



TUGAS AKHIR - KS 141501

**RANCANG BANGUN APLIKASI SISTEM PENJAMINAN
MUTU BERBSAIS BORANG AKREDITASI BAN-PT
DENGAN FORM DINAMIS PADA LINGKUNGAN BASIS
DATA RELASIONAL**

***ARCHITECTURE-BASED QUALITY ASSURANCE SYSTEM
APPLICATION FORM OF ACCREDITATION BAN-PT
WITH DYNAMIC FORM ON THE RELATIONAL
DATABASE ENVIRONMENT***

HARUN RIZAL
NRP 5213 100 125

Dosen Pembimbing :
Radityo Prasetianto.W, S.Kom, M.Kom
Nisfu Asrul Sani, S.Kom, M.Sc

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

RANCANG BANGUN APLIKASI SISTEM PENJAMINAN MUTU BERBSAIS BORANG AKREDITASI BAN-PT DENGAN FORM DINAMIS PADA LINGKUNGAN BASIS DATA RELASIONAL)

HARUN RIZAL
NRP 5213 100 125

Dosen Pembimbing :
Radityo Prasetianto, W, S.Kom, M.Kom
Nisfu Asrul Sani, S.Kom, M.Sc

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[halaman ini sengaja dikosongkan]



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 141501

***ARCHITECTURE-BASED QUALITY ASSURANCE SYSTEM
APPLICATION FORM OF ACCREDITATION BAN-PT
WITH DYNAMIC FORM ON THE RELATIONAL
DATABASE ENVIRONMENT***

HARUN RIZAL
NRP 5213 100 125

Supervisor :
Radityo Prasetianto.W, S.Kom, M.Kom
Nisfu Asrul Sani, S.Kom, M.Sc

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI SISTEM PENJAMINAN MUTU BERBASIS BORANG AKREDITASI BAN-PT DENGAN FORM DINAMIS PADA LINGKUNGAN BASIS DATA RELASIONAL

TUGAS AKHIR

Dibuat Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

HARUN RIZAL
NRP. 5213 100 125

Sidikaya, 17 Juli 2017

KEPALA DEPARTEMEN SISTEM INFORMASI



Dr. Ir. Aris Tjahyanto, M.Kom
NIP.19650310 199102 1 001

[halaman ini sengaja dikosongkan]

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI SISTEM PENJAMINAN MUTU BERBASIS BORANG AKREDITASI BAN-PT DENGAN FORM DINAMIS PADA LINGKUNGAN BASIS DATA RELASIONAL

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Meng peroleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

HABUN REZAL
NRP. 5213 100 125

Disetujui Tim Penguji : Tanggal Ujian : 10 Juli 2017
Periode Wisuda : September 2017

Radhya Prasasti W., S.Kom., M.Kom


(Pembimbing I)

Niwa Azzul Sari, S.Kom., M.Sc


(Pembimbing II)

Renny Pradina, S.T., M.T


(Penguji I)

Nur Anil H., S.Kom., M.Sc.Eng., Ph.D


(Penguji II)

[halaman ini sengaja dikosongkan]

**RANCANG BANGUN APLIKASI SISTEM
PENJAMINAN MUTU BERBSAIS BORANG
AKREDITASI BAN-PT DENGAN FORM DINAMIS
PADA LINGKUNGAN BASIS DATA RELASIONAL
(STUDI KASUS: SPMI ITS)**

Nama Mahasiswa : Harun Rizal
NRP : 5213100125
Jurusan : Sistem Informasi FTIf - ITS
Dosen Pembimbing 1 : Radityo Prasetyanto W, S.Kom,
M.Kom
Dosen Pembimbing 2 : Nisfu Asrul Sani, S.Kom, M.Sc

ABSTRAK

Salah satu syarat satuan pendidikan dapat mengeluarkan sertifikat ialah terdapatnya akreditasi baik ditingkat institusi maupun di tingkat program studi. Akreditasi ini dilakukan salah satunya melalui asesmen terhadap borang akreditasi oleh tim asesor yang terdiri atas berbagai keahlian terkait.

Institut Teknologi Sepuluh Nopember memiliki Sistem Penjaminan Mutu Internal (SPMI) yang sudah berjalan beberapa tahun. Namun pada praktiknya terdapat beberapa permasalahan yang terjadi mulai dari kesulitan pengguna dalam pemakaian nya hingga kekurangan sistem dalam melakukan otomasi penilaian kualitas isian borang. Penilaian ini berupa range skor yang diberikan terhadap isian berdasarkan indikator-indikator yang sudah ditentukan sebelumnya serta perubahan kebijakan yang berdampak pada perubahan format isian akreditasi sehingga dibutuhkan

aplikasi yang mampu dikustomisasi secara dinamis tanpa mengubah kode program aplikasinya.

Tugas akhir ini membuat form dinamis pada lingkungan basis data relasional dengan tujuan untuk membangun aplikasi otomasi penilaian kualitas isian borang; membandingkan penilaian borang baik dari antar jurusan maupun perkembangan dari tahun ke tahun; dan integrasi data isian borang dalam format yang mudah diakses, dicetak, dan di review tanpa mengorbankan kenyamanan user dalam melakukan penginputan data. Teknologi pada tugas akhir memanfaatkan Javascript-layer diatas basis data relasional pada framework Laravel sehingga dapat membentuk form yang dinamis.

Kata Kunci: Accreditation, Dynamic Form, Role-based System

KATA PENGANTAR

Puji syukur penulis haturkan ke hadirat Allah SWT yang telah memberikan anugerah dan hidayah kepada penulis sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Aplikasi Sistem Penjaminan Mutu Berbasis Borang Akreditasi BAN-PT Menggunakan Role-based System” sebagai salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi nyata bagi kampus Sistem Informasi, ITS, dan bangsa Indonesia.

Penyusunan tugas akhir ini senantiasa mendapatkan dukungan dari berbagai pihak baik dalam bentuk doa, motivasi, semangat, kritik, saran dan berbagai bantuan lainnya. Untuk itu, secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam dalamnya kepada:

1. Allah SWT yang telah memberikan kesehatan, kemudahan, kelancaran dan kesempatan untuk penulis hingga dapat menyelesaikan Tugas Akhir ini.
2. Segenap keluarga besar terutama kedua orang tua, kakak dan adik penulis, Bapak Fachrizal (Alm), Ibu Haslina, Adnan Rizal, Lukman Rizal dan Ahmad Musa Rizal yang senantiasa mendoakan, memberikan motivasi dan semangat, sehingga penulis mampu menyelesaikan pendidikan S1 ini dengan baik.
3. Bapak Dr. Ir. Aris Tjahyanto, M.Kom. selaku Ketua Jurusan Sistem Informasi ITS serta seluruh dosen pengajar beserta staf dan karyawan di Jurusan Sistem Informasi, FTIF ITS Surabaya selama penulis menjalani kuliah.

