan mencapai tingkat kepuasan pengguna lebih dari 50%. Berdasarkan uraian tersebut, secara keseluruhan mengangkat topik untuk mendapat akreditasi yang baik maka program studi harus memberikan yang terbaik juga. Salah satunya dengan didukung alat bantu untuk mengukur akreditasi tersebut. Hal tersebut juga melandasi penelitian ini karena dibutuhkannya alat bantu dan belum adanya pengelolaan sistem dalam menunjang akreditasi pada Program Studi di Universitas Lampung.

2. METODOLOGI PENELITIAN

Metode pengembangan sistem yang digunakan pada pengembangan sistem adalah metode *Prototyping* yang ditunjukkan pada Gambar 1. *Prototype* didefinisikan sebagai satu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, serta bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. Dasar dari pemikiran ini adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam, lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat. Semua rancangan diagram atau model yang dibuat tidak diharuskan telah sempurna dan final dalam pendekatan *prototype* [7]. Metode *prototype* sesuai untuk menjelaskan kebutuhan pengguna secara lebih rinci karena pengguna sering mengalami kesulitan dalam penyampaian kebutuhannya secara detail tanpa melihat gambaran yang jelas.



Gambar 1. Metode *prototyping*.

Pada awal tahapan perencanaan dan perancangan sistem, digunakan *unified modelling language* (UML). *Unified modelling language* adalah salah satu *tool/model* untuk merancang pengembangan *software* yang berbasis object-oriented. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software* [8]. Setelah analisis dan perancangan dengan UML selesai, selanjutnya akan dilakukan desain tampilan *user interface* sistem dengan menggunakan *mockup*.

Mockup adalah sketsa atau rancangan awal dari sebuah desain web yang dibuat menggunakan *software* pengedit gambar atau pembuat mockup. *Mockup* berfungsi sebagai acuan kerja pembuatan *website* agar tidak menyimpang dari tujuan awal membuatnya [9]. Di akhir sebelum sistem dirilis, dilakukan pengujian sistem dengan metode *black-box testing*. *Black-box testing* adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [10].

3. HASIL DAN PEMBAHASAN

Tahapan pengerjaan sistem dengan menggunakan metode prototyping adalah *listen to customer* (mendengarkan pelanggan), *bulid/revise mock-up* (membangun/memperbaiki *mock-up*), dan *customer test-drives mock-up* (melihat/menguji *mock-up*).

3.1. *Listen to customer* (mendengarkan pelanggan)

Pada tahap ini akan dilakukan proses analisis kebutuhan sistem berdasarkan kebutuhan pelanggan. Berikut ini adalah yang disampaikan pelanggan mengenai kebutuhan sistem.

- a. Pertemuan pertama dilakukan pembahasan kebutuhan dan cakupan sistem. Pembahasan kebutuhan sistem dilakukan melalui diskusi. Hasilnya didapatkan tiga menu utama yang nanti ada di sistem yaitu, menu pengguna, menu LED (Laporan Evaluasi Diri) & LKPS (Laporan Kinerja Program Studi), dan menu borang penilaian.
- b. Pertemuan kedua ada penambahan pada menu untuk bukti di setiap data LED dan LKPS, serta bukti pada data LED diminta untuk dapat berupa *file voice record*. Adanya penambahan data evaluasi program studi yaitu data IKU (Indikator Kinerja Utama).
- c. Pertemuan ketiga adanya penambahan menu berita acara, catatan pada menu berita acara diambil dari kolom catatan pada borang penilaian. Penambahan perhitungan penilaian di kolom capaian dan persen kinerja. Penambahan *dropdown* jenis borang, borang BAN-PT S1, S2, S3, dan D3.

Perubahan pada bagian LKPS beberapa tabel dengan kolom bernama TS diubah menjadi tahun. Penentuan kebutuhan *customer* juga berlandaskan pada IAPS 4.0. IAPS (Instrumen Akreditasi Program Studi) 4.0 merupakan instrument akreditasi program studi baru menggantikan yang sebelumnya. Perbedaan paling mendasar dari perubahan ke IAPS 4.0 adalah menggunakan 9 kriteria, sebelumnya hanya menggunakan 7 kriteria IAPS 4.0 memecah kriteria nomor 7 menjadi 2 yaitu penelitian dan pengabdian kepada masyarakat, juga memunculkan satu kriteria baru yaitu luaran dan capaian tridharma. IAPS 4.0 terdiri dari dua bagian besar, yaitu LED (Laporan Evaluasi Diri) dan LKPS (Laporan Kinerja Program Studi). LED menggambarkan status dan analisis capaian dari masing-masing kriteria. Unit pengelola program studi diharapkan mampu mengenali kekuatan yang dimiliki serta aspek apa saja yang perlu diperbaiki. Sementara LKPS memuat capaian indikator kinerja program studi yang secara bertahap diintegrasikan dengan PD-DIKTI [11]. Berdasarkan kebutuhan *customer* dan IAPS tersebut, maka diperoleh kebutuhan fungsional dan nonfungsional sebagai berikut.