4. Bapak Radityo Prasetianto Wibowo, S.Kom, M.Kom. selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan mendukung dengan memberikan ilmu, petunjuk, dan motivasi dalam penyelesaian Tugas Akhir.
5. Bapak Nisfu Asrul Sani, S.Kom, M.Sc. sebagai dosen wali penulis selama menempuh pendidikan di Jurusan Sistem Informasi dan juga sebagai dosen pembimbing tugas akhir yang telah banyak meluangkan waktu untuk membimbing dan memberikan motivasi.
6. Ibu Renny Pradina, S.T, M.T serta Ibu Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D selaku dosen penguji yang telah memberikan kritik, saran dan masukan yang dapat menyempurnakan Tugas Akhir ini.
7. Teman-teman Sistem Informasi angkatan 2013 (13ELTRANIS) yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir.
8. Sahabat-sahabat dekat "Mb Ariesty & Friends", "Sekitar Kita", "Sahabat Tomgun", "UKM PENALARAN", teman-teman di "ADDI AMAN" yang senantiasa menjadi sumber semangat bagi penulis dan selalu berbagi canda dan tawa selama pengerjaan Tugas Akhir ini.
9. Kirana Gita Larasati dan Marina Safitri yang sudah sangat membantu dalam penyelesaian dokumen Tugas Akhir ini.
10. Tetha Valianta, Alvin Rahman K, Wisnu Tri S, Muhammad Zuhri, Nikolaus Herjuno, Stezar Priansya dan para juru kunci laboratorium ADDI yang selalu menemani siang dan malam saat pengerjaan Tugas Akhir ini.
11. Haikal Rusdi, Bambang Setyawan, Risa Perdana, Novi Azizah, Adnan Mauludin, I Made Kusnanta, Lutfi Nur F, dan teman-teman se-lab seperjuangan yang sama-

sama saling memberikan dukungan dalam pengerjaan Tugas Akhir ini.

12. Bintang Setyawan, Farin Reggie, Robbigh Faubendri, Shania Olivia Z, Ervi Ritya, Pramita Lucianna, Delina Rahayu, Fitri Rifqatusa'adah, Achmad Yusuf Fauzan, Edo Haidar, Umar Al Aqsho, Pri Rezki yang senantiasa meramaikan suasana laboratorium dan memberikan dukungan saat pengerjaan Tugas Akhir ini.
13. Dedi Puji, Alimul Hakim, Hanif Rusdiansyah, Zetry Prawira, Diyan Wahyu, Miftah Reza, Cahya Yunita, Alvisha Farrasita, Ramadhan Pratama, M Farhan Naufal, Febrian A Harimurti, Wahyu Aditya, Hendra Rahma W, Adlie Yudha T, Ikhwan Aziz, Fikri Basalamah, Richardo Tiono yang telah memberi dukungan dan masih banyak lagi yang tidak bisa saya sebutkan satu-persatu, Serta seluruh pihak-pihak lain yang tidak dapat disebutkan satu persatu yang telah banyak membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Terima kasih atas segala bantuan, dukungan, serta doanya. Semoga Allah SWT senantiasa melimpahkan anugerah serta membalas kebaikan yang telah diberikan kepada penulis.

Penulis menyadari bahwa masih terdapat kekurangan dalam penyusunan tugas akhir ini, oleh karena itu penulis mengharapkan saran dan kritik yang membangun demi kebaikan penulis dan tugas akhir ini. Akhir kata, penulis berharap bahwa tugas akhir ini dapat memberikan kebermanfaatan bagi perkembangan ilmu pengetahuan dan semua pihak.

Surabaya, 7 Juli 2017

Penulis

[halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR	iii
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
DAFTAR KODE PROGRAM.....	xiv
DAFTAR DIAGRAM.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian Tugas Akhir	3
1.5. Manfaat Penelitian Tugas Akhir	4
1.6. Relevansi Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Sebelumnya	5
2.1.1. Sistem Penjaminan Mutu Internal (SPMI-ITS) - 2016 5	5
2.2. Dasar Teori.....	6
2.2.1. Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) 6	6
2.2.2. Role-based Access Control	7
2.2.3. Laravel Framework.....	8

2.2.4.	SQL Server.....	10
2.2.5.	Iterative and Incremental Development	10
2.2.6.	Dynamic Database Schema	11
2.2.7.	JSON.....	12
2.2.8.	Apache JMeter.....	13
2.2.9.	Application Performance Index (APDEX).....	14
2.2.10.	System Usability Scale	15
BAB III METODOLOGI PENELITIAN		17
3.1.	Tahapan Pelaksanaan Tugas Akhir.....	17
3.2.	Uraian Metodologi.....	18
3.2.1.	Studi Literatur	18
3.2.2.	Analisa Kebutuhan Aplikasi Borang	18
3.2.3.	Perancangan dan Pengembangan Aplikasi Borang	18
3.2.4.	Penyusunan Buku TA	22
BAB IV PERANCANGAN		23
4.1.	Analisis Kebutuhan	23
4.1.1.	Kebutuhan Fungsional	23
4.1.2.	<i>Use Case Diagram</i>	25
4.2.	Pemetaan navigasi / sitemap.....	27
4.2.1.	Borang.....	27
4.2.2.	Akses.....	28
4.3.	Desain.....	28
4.3.1.	Desain Database	28
4.3.2.	Desain Interface.....	33

4.4. Jenis & Struktur Penyajian Tabel dalam Aplikasi Borang41	
4.4.1. Tabel Induk (Master Table)	41
4.4.2. Tabel Silang/ Nested.....	45
BAB V IMPLEMENTASI.....	49
5.1. Lingkungan Implementasi.....	49
5.2. Pengembangan Sistem.....	50
5.2.1. Iterasi Pertama	51
5.2.2. Iterasi Kedua.....	61
5.2.3. Iterasi Ketiga.....	66
BAB VI HASIL DAN PEMBAHASAN	73
6.1. Hasil Pengujian	73
6.1.1. Pengujian Fungsional (<i>Functional Testing</i>).....	73
6.1.2. Pengujian Non Fungsional (<i>Non-Functional Testing</i>)	79
6.1.3. User Acceptance Test.....	82
6.2. Pembahasan.....	85
BAB VII KESIMPULAN DAN SARAN	87
7.1. KESIMPULAN	87
7.2. SARAN	88
DAFTAR PUSTAKA	89
BIODATA PENULIS	91
LAMPIRAN 1	93
LAMPIRAN 2.....	Error! Bookmark not defined.

[halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Formula pencarian APDEX.....	15
Gambar 3.1 Gambaran Iterative Development yang digunakan	19
Gambar 3.2 Laman Pembuatan Borang	20
Gambar 3.3 Laman Pengeditan Borang	20
Gambar 3.4 Laman Pengisian Borang.....	21
Gambar 4.1 Menu navigasi pada halaman borang	27
Gambar 4.2 Menu navigasi pada header	28
Gambar 4.3 Tampilan daftar Unit (level).....	34
Gambar 4.4 Tampilan daftar Grup (jurusan).....	34
Gambar 4.5 Tampilan untuk menempatkan user ke grup	35
Gambar 4.6 Tampilan untuk menambah server	35
Gambar 4.7 Tampilan untuk menambah API pada server	36
Gambar 4.8 Tampilan pembuatan borang (page).....	37
Gambar 4.9 Tampilan pembuatan section baru.....	37
Gambar 4.10 Tampilan pembuatan kolom (field) pada section	38
Gambar 4.11 Tampilan pembuatan indikator.....	39
Gambar 4.12 Tampilan isian borang	40
Gambar 4.13 Tampilan laman evaluasi.....	40
Gambar 5.1 Tampilan halaman register	53
Gambar 5.2 Command line untuk melihat route yang terdaftar	54
Gambar 5.3 Tampilan manajemen akses (memasukkan role ke user).....	58
Gambar 5.4 Tampilan laman borang dengan table.....	61
Gambar 5.5 Tampilan list isian borang (editor yang sudah mengisi).....	61
Gambar 5.6 Tampilan list isian borang (belum ada yg mengisi)	62
Gambar 5.7 Tampilan isian borang.....	62
Gambar 5.8 Tampilan isian borang.....	63

Gambar 5.9 Tampilan list revisi pada isian borang	64
Gambar 5.10 Tampilan laman evaluasi (role auditor)	66
Gambar 5.11 Tampilan periode	66
Gambar 5.12 Tampilan Tabel pada role Creator	67
Gambar 5.13 Tampilan tabel pada role Editor (formulasi tabel)	68
Gambar 5.14 Tampilan pengisian formulasi pada indikator ..	69
Gambar 5.15 Tampilan pembuatan server	70
Gambar 5.16 Tampilan pembuatan API	70
Gambar 5.17 Tampilan menghubungkan API dengan tabel borang	71
Gambar 5.18 Tampilan pada tabel isian borang	71
Gambar 6.1 Rules pada Apache JMeter	80
Gambar 6.2 Perintah eksekusi testing	80
Gambar 6.3 Persentase keberhasilan request dari testing	81
Gambar 6.4 Tabel hasil testing Apache Jmeter	Error!
Bookmark not defined.	
Gambar 6.5 Tabel hasil testing Apache Jmeter	Error!
Bookmark not defined.	
Gambar 6.6 Top 5 Errors by sampler	Error! Bookmark not defined.
Bookmark not defined.	
Gambar 6.7 APDEX (Application Performance Index)	Error!
Bookmark not defined.	
Gambar 6.8 Hasil Perhitungan SUS	84

DAFTAR TABEL

Tabel 4.1 Rincian keterangan atribut tiap entitas	30
Tabel 4.2 Relasi antar entitas	32
Tabel 4.3 Contoh tabel umpan balik	41
Tabel 4.4 Contoh tabel pencapaian mahasiswa.....	42
Tabel 4.5 Contoh tabel layanan mahasiswa	42
Tabel 4.6 Contoh tabel data dosen tetap	42
Tabel 4.7 Contoh tabel aktivitas mengajar dosen.....	43
Tabel 4.8 Contoh tabel program tugas belajar	43
Tabel 4.9 Contoh tabel kegiatan dosen tetap.....	44
Tabel 4.10 Contoh tabel pencapaian reputasi.....	44
Tabel 4.11 Contoh tabel sks program studi.....	44
Tabel 4.12 Contoh tabel profil mahasiswa & lulusan	45
Tabel 4.13 Contoh tabel profil mahasiswa & lulusan	46
Tabel 4.14 Contoh tabel sks rata-rata per semester.....	46
Tabel 4.15 Contoh tabel struktur kurikulum	46
Tabel 4.16 Contoh tabel distribusi frekuensi kumulatif.....	47
Tabel 4.17 Contoh tabel tenaga kependidikan	47
Tabel 4.18 Contoh tabel distribusi frekuensi persentase.....	48
Tabel 6.1 Daftar fitur utama aplikasi yang akan diuji.....	73
Tabel 6.2 Tabel Functional Testing.....	74
Tabel 6.3 Daftar Pertanyaan yang diajukan	83
Tabel 6.4 Pertanyaan setelah di negasi dan hasil usability test	84

DAFTAR KODE PROGRAM

Kode Program 5.1 Command line untuk inisiasi project laravel	51
Kode Program 5.2 Contoh skema tabel file migration	52
Kode Program 5.3 Command line untuk melakukan migration	52
Kode Program 5.4 Command line untuk membuat otentikasi	53
Kode Program 5.5 Command line untuk membuat controller	53
Kode Program 5.6 Contoh file route untuk controller	53
Kode Program 5.7 Contoh isi method index pada controller .	54
Kode Program 5.8 Contoh isi method create pada controller.	55
Kode Program 5.9 Contoh isi index pada view	56
Kode Program 5.10 Contoh isi show pada view.....	56
Kode Program 5.11 Contoh isi store pada view	57
Kode Program 5.12 Contoh tampilan pada client-side.....	59
Kode Program 5.13 Contoh pengambilan data menggunakan JSON	60
Kode Program 5.14 Contoh JSON isian borang.....	63
Kode Program 5.15 Contoh JSON indikator borang	65

DAFTAR DIAGRAM

Diagram 3.1 Metodologi Penelitian	17
Diagram 4.1 Use case untuk role Creator.....	25
Diagram 4.2 Use case untuk role Editor.....	26
Diagram 4.3 Use case untuk role Auditor	26
Diagram 4.4 Use case untuk role Sistem.....	27
Diagram 4.5 Relationship antar tabel	29

[halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1. Latar Belakang Masalah

Salah satu syarat satuan pendidikan dapat mengeluarkan sertifikat ialah terdapatnya akreditasi baik ditingkat institusi maupun di tingkat program studi. Akreditasi ini didapatkan melalui evaluasi terhadap mutu kinerja institusi perguruan tinggi yang dilakukan melalui asesmen terhadap borang akreditasi dan laporan evaluasi diri institusi perguruan tinggi oleh tim asesor yang terdiri atas berbagai keahlian terkait yang berpengalaman dan memahami hakikat penyelenggaraan perguruan tinggi.

Borang adalah alat untuk mengumpulkan dan mengungkapkan data dan informasi yang digunakan untuk menilai kelayakan dan mutu institusi perguruan tinggi. Evaluasi-diri yaitu upaya institusi perguruan tinggi untuk mengetahui gambaran mengenai kinerja dan keadaan dirinya melalui pengkajian dan analisis yang dilakukan oleh institusi perguruan tinggi sendiri [1].

Institut Teknologi Sepuluh Nopember (ITS) merupakan salah satu institusi terkemuka di Indonesia, institusi yang bergerak pada bidang teknik tersebut memiliki Sistem Penjaminan Mutu Internal (SPMI) yang berkaitan dengan evaluasi diri terhadap mutu kinerja institusi perguruan tinggi.