3.1.1. Kebutuhan Fungsional

- a. Sistem dapat menyimpan data LED, LKPS, dan IKU.
- b. Sistem dapat menampilkan data borang penilaian yang telah diisi asesor.
- c. Sistem dapat menyambungkan data fakultas dan program studi.
- d. Sistem dapat melakukan proses simpan, edit dan hapus dalam setiap borang penilaian.
- e. Sistem dapat melakukan cetak borang penilaian dan berita acara.

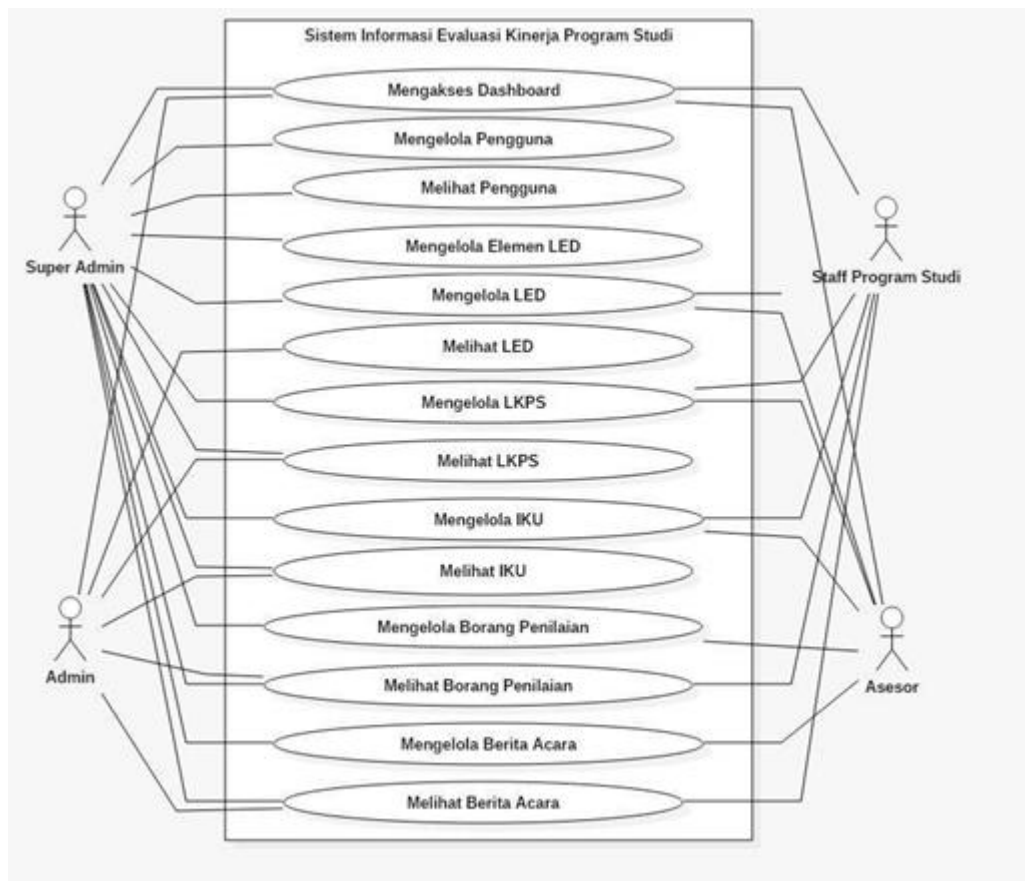
3.1.2. Kebutuhan Nonfungsional

- a. Sistem dapat dijalankan oleh beberapa *web browser* diantaranya Google Chrome dan Mozilla Firefox.

- b. Sistem dibuat dengan desain sederhana dan mudah digunakan.

3.2. *Bulid/revise mock-up* (membangun/memperbaiki *mock-up*)

Pada tahap ini dilakukan proses desain dan *coding*/pengkodean yang sesuai dengan permintaan dari *customer*. Pada Gambar 2 *use case diagram* terdapat empat level aktor dalam sistem informasi evaluasi kinerja program studi yaitu *super administrator*, *administrator*, asesor dan *staff* program studi. Detail masing-masing aktor tersedia pada Tabel 1.



Gambar 2. *Usecase* diagram sistem informasi evaluasi kinerja program studi.

Tabel 1. Aktor yang ada pada sistem informasi evaluasi kinerja program studi.

No	Aktor	Keterangan
1	<i>Super Administrator</i>	Aktor yang mengelola keseluruhan sistem. Aktor dapat mengakses dashboard, mengelola data pengguna, data LED, data LKPS, data IKU, data borang penilaian dan berita acara.
2	<i>Administrator</i>	Aktor yang hanya dapat melihat semua data di setiap program studi tanpa dapat mengubahnya.
3	Asesor	Aktor yang hanya dapat melihat data LED, LKPS, IKU yang telah diisi <i>staff</i> program studi dan memberikan penilaian dan berita acara.
4	<i>Staff</i> Program Studi (Prodi)	Aktor dalam sistem yang dapat mengisi data setiap program studi.

a. Penulisan Kode Program

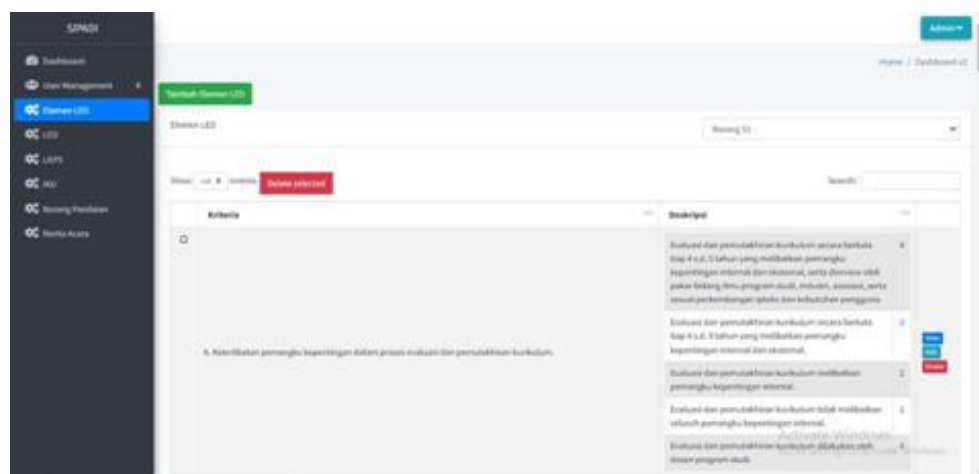
Penulisan kode program menggunakan bahasa PHP, HTML, Javascript, serta dengan menggunakan *framework* Laravel. Laravel adalah sebuah MVC *web development framework* PHP yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintaks yang bersih dan set fungsional yang dapat mengurangi banyak waktu untuk implementasi [12].

b. Implementasi Sistem

Pada implementasi sistem, dihasilkan sistem dengan beberapa halaman yang tersedia. Halaman *login* menjadi halaman awal yang harus dilalui oleh setiap level *user* untuk masuk ke sistem, baik *super administrator*, *administrator*, *staff* prodi, maupun asesor. Gambar 3 menampilkan tampilan halaman login.

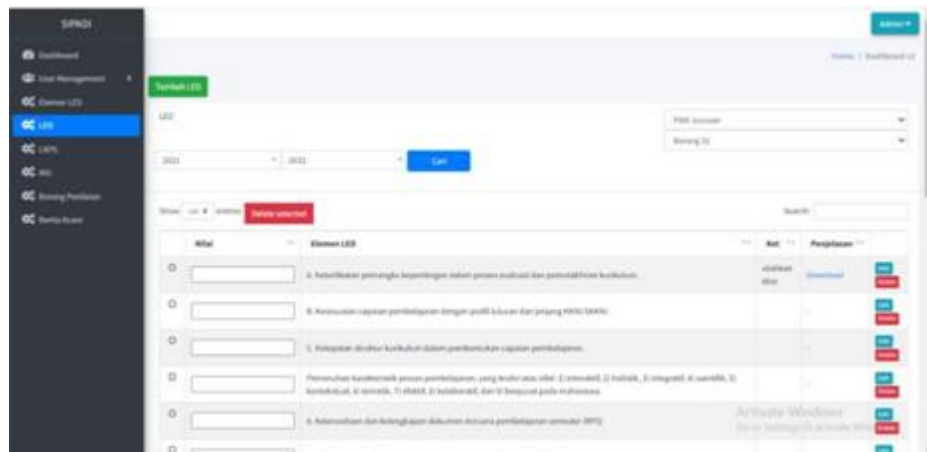
Gambar 3. Tampilan halaman login sistem informasi evaluasi program studi (SIEVAL).

Saat *login* sebagai *super administrator*, maka dapat mengakses halaman elemen LED yang berisi kriteria dan deskripsi pada setiap data LED. Gambar 4 menunjukkan tampilan halaman elemen LED pada level user *super administrator*.