SPMI ITS dikembangkan dan berjalan sejak 2013, namun pada praktiknya terdapat beberapa permasalahan yang terjadi mulai dari kesulitan pengguna dalam pemakaiannya hingga kekurangan sistem dalam melakukan otomatisasi penilaian kualitas isian borang. Penilaian ini berupa range skor yang diberikan terhadap isian berdasarkan indikator-indikator yang sudah ditentukan sebelumnya.

Selain itu, beberapa form dalam pembuatan borang masih dilakukan secara hard-code, sehingga jika dibutuhkan perubahan maka harus dilakukan pula perubahan disisi program. Hal ini akan berdampak pada data integrity apabila dilakukan perbandingan data statistik antar borang dengan isian terdahulunya.

Tugas Akhir ini bertujuan untuk mengatasi berbagai permasalahan tersebut dengan melakukan pengembangan dari aplikasi web SPMI ITS menggunakan desain table yang dinamis pada database berstruktur statis. Hal ini dilakukan karena pengembangan document berbasis NoSQL tidak memungkinkan karena lingkungan server yang digunakan masih menggunakan SQL Server 2008 yang tidak mendukung penyimpanan berbasis objek.

Dengan adanya sistem ini diharapkan pembuatan borang akreditasi akan lebih efisien, dan melakukan perbandingan data dalam rangka pemeringkatan program studi juga lebih mudah, serta pengembangan borang untuk unit lain seperti diploma, magister ataupun tingkat fakultas dan institusi akan lebih cepat dikarenakan pembuatan borang tidak lagi memerlukan *hardcode*.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang, maka rumusan permasalahan yang menjadi fokus dan akan diselesaikan dalam Tugas Akhir ini antara lain:

1. Bagaimana pembuat borang dapat melakukan perubahan dan melakukan formulasi dasar (penjumlahan, pengurangan, pembagian, pengalian, dan rata-rata) pada borang tanpa merubah kode sumber?
2. Bagaimana pengisi borang dapat mengisi sesuai format yang ditentukan dan asesor dapat melihat skor penilaian sesuai dengan ketentuan matriks penilaian yang ditentukan oleh pembuat borang?

1.3. Batasan Masalah

Dari permasalahan yang disebutkan di atas, batasan masalah dalam tugas akhir ini adalah:

1. Studi kasus yang digunakan pada tugas akhir ini menggunakan borang dan matriks penilaian akreditasi program studi sarjana versi 08-04-2010.
2. Aplikasi berikut menggunakan database relational agar kompatibel dengan lingkungan implementasi.
3. Integrasi API dalam pengambilan data menggunakan data dummy karena belum tersedianya API resmi.

1.4. Tujuan Penelitian Tugas Akhir

Berdasarkan hasil perumusan masalah dan batasan masalah yang telah disebutkan sebelumnya, maka tujuan yang ingin dicapai dari tugas akhir ini adalah agar hasil rancang bangun aplikasi ini dapat digunakan untuk menilai kelayakan dan mutu institusi perguruan tinggi dan untuk mengetahui gambaran mengenai kinerja dan keadaan dirinya secara efisien dan terukur.

Tujuan dari aplikasi ini sendiri adalah untuk otomasi penilaian kualitas isian borang yang dapat melakukan perbandingan penilaian borang baik dari antar jurusan maupun perkembangan dari tahun ke tahun, dan integrasi data isian borang dalam format yang mudah diakses, dicetak, dan di review tanpa

mengorbankan kenyamanan user dalam melakukan penginputan data.

1.5. Manfaat Penelitian Tugas Akhir

Manfaat yang diharapkan dapat diperoleh dari tugas akhir ini adalah:

1. Aplikasi ini dapat digunakan untuk keperluan evaluasi masing-masing jurusan dalam rangka akreditasi Program Studi Sarjana.
2. Aplikasi ini dapat dikembangkan bukan hanya untuk akreditasi Program Studi Sarjana, namun juga untuk Program Studi Diploma; Magister; Doktor; Fakultas; Unit Pengelola; bahkan Perguruan Tinggi.
3. Hasil penilaian dalam aplikasi ini bisa digunakan untuk menentukan peringkat Prodi berkinerja terbaik atau Penilaian Program Studi Berkinerja Terbaik (PPSBT).

1.6. Relevansi Tugas Akhir

Tugas akhir ini berkaitan dengan mata kuliah Desain Basis Data; Manajemen dan Administrasi Basis Data; Analisa dan Desain Perangkat Lunak, Konstruksi Pengembangan Perangkat Lunak; Pemrograman Berorientasi Objek; Pemrograman Berbasis Web; Interaksi Manusia dan Komputer; Integrasi Aplikasi Korporasi dan Sistem Pendukung Keputusan.

Selain itu, tugas akhir ini berkaitan dengan Lab Akuisisi Data dan Diseminasi Informasi; dan Lab Infrastruktur & Keamanan Teknologi Informasi. Dimana tugas akhir ini berkaitan dengan perancangan skema yang dinamis dalam database yang berstruktur statis (SQL - Structured Query Language); dan rancang bangun aplikasi yang berkaitan dengan pemrograman berbasis objek.

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir ini.

2.1. Penelitian Sebelumnya

Penelitian yang dijadikan acuan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

2.1.1. Sistem Penjaminan Mutu Internal (SPMI-ITS) - 2016

SPMI-ITS dikembangkan oleh Lembaga Penjaminan Mutu, Pengelolaan Dan Perlindungan Kekayaan Intelektual Institut Teknologi Sepuluh Nopember (ITS). SPMI-ITS atau kata lain disebut sebagai Quality Assurance-ITS dilakukan dan didokumentasikan sesuai dengan pedoman Sistem Penjaminan Mutu Perguruan Tinggi (SPM-PT). Dengan pelaksanaan SPMI, ITS diharapkan mampu mencapai Visi dan Misi yang telah ditetapkan pada Statuta ITS, yang dituankan dalam Peraturan Pemerintah (PP) No. 54 tahun 2015, serta pada Rencana Strategis ITS Tahun 2014 – 2018.

Penilaian Program Studi Berkinerja Terbaik (PPSBT) untuk tahun 2016 didasarkan pada kriteria yang terdapat pada BAN PT dengan instrumen tahun 2010, Standar Internal ITS. Standar Internal ITS terdiri dari: Standar Nasional Dikti yang tertuang pada Permenristekdikti No. 44 Tahun 2015, dan standar yang diturunkan dari Visi, Misi ITS dalam Statuta – PP No. 54 Tahun 2015.

Standar pada SPMI ITS adalah gabungan dari kriteria Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) tahun 2008,

dengan ditambahkan standar dari SN Dikti yang diamanahkan dalam Peraturan Menteri Riset, Teknologi dan Pendidikan Tinggi (Ristekdikti) No 44 tahun 2015, yang belum terakomodasi di dalam kriteria BAN PT dan standar turunan yang sudah menjadi budaya dalam kehidupan akademik di ITS [1].