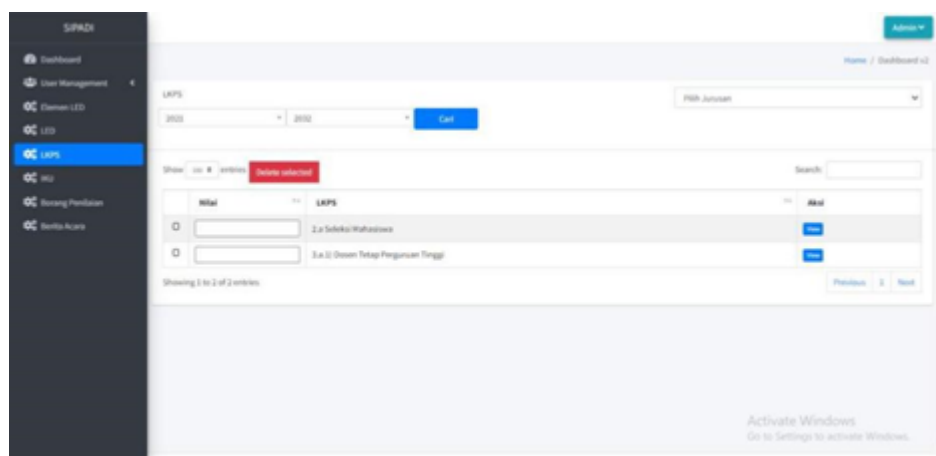


Gambar 4. Tampilan menu halaman elemen LED pada *super administrator*.

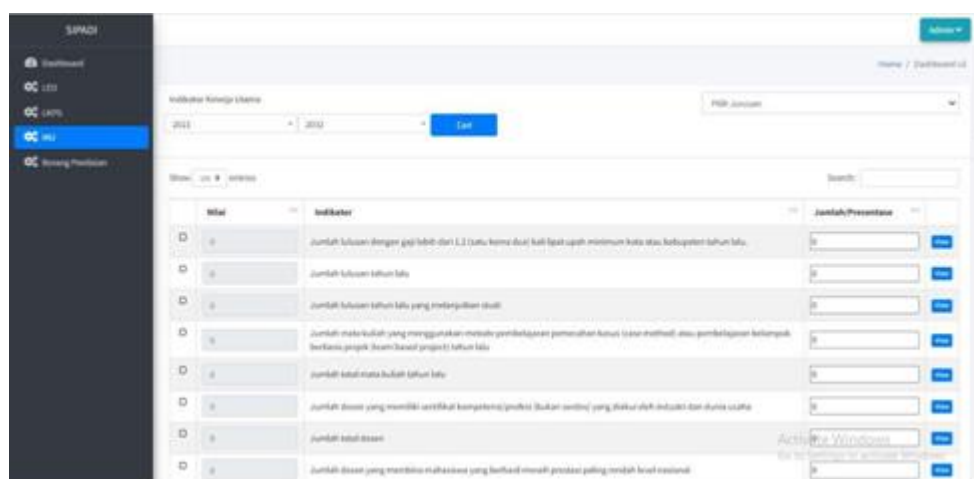
Pada halaman LED *super administrator*, dapat dilakukan proses menambah, mengubah, dan menghapus data LED yang dapat dilihat pada Gambar 5. Hal serupa juga dapat dilakukan oleh *super administrator* di halaman LKPS yang dapat dilihat pada Gambar 6. Pada halaman IKU level *administrator*, *user* dapat melihat data IKU untuk setiap program studi seperti yang ditampilkan pada Gambar 7.



Gambar 5. Tampilan halaman LED pada *super administrator*.

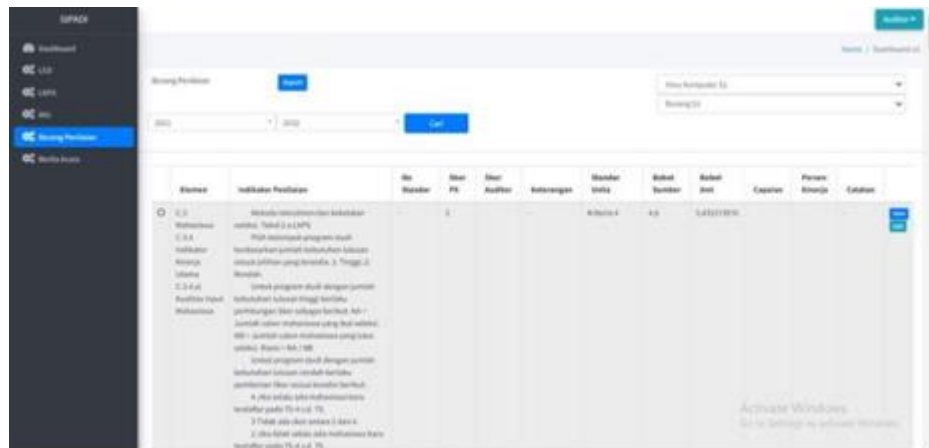


Gambar 6. Tampilan halaman LKPS pada *super administrator*.

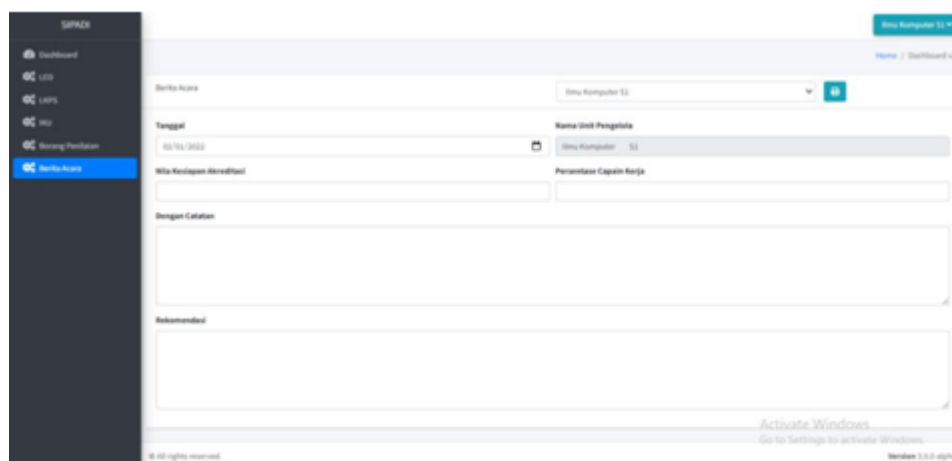


Gambar 7. Tampilan halaman IKU pada *administrator*.

Pada halaman borang penilaian untuk *user* asesor, asesor dapat mengubah data borang penilaian untuk setiap program studi yang sedang dilakukan penilaian. Tampilan halaman borang penilaian tersedia pada Gambar 8. Bagi *staff* program studi, saat penilaian dapat melihat berita acara yang sudah diisi oleh asesor untuk program studinya yang tampilannya dapat dilihat pada Gambar 9.



Gambar 8. Tampilan halaman borang penilaian asesor.



Gambar 9. Tampilan halaman berita acara *staff* program studi.

3.3. Test drives mockup (menentukan apakah *prototype* dapat diterima)

Untuk menentukan apakah *prototype* dapat diterima atau tidak maka perlu dilakukan pengujian sistem. Tahapan ini dilakukan pengujian sistem yang telah dibuat kepada pengguna. Pengujian bertujuan untuk mengetahui fitur berjalan dengan baik atau tidak. Pengujian pada penelitian ini menggunakan metode *black-box testing* dengan teknik *equivalence partitioning*. Teknik ini menguji fungsionalitas sistem berjalan sebagaimana mestinya atau tidak.

1) Pengujian Sistem Level *Super Administrator*

Tabel 2. Skenario pengujian level *super administrator*.

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
1	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar pada form <i>login</i> .	Berhasil masuk ke sistem dan menampilkan halaman <i>dashboard</i> .	Berhasil
2	Mengakses halaman <i>dashboard</i> dengan	Menampilkan pesan kesalahan.	Berhasil