2.2. Dasar Teori

2.2.1. Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT)

BAN-PT dibentuk pada tahun 1994 oleh Menteri Pendidikan dan Kebudayaan. BAN-PT memiliki tugas dan wewenang berdasarkan Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi Nomor 32 tahun 2016 adalah [2]:

- a) mengembangkan sistem akreditasi Program Studi dan Perguruan Tinggi selaras dengan kebijakan pengembangan pendidikan tinggi;
- b) menyusun dan menetapkan instrumen akreditasi Perguruan Tinggi berdasarkan Standar Pendidikan Tinggi;
- c) melakukan akreditasi Perguruan Tinggi;
- d) menerbitkan, mengubah, atau mencabut keputusan tentang status akreditasi dan peringkat terakreditasi Perguruan Tinggi;
- e) memeriksa, melakukan uji kebenaran, dan memutuskan keberatan yang diajukan atas status akreditasi dan/atau peringkat terakreditasi Perguruan Tinggi;
- f) membangun dan mengembangkan jejaring dengan pemangku kepentingan baik di tingkat nasional maupun internasional;
- g) melakukan penilaian kelayakan pendirian LAM sebagai dasar rekomendasi pengakuan Menteri kepada LAM;
- h) mengevaluasi kinerja LAM secara berkala yang hasilnya disampaikan kepada Menteri;

- i) menyusun instrumen evaluasi pendirian Perguruan Tinggi berdasarkan Standar Nasional Pendidikan Tinggi bersama dengan Direktur Jenderal Kelembagaan Ilmu Pengetahuan, Teknologi, dan Pendidikan Tinggi;
- j) memberikan rekomendasi pemenuhan persyaratan minimum akreditasi untuk pendirian Perguruan Tinggi kepada Direktorat Jenderal Kelembagaan Ilmu Pengetahuan, Teknologi, dan Pendidikan Tinggi; dan
- k) menyampaikan laporan hasil akreditasi dilengkapi dengan rekomendasi secara berkala kepada Menteri.

2.2.2. Role-based Access Control

Ialah metode dalam mengendalikan akses ke fungsi berdasarkan pada "peran" (role) pengguna, bukan berdasarkan identitasnya. Sebagai contoh, sebuah sistem pada blog mendefinisikan peran "Author" dan peran "Editor". Sebuah "Author" mungkin memiliki izin untuk membuat cerita baru, tetapi tidak untuk mempublikasikannya. Editor akan memiliki izin untuk meninjau dan memodifikasi, dan mempublikasikan cerita yang ada.

Setiap pengguna tertentu mungkin memiliki lebih dari satu peran secara permanen, maupun sementara yang otorisasinya diberikan dalam sebuah sesi. Tapi intinya, hak akses dan kemampuan yang diberikan tidak pernah langsung ke pengguna, tetapi selalu ditugaskan untuk melalui peran. Pengguna mendapatkan izin yang secara tidak langsung dengan mengambil peran yang diberikan [3].

2.2.3. Laravel Framework

Laravel adalah *web application framework* berbasis PHP yang *open source*, menggunakan konsep *model-view-controller* (MVC). Laravel memudahkan developer dalam mengembangkan aplikasi web dengan beberapa kelebihan sebagai berikut [4]:

- **Authentication.** Laravel mendukung autentikasi *out-of-the-box*. Fitur autentikasi (login, register, lupa password dan logout) berikut struktur database sudah di generate oleh Laravel dengan hanya sebuah *command*.
- **Testing.** Laravel sangat mendukung testing dalam pengembangan aplikasi. Secara default, Laravel menyediakan *phpunit.xml* yang mempersingkat waktu developer untuk menyiapkan *environment* untuk testing. Laravel juga menyediakan generator untuk membuat template testing.
- **Routing.** Routing adalah teknik untuk membangun URL di aplikasi. Routing di Laravel sederhana, fleksibel namun tetap powerful. Laravel mendukung pembuatan routing manual maupun Restful routing dengan satu baris syntax.
- **Database Migration.** Struktur database (bahkan sample data) dapat masuk ke codebase. Hingga dengan satu perintah dan rekan satu team akan mendapatkan struktur database dan sample datanya. Tidak perlu dump, drop dan import sql manual.
- **Eloquent ORM.** Eloquent adalah implementasi active record yang akan memudahkan kita berinteraksi dengan database, terutama database relasional. Menggunakan Eloquent, tidak perlu membuat join manual. Setiap table di database akan memiliki model masing-masing. Untuk setiap operasi CRUD ke database, cukup menggunakan model tersebut.
- **Homestead.** Homestead ibaratnya sebuah image pada virtual box. Dengan homestead kita dapat

mengkonfigurasi environment untuk aplikasi sendiri, sehingga aplikasi dapat berjalan di server manapun.

- **Query Builder.** Terkadang Eloquent saja tidak cukup untuk query yang sangat kompleks. Tetapi, kita juga tidak ingin menggunakan raw sql. Nah, query builder solusinya. Dengan fitur ini, kita dapat membuat query yang cukup kompleks tapi tetap readable oleh programmer lain.
- **HTTP Middleware.** Fitur ini akan memudahkan kita untuk menambahkan berbagai filter ke request yang masuk ke aplikasi maupun memodifikasi response yang diberikan ke user.
- **Artisan Console.** Dengan artisan, kita dapat berinteraksi dengan Laravel menggunakan command line. Ini sangat berguna saat development untuk mengecek berbagai logic bisnis dari aplikasi tanpa harus membuka browser.
- **Caching.** Laravel menyediakan API yang sama untuk berbagai tipe backend caching. Beberapa backend caching yang didukung out-of-the-box oleh Laravel diantaranya Redis dan Memcached.
- **Blade Template.** Blade adalah salah satu templating engine yang sederhana namun tetap powerful di PHP. Tentunya, Laravel tidak melarang kita untuk menggunakan PHP biasa untuk membuat view. Tapi, menggunakan Blade struktur dari view kita akan lebih rapi.
- **Filesystem / Cloud Storage.** Seperti caching, Laravel menyediakan API yang sama untuk berbagai provider cloud storage. Ini akan sangat memudahkan developer karena mereka tidak perlu memikirkan API call yang berbeda dari tiap provider tersebut. Beberapa provider yang didukung secara native oleh Laravel diantaranya local, Rackspase Cloud Storage dan Amazon S3.

2.2.4. SQL Server

Microsoft SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur client server. Fitur pada Microsoft SQL Server yaitu mempunyai kemampuan untuk membuat basis data mirroring dan clustering. Berikut ini beberapa kelebihan SQL Server sebagai database antara lain [5]:

- Dengan kemampuannya untuk mengolah data yang besar maka DBMS ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar sekalipun.
- DBMS jenis ini memiliki kelebihan memmanage user data serta masing-masing user dapat diatur hak aksesnya terhadap pengaksesan data base oleh DBA
- Mempunyai tingkat keamanan data yang sangat baik
- Dapat melakukan back up, recovery dan rollback data dengan mudah
- Mempunyai kelebihan untuk membuat data base mirroring dan clustering

2.2.5. Iterative and Incremental Development

Ide dasar dari metode pengembangan ini ialah untuk mengembangkan sebuah sistem melalui siklus berulang (iterative) dan dalam porsi yang lebih kecil pada suatu waktu (incremental), memungkinkan pengembang perangkat lunak untuk mengambil keuntungan dari apa yang telah dipelajari selama pengembangan bagian-bagian awal versi sistem.