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
	memasukkan <i>username</i> dan <i>password</i> yang salah pada form <i>login</i> .		
3	Klik tombol <i>logout</i> .	Berhasil keluar dari sistem.	Berhasil
4	Menambah data <i>user</i> .	Berhasil menambahkan data.	Berhasil
5	Mengubah data <i>user</i> .	Berhasil mengubah data.	Berhasil
6	Menghapus data <i>user</i> .	Berhasil menghapus data.	Berhasil
7	Menambah elemen LED di setiap borang.	Berhasil menambahkan data.	Berhasil
8	Mengubah elemen LED di setiap borang.	Berhasil mengubah data.	Berhasil
9	Menghapus elemen LED di setiap borang.	Berhasil menghapus data.	Berhasil
10	Mengakses halaman dengan klik menu LED pada panel navigasi sistem.	Sistem berhasil menampilkan halaman LED.	Berhasil
11	Mengakses halaman dengan klik tombol tambah LED di setiap borang.	Sistem dapat menampilkan <i>form</i> tambah LED.	Berhasil
12	Mengisi <i>form</i> LED sesuai dengan format <i>input</i> dan menekan tombol simpan.	Sistem berhasil menyimpan data dan kembali ke halaman LED.	Berhasil
13	Mengubah data LED dengan klik tombol edit di setiap jurusan.	Sistem dapat menampilkan halaman edit LED sesuai data yang dipilih.	Berhasil
14	Mengubah data LED lalu klik tombol simpan.	Sistem berhasil menyimpan perubahan data.	Berhasil
15	Mengakses halaman detail LED dengan klik tombol <i>view</i> .	Sistem dapat menampilkan halaman detail data LED yang dipilih.	Berhasil
16	Menekan tombol hapus pada data LED yang dipilih.	Sistem menampilkan <i>pop up</i> konfirmasi penghapusan data.	Berhasil
17	Menghapus data dengan klik tombol hapus pada <i>pop up</i> .	Data yang dipilih terhapus dan kembali ke halaman LED.	Berhasil
18	Membatalkan penghapusan data dengan klik tombol batal pada <i>pop up</i> .	Sistem mengembalikan ke halaman LED.	Berhasil
19	Mengakses halaman dengan klik <i>tab</i> menu LKPS pada panel navigasi sistem.	Sistem dapat menampilkan halaman LKPS.	Berhasil
20	Mengakses tabel LKPS dengan memilih jurusan lalu klik tombol <i>view</i> .	Sistem menampilkan halaman tabel LKPS yang dipilih.	Berhasil
21	Mengakses halaman dengan klik tambah LKPS.	Sistem dapat menampilkan <i>form</i> tambah LKPS.	Berhasil
22	Mengisi <i>form</i> sesuai dengan format <i>input</i> dan klik tombol simpan.	Sistem berhasil menyimpan data dan kembali ke halaman LKPS.	Berhasil
23	Mengisi <i>form</i> dengan mengosongkan semua <i>field</i> dan klik tombol simpan.	Sistem tidak berhasil menyimpan data dan menampilkan pesan <i>error</i> .	Berhasil
24	Mengakses halaman <i>edit</i> dengan klik tombol <i>edit</i> pada kolom aksi data yang dipilih.	Sistem dapat menampilkan halaman <i>edit</i> LKPS sesuai data yang dipilih.	Berhasil
25	Mengubah data LKPS sesuai format dan	Sistem berhasil menyimpan perubahan	Berhasil

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
	klik tombol simpan.	data dan kembali ke halaman LKPS.	
26	Mengubah data LKPS dengan mengosongkan field dan klik tombol simpan.	Sistem tidak berhasil menyimpan perubahan data dan menampilkan pesan <i>error</i> .	Berhasil
27	Mengakses halaman detail dengan klik tombol <i>view</i> .	Sistem dapat menampilkan halaman detail data LKPS yang dipilih.	Berhasil
28	Klik tombol hapus pada data LKPS yang dipilih.	Sistem menampilkan <i>pop up</i> konfirmasi penghapusan data.	Berhasil
29	Menghapus data dengan klik tombol hapus pada <i>pop up</i> .	Data yang dipilih terhapus dan kembali ke halaman LKPS.	Berhasil
30	Membatalkan penghapusan data dengan klik tombol batal pada <i>pop up</i> .	Sistem mengembalikan ke halaman LKPS.	Berhasil
31	Mengakses halaman dengan klik menu IKU pada panel navigasi sistem..	Sistem berhasil menampilkan halaman IKU.	Berhasil
32	Mengakses halaman dengan klik tombol tambah IKU di setiap borang.	Sistem dapat menampilkan <i>form</i> tambah IKU.	Berhasil
33	Mengisi <i>form</i> IKU sesuai dengan format <i>input</i> dan menekan tombol simpan.	Sistem berhasil menyimpan data dan kembali ke halaman IKU.	Berhasil
34	Mengubah data IKU di setiap jurusan.	Berhasil mengubah data IKU.	Berhasil
35	Mengakses halaman detail IKU dengan klik tombol <i>view</i> .	Sistem dapat menampilkan halaman detail data IKU yang dipilih.	Berhasil
36	Menekan tombol hapus pada data IKU yang dipilih.	Sistem menampilkan <i>pop up</i> konfirmasi penghapusan data.	Berhasil
37	Menghapus data IKU dengan klik tombol hapus pada <i>pop up</i> .	Data yang dipilih terhapus dan kembali ke halaman IKU.	Berhasil
38	Membatalkan penghapusan data IKU dengan klik tombol batal pada <i>pop up</i> .	Sistem mengembalikan ke halaman IKU.	Berhasil
39	Mengakses halaman dengan klik menu Borang Penilaian pada panel navigasi sistem	Sistem berhasil menampilkan halaman Borang Penilaian.	Berhasil
40	Mengakses halaman dengan klik tombol tambah Borang Penilaian di setiap borang	Sistem dapat menampilkan <i>form</i> tambah Borang Penilaian.	Berhasil
41	Mengisi <i>form</i> Borang Penilaian sesuai dengan format <i>input</i> dan klik tombol simpan.	Sistem berhasil menyimpan data dan kembali ke halaman Borang Penilaian.	Berhasil
42	Mengubah data Borang Penilaian dengan klik tombol edit di setiap jurusan.	Sistem dapat menampilkan halaman edit Borang Penilaian sesuai data yang dipilih.	Berhasil
43	Mengubah data Borang Penilaian lalu klik tombol simpan.	Sistem berhasil menyimpan perubahan data Borang Penilaian	Berhasil
44	Mengakses halaman detail Borang Penilaian dengan klik tombol <i>view</i> .	Sistem dapat menampilkan halaman detail data Borang Penilaian yang dipilih.	Berhasil
45	Menekan tombol hapus pada data Borang Penilaian yang dipilih.	Sistem menampilkan <i>pop up</i> konfirmasi penghapusan data.	Berhasil
46	Menghapus data Borang Penilaian dengan klik tombol hapus pada <i>pop up</i> .	Data yang dipilih terhapus dan kembali ke halaman Borang Penilaian.	Berhasil
47	Membatalkan penghapusan data Borang Penilaian dengan klik tombol batal pada <i>pop up</i> .	Sistem mengembalikan ke halaman Borang Penilaian.	Berhasil
48	Mengakses halaman dengan klik menu Berita Acara pada panel navigasi sistem.	Sistem berhasil menampilkan halaman Berita Acara.	Berhasil

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
49	Mengubah <i>form</i> berita acara di setiap jurusan lalu klik simpan	Berhasil menyimpan berita acara	Berhasil

2) Pengujian Level *Administrator*

Tabel 3. Skenario pengujian level *administrator*.

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
1	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar pada form <i>login</i> .	Berhasil masuk ke sistem dan menampilkan halaman <i>dashboard</i> .	Berhasil
2	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang salah pada form <i>login</i> .	Menampilkan pesan kesalahan.	Berhasil
3	Klik tombol <i>logout</i> .	Berhasil keluar dari sistem.	Berhasil
4	Melihat data <i>user</i> .	Sistem menampilkan data <i>user</i> .	Berhasil
5	Melihat elemen LED di setiap borang.	Sistem menampilkan data elemen LED.	Berhasil
6	Melihat LED di setiap jurusan.	Sistem berhasil menampilkan data LED di setiap jurusan.	Berhasil
7	Melihat LKPS di setiap jurusan.	Sistem dapat menampilkan data LKPS di setiap jurusan.	Berhasil
8	Melihat IKU di setiap jurusan.	Sistem dapat menampilkan data IKU di setiap jurusan.	Berhasil
9	Melihat borang penilaian di setiap jurusan.	Sistem dapat menampilkan data borang.	Berhasil
10	Melihat berita acara di setiap jurusan.	Sistem dapat menampilkan data berita.	Berhasil

3) Pengujian Level *Staff* Program Studi

Tabel 4. Skenario pengujian level *staff* program studi.

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
1	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar pada form <i>login</i> .	Berhasil masuk ke sistem dan menampilkan halaman <i>dashboard</i> .	Berhasil
2	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang salah pada form <i>login</i> .	Menampilkan pesan kesalahan.	Berhasil
3	Klik tombol <i>logout</i> .	Berhasil keluar dari sistem.	Berhasil
4	Memilih tipe borang lalu mengisi dan Meng- <i>upload</i> data LED, klik simpan.	Sistem berhasil menyimpan data LED.	Berhasil
5	Mengisi data LKPS lalu klik simpan.	Sistem berhasil menyimpan data LKPS.	Berhasil
6	Mengisi data IKU.	Sistem berhasil menyimpan data IKU.	Berhasil
7	Mengisi kolom skor PS di borang penilaian.	Sistem berhasil menyimpan data borang penilaian.	Berhasil
8	Melihat berita acara.	Sistem berhasil menampilkan berita.	Berhasil

4) Pengujian Level Asesor

Tabel 5. Skenario pengujian level *asesor*.