Adapun fase dalam proses pengembangan dengan Iterative development yakni [6]:

1. Inception

Inception adalah tahap permulaan untuk mengidentifikasi pengembangan sistem. Pada tahap ini dilakukan beberapa aktivitas diantaranya adalah menentukan ruang lingkup dari penelitian, proses penggalan dan analisis kebutuhan user, proses perancangan awal perangkat lunak, dan permodelan UML yang melingkupi berbagai diagram proses seperti

domain model, use case diagram, activity diagram & sequence diagram.

2. Elaboration

Fase elaboration yang adalah fase perluasan atau perencanaan. Pada tahap ini dilakukan desain secara lengkap berdasarkan hasil analisis di tahap inception. Aktivitas terkait yang ada di tahap ini antara lain adalah pembuatan desain arsitektur sistem (architecture pattern), desain format data/ struktur data, desain database sistem, tampilan sistem (user interface), serta menentukan design pattern yang digunakan.

3. Construction

Dalam tahapan konstruksi, dilakukan proses implementasi hasil desain dan proses pengujian hasil implementasi. Aktivitas yang ada dalam tahap ini diantaranya adalah pengujian hasil analisis dan desain, penentuan coding program yang digunakan, pembuatan program, dan optimasi program.

4. Transition

Transition (transisi), merupakan tahap untuk menyerahkan sistem aplikasi ke konsumen (roll-out), yang umumnya mencakup pelaksanaan pelatihan kepada pengguna dan testing beta aplikasi terhadap ekspektasi pengguna.

2.2.6. Dynamic Database Schema

Schema-less design database pada umumnya diselesaikan dengan menggunakan Document-Oriented Database seperti MongoDB dengan data model BSON (Binary JSON) yang diklasifikasikan sebagai basis data NoSQL.

Sedangkan SQL pada umumnya menggunakan skema database statis dimana tabel dan kolom nya sudah ditentukan sebelumnya (pre-defined).

2.2.7. JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.
- Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [7].

2.2.8. Apache JMeter

Apache JMeter adalah perangkat lunak *open source*, murni 100% aplikasi *Java* yang dirancang untuk melakukan uji fungsional dan mengukur kinerja suatu server perangkat lunak (seperti aplikasi web). JMeter dapat mengukur kinerja HTTP (aplikasi web), FTP, JDBC, bahkan EJB, SOAP atau CORBA.

Dengan JMeter kita dapat mensimulasikan beban kerja yang sangat berat pada sistem client/server yang kita kembangkan, termasuk server dan jaringannya. Walaupun dibuat dengan 100% murni *Java*, JMeter dapat menguji aplikasi web yang ditulis dengan semua bahasa pemrograman web yang Anda sukai, seperti JSP/Servlet, PHP, ASP, Cold Fusion, CGI, Ruby, dan sebagainya.

Konsep yang diusung dalam JMeter adalah penyusunan sebuah *TestPlan* yang berisi satu atau lebih *ThreadGroup*. Dalam sebuah *ThreadGroup* berisi beberapa *thread*. Sebuah *thread* dapat di analogikan sebagai seorang virtual pemakai yang sedang mengakses aplikasi web/server Anda. Sebuah *timer* (waktu tunda) dapat ditambahkan ke *ThreadGroup* sebagai simulasi waktu pengaksesan yang dilakukan di antara *thread*. Dengan adanya *timer* ini, kita bisa mensimulasikan waktu pengaksesan dari pemakai (diwakili dengan *thread*) secara teratur (konstan), Gaussian random atau uniform. Biasanya kita gunakan *timer* konstan untuk membuat pengujian yang berulang-ulang, dan Gaussian random untuk mensimulasikan aktifitas pemakai pada kenyataannya. Elemen lain dalam JMeter adalah *controller*. JMeter menggunakan dua tipe *controller*, yaitu *testing* dan *logic*. Anda dapat menggunakan sebuah *controller testing* untuk menguji sistem kita dengan berbagai protokol, tapi tidak mencatat hasil pengujian ini. Sedangkan untuk *controller logic* kita dapat menguji aliran pengujian yang dilakukan. Sebuah *controller*

dapat berisi beberapa elemen config yang membantu kita untuk mengkonfigurasi controller [8].

2.2.9. Application Performance Index (APDEX)

Apdex adalah ukuran numerik kepuasan pengguna dengan kinerja aplikasi perusahaan. Apdex mengubah banyak pengukuran menjadi satu nomor pada skala seragam 0 sampai 1 (0 = tidak ada pengguna yang puas, 1 = semua pengguna puas). Metrik ini dapat diterapkan pada setiap sumber pengukuran kinerja pengguna akhir. Jika Anda memiliki alat ukur yang mengumpulkan data waktu yang serupa dengan yang dapat diperoleh pengguna akhir yang termotivasi dengan stopwatch, Anda dapat menggunakan metrik ini. Apdex mengisi kesenjangan antara data waktu dan wawasan dengan menentukan cara yang seragam untuk mengukur dan melaporkan pengalaman pengguna.

Indeks menerjemahkan banyak waktu tanggapan individual, yang diukur pada tingkat tugas pengguna, menjadi satu nomor tunggal. Tugas adalah interaksi individu dengan sistem, dalam proses yang lebih besar. Waktu respons tugas didefinisikan sebagai waktu yang telah berlalu antara saat pengguna melakukan sesuatu (klik mouse, masuk atau balik, dll.) Dan saat sistem (klien, jaringan, server) merespons sehingga dia dapat melanjutkan prosesnya. Inilah saat dimana manusia menunggu sistem. Masa tunggu individu inilah yang menentukan "responsivitas" aplikasi kepada pengguna.

Formula Apdex adalah jumlah sampel yang puas ditambah setengah dari sampel yang dapat ditoleransi dan tidak ada sampel frustrasi, dibagi dengan semua sampel [9]:

$$Apdex_t = \frac{SatisfiedCount + \frac{ToleratingCount}{2}}{TotalSamples}$$

Gambar 2.1 Formula pencarian APDEX

2.2.10. System Usability Scale

Usability adalah suatu ukuran, di mana pengguna dapat mengakses fungsionalitas dari sebuah sistem dengan efektif, efisien dan memuaskan dalam mencapai tujuan tertentu.

Salah satu metode yang paling sering digunakan dalam usability testing menggunakan System Usability Scale (SUS). System Usability Scale merupakan alat yang dibuat oleh John Brooke pada tahun 1986 yang digunakan untuk mengevaluasi berbagai macam produk dan layanan, termasuk perangkat keras, perangkat lunak, perangkat mobile, website dan aplikasi. Pada pengukuran System Usability Scale terdiri dari 10 item kuesioner dengan masing – masing 5 respon setiap item yang diukur. Sangat tidak setuju sampai Sangat Setuju.