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
1	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar pada form <i>login</i> .	Berhasil masuk ke sistem dan menampilkan halaman <i>dashboard</i> .	Berhasil

No.	Skenario pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh
2	Mengakses halaman <i>dashboard</i> dengan memasukkan <i>username</i> dan <i>password</i> yang salah pada form <i>login</i> .	Menampilkan pesan kesalahan.	Berhasil
3	Klik tombol <i>logout</i> .	Berhasil keluar dari sistem.	Berhasil
4	Mengisi nilai pada menu LED sesuai pilihan jurusan.	Sistem berhasil menyimpan nilai.	Berhasil
5	Mengisi nilai pada menu LKPS sesuai pilihan jurusan.	Sistem berhasil menyimpan nilai.	Berhasil
6	Mengisi nilai pada menu IKU sesuai pilihan jurusan.	Sistem berhasil menyimpan nilai.	Berhasil
7	Mengisi data borang penilaian sesuai pilihan jurusan dan pilihan jenis borang.	Sistem berhasil menyimpan data.	Berhasil
8	Mengisi berita acara.	Sistem berhasil menyimpan data.	Berhasil

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah diuraikan maka dapat disimpulkan bahwa sistem informasi evaluasi kinerja program studi di Universitas Lampung menggunakan *framework* Laravel berhasil dibuat dan dapat digunakan sebagaimana mestinya. Pembuatan sistem dilakukan dengan metode *prototype*. Sistem informasi evaluasi kinerja program studi di Universitas Lampung mampu diimplementasikan dengan baik sesuai dengan hasil pengujian yang dilakukan dengan metode *black box testing*. *Staff* program studi dapat mengisi data dengan mudah dengan memilih *template form* sesuai dengan strata program studinya. Asesor dapat melakukan penilaian secara lebih mudah hanya dengan memilih prodi mana yang akan dinilai, data yang diberikan oleh *staff* program studi pastinya tersampaikan ke asesor, terdapat perhitungan kinerja dan persen secara otomatis, sehingga memudahkan asesor. Sistem informasi evaluasi kinerja program studi di Universitas Lampung dibuat mengikuti pedoman Peraturan BAN-PT No. 5 Tahun 2019 tentang Instrumen APS.

DAFTAR PUSTAKA

- [1] D. K. Sofyan, "Pengaruh Lingkungan Kerja Terhadap Kinerja Kerja Pegawai BAPPEDA," Malikussaleh Industrial Engineering, 2013.
- [2] M. Abdullah, "Manajemen dan Evaluasi Kinerja Karyawan," In Perpustakaan Nasional RI, 2014.
- [3] J. P. L. Laudon & K. C. Laudon "Management Information Systems: Managing the Digital Firm, Global Edition." In International Journal of Information Management, 2017.
- [4] J. W. Supit, V. Tulenan, dan S. R. Sentinuwo, "Rancang Bangun Simulasi Akreditasi Program Studi Berbasis Web", *Jurnal Teknik Informatika*, 15(3), 163-170, 2020, <https://doi.org/10.35793/jti.15.3.2020.29669>.
- [5] A. A. Raditya, Kartono, dan I. K. Raharjana, "Sistem Dashboard Untuk Persiapan Akreditasi Program Studi Sarjana Berdasarkan Standar BAN-PT", *Sriwijaya Journal of Information Systems*, 8(1), 871-882, 2016, <https://doi.org/10.36706/jsi.v8i1.2956>.
- [6] R. I. Laelaningrum, Y. Islamiati, P. Raharjo, dan M. Mardiyono, "Sistem Informasi Visitasi Akreditasi Program Studi Teknik Informatika," *Transmisi: Jurnal Ilmiah Teknik Elektro*, 20(3), 120-126, 2018, <https://doi.org/10.14710/transmisi.20.3.120-126>.

- [7] W. N. Cholifah, Y. Yulianingsih dan S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, <https://doi.org/10.30998/string.v3i2.3048>, 2018.
- [8] F. Sonata, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer," *Jurnal Komunika: Jurnal Komunikasi, Media dan Informatika*, 2019, <https://doi.org/10.31504/komunika.v8i1.1832>.
- [9] S. Nurajizah, "Sistem Informasi Perpustakaan Berbasis Web dengan Metode Prototype: Studi Kasus Sekolah Islam Gema Nurani Bekasi," *Seminar Nasional Inovasi dan Tren (SNIT)*, 1(1), 2015.
- [10] D. A. Ahmadi dan W. Wardhana, "Rancang Bangun Aplikasi Search Engine Repository Software Desktop Menggunakan Teknik Web Extraction," *Jurnal Komputasi*, 4(1), 139-147, 2016, <http://dx.doi.org/10.23960%2Fkomputasi.v4i1.1182>.
- [11] I. Ulum, "Intellectual Capital Framework Perguruan Tinggi di Indonesia Berdasarkan Instrumen Akreditasi Program Studi (IAPS) 4.0," *Jurnal Reviu Akuntansi dan Keuangan*, 9(3), 309-318, 2019, <https://doi.org/10.22219/jrak.v9i3.10227>.
- [12] B. P. Widodo dan H. D. Purnomo, "Perancangan Aplikasi Pencarian Layanan Kesehatan Berbasis Html 5," *Jurnal Sistem Komputer*, 6(1), 2016.