Sepuluh item tersebut adalah sebagai berikut:

1. Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin
2. Saya menemukan sistem sangat rumit dan tidak perlu
3. Saya berpikir sistem ini mudah digunakan
4. Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini
5. Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik
6. Saya berpikir terlalu banyak inkonsistensi dari sistem ini
7. Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat

8. Saya berpikir bahwa sistem ini sangat sulit untuk digunakan
9. Saya merasa sangat percaya diri dalam menggunakan sistem ini
10. Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini [10]

Dalam mengolah hasil responden yang diterima, terdapat beberapa aturan atau tahapan

1. Untuk soal ganjil: Kurangi 1 dari setiap jawaban Responden
2. Untuk soal genap: kurangi lima untuk setiap jawaban responden
3. Skala ini diubah menjadi 0 – 4 (empat menjadi respon yang paling positif)
4. Seluruh tanggapan dari pengguna dikonversi dan dijumlahkan dan dikalikan dengan 2,5. Ini mengubah rentang nilai yang mungkin dari 0 sampai 100.

Penilaian SUS ini telah digunakan selama bertahun-tahun dalam berbagai evaluasi kegunaan. Pada penelitiannya Jeff Sauro telah melakukannya ke lebih dari 5000 pengguna/responden di lebih dari 500 jenis penelitian. Ini menunjukkan bahwa SUS menjadi alat ukur yang handal dan valid untuk evaluasi kegunaan [11]

BAB III METODOLOGI PENELITIAN

Pada bab metode penelitian akan dijelaskan mengenai tahapan – tahapan apa saja yang dilakukan dalam pengerjaan tugas akhir ini beserta deskripsi dan penjelasan tiap tahapan tersebut. Lalu disertakan jadwal pengerjaan tiap tahapan.

3.1. Tahapan Pelaksanaan Tugas Akhir

Pada sub bab ini akan menjelaskan mengenai metodologi dalam pelaksanaan tugas akhir. Metodologi ini dapat dilihat pada Gambar berikut:

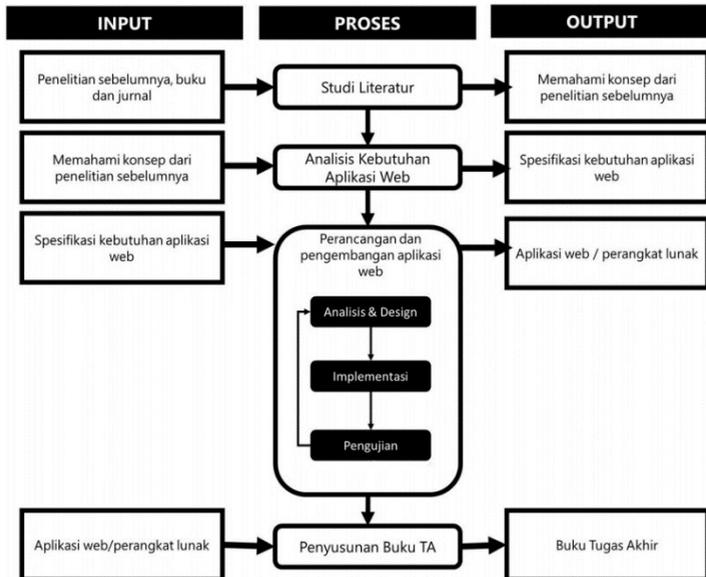


Diagram 3.1 Metodologi Penelitian

3.2. Uraian Metodologi

Berdasarkan pada diagram alur metodologi yang telah dikemukakan pada sub bab sebelumnya, di bawah ini adalah penjelasan pada setiap proses pengerjaan tugas akhir sesuai dengan metodologi yang telah dibuat:

3.2.1. Studi Literatur

Pada tahap ini dilakukan pengumpulan literatur yang mendukung dalam menyelesaikan tugas akhir ini. Literatur disini adalah penjelasan konsep-konsep atau penelitian sebelumnya yang pernah dilakukan dan didokumentasikan dalam buku, jurnal, maupun website. Output atau keluaran proses ini adalah pemahaman mengenai konsep dan knowledge gap pada penelitian sebelumnya.

3.2.2. Analisa Kebutuhan Aplikasi Borang

Pada tahap ini dilakukan analisa dan desain aplikasi borang. Setelah mengetahui konsep dan penelitian sebelumnya maka dapat melakukan analisa aplikasi borang yang akan dibuat. Pada analisa yang harus dilakukan adalah merancang siapa saja user yang akan menggunakan dan kebutuhan masing-masing user pada pengerjaan iterasi pertama.

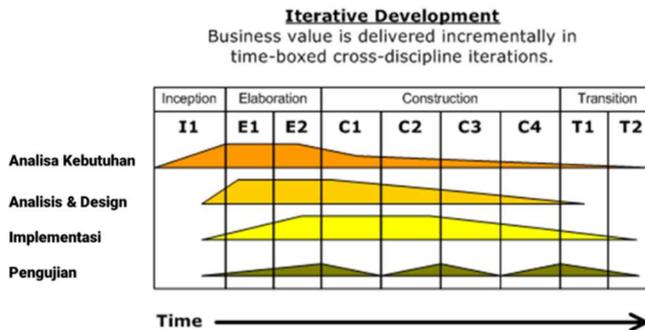
Kemudian untuk template borang, dilakukan pemetaan bagian-bagian berdasarkan matriks penilaian yang ditentukan di awal. Sedangkan untuk formatting, dilakukan listing format dan rule apa saja yang diperlukan dalam penilaian isian borang. Hal ini dilakukan pada pengerjaan iterasi kedua.

3.2.3. Perancangan dan Pengembangan Aplikasi Borang

Pada tahap ini dilakukan perancangan dan pengembangan aplikasi yang merupakan implementasi dari hasil melakukan analisa kebutuhan aplikasi borang. Tahapan yang dilakukan dalam proses perancangan serta pengembangan web repositori

menggunakan metode pengembangan *Iterative and Incremental*.

Beberapa proses dalam metode pengembangan iterative dapat dilakukan bersamaan tanpa menunggu proses sebelumnya selesai terlebih dahulu. Adapun fase dalam proses pengembangan dengan Iterative development direpresentasikan melalui kode di setiap fasenya:



Gambar 3.1 Gambaran Iterative Development yang digunakan [12]

1. Analisis & Desain (Fase Elaboration → E1, E2)

Proses ini berada paling banyak pada metode elaboration dimana seluruh desain dibuat dari hasil perencanaan. Berdasarkan hasil analisis kebutuhan pada tahap sebelumnya akan digunakan untuk membuat gambaran desain dari fitur dan fungsionalitas terhadap aplikasi. Pada tahap ini akan dibuat beberapa desain secara bertahap antara lain:

 - a) Desain database (pada iterasi pertama)
 - b) Pemetaan navigasi /sitemap (pada iterasi kedua)
 - c) Desain aplikasi (pada iterasi ketiga)

Pengembangan diatas merupakan fokus yang dilakukan pada iterasi tertentu, tapi tidak terbatas hanya di satu iterasi,

melainkan dapat berkembang tergantung dari hasil pengerjaan di iterasi sebelumnya.

2. Implementasi (Fase Construction)

Proses ini terjadi dan mendominasi fase construction. Setelah melalui tahap desain, berikutnya dilakukan pengkodean terhadap web yang menggunakan bahasa pemrograman PHP dengan framework Laravel. Berikut beberapa hasil prototipe awal implementasi aplikasi:

Buat Borang Baru

Judul Borang

Deskripsi Borang

List Borang

Buat Testing
Akreditasi Program Studi Sarjana
Standar 4 - Sumber Daya Manusia

Gambar 3.2 Laman Pembuatan Borang

Cscpro Labs List Borang Lihat Borang Edit Borang

Homepage dan E-mail PS Short answer

(*) Lampirkan fotokopi SK terakhir Heading 5

Bagi PS yang dibina oleh Departemen Pendidikan Nasional, sebutkan nama dosen tetap institusi yang terdaftar sebagai dosen tetap PS berdasarkan SK 034/DIKTI/Kep/2002, dalam tabel di bawah ini. Heading 4

No.	Nama Dosen Tetap	NIDN**	Tgl. Lahir	Jabatan Akademik	Gelar Akademik	Pendidikan S1, S2, S3 dan Asal PT	Bidang Keahlian untuk Setiap Jenjang Pendidikan
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

** NIDN : Nomor Induk Dosen Nasional

IDENTITAS PENGISI BORANG PROGRAM STUDI

Add New Field

Heading 1 1

Gambar 3.3 Laman Pengeditan Borang

Gambar 3.4 Laman Pengisian Borang

Pada tahap ini akan dibuat beberapa implementasi secara bertahap antara lain:

- a) Implementasi pembuatan, perubahan, dan pengisian borang (pada iterasi pertama)
- b) Implementasi penilaian kualitas borang (pada iterasi kedua)
- c) Implementasi statistik dan pemeringkatan antar borang (pada iterasi ketiga)

Pengembangan diatas merupakan fokus yang dilakukan pada iterasi tertentu, tapi tidak terbatas hanya di satu iterasi, melainkan dapat berkembang tergantung dari hasil pengerjaan di iterasi sebelumnya.

3. Pengujian (Construction-Transition)

Proses ini terjadi seiring proses implementasi yang berada pada fase construction hingga transition. Pada tahap ini akan dilakukan pengujian aplikasi untuk memastikan aplikasi dapat berjalan sesuai dengan yang diharapkan oleh pengguna dan mencatat semua bug dan error yang ada, pengujian dilakukan di setiap akhir iterasi. Berikut adalah daftar pengujian yang dapat di lakukan pada system:

- a) Uji fungsionalitas aplikasi web
- b) User Acceptance Test
- c) Functional Testing, ialah pengujian terhadap fungsionalitas dari perangkat lunak yang telah dibangun untuk memastikan fungsionalitas perangkat lunak yang diuji melalui percobaan menggunakan skenario yang telah disiapkan. Setiap skenario penggunaan perangkat lunak akan menguji setiap fungsionalitas berbeda dari perangkat lunak. Sisi fungsional yang akan diuji salah satunya mencoba beberapa akurasi penilaian kualitas isian borang.
- d) Non-Functional testing, ialah pengujian terhadap sisi non-fungsional dari perangkat lunak. Sisi non fungsional yang akan diuji adalah uji coba bagaimana kecepatan akses aplikasi web.

4. Deployment (Transition)

Proses ini merupakan proses yang terjadi pada fase transisi, dimana aplikasi web yang telah dibuat ditempatkan pada lingkungan organisasi (dalam kasus ini program studi S1) yang akan menggunakan aplikasi ini

3.2.4. Penyusunan Buku TA

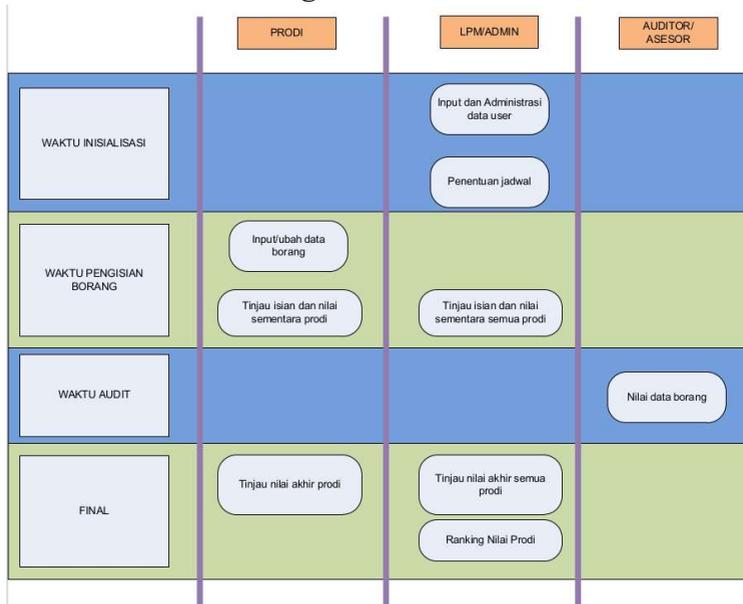
Penulisan Buku Tugas Akhir adalah tahap yang menghasilkan buku Tugas Akhir. Penulisan buku ini dilakukan bersamaan dengan tahapan penelitian yang lainnya. Harapannya, luaran berupa buku tugas akhir dapat dijadikan sebagai referensi untuk penelitian selanjutnya.

BAB IV PERANCANGAN

Pada bab ini, akan dijelaskan perancangan yang akan dilakukan untuk membangun aplikasi pada tugas akhir ini. Aplikasi borang akreditasi ini dirancang dengan beberapa langkah yakni analisis kebutuhan, desain sistem, desain database, dan desain aplikasi.

4.1. Analisis Kebutuhan

4.1.1. Kebutuhan Fungsional



Gambar 4.1 Alur kerja SPMI berdasarkan peran setiap elemen

Berdasarkan buku panduan Sistem Penjaminan Mutu Internal ITS terdapat tiga role pada SPMI yaitu LPM (Lembaga Penjaminan Mutu) / Admin yang melakukan input dan administrasi data user serta menentukan jadwal pengisian borang dan waktu audit saat inisialisasi. Lalu role Prodi (Program Studi) yang menginput data borang dan meninjau

isian sementara pada waktu pengisian borang. Yang ketiga yaitu role Auditor / Asesor yang akan menilai data borang pada saat waktu audit. Sistem eksisting berikut dimulai dari role LPM/Admin melakukan input dan administrasi data user hingga berakhir pada role LPM/Admin melakukan Rangkaing nilai Prodi.

Pada fitur saat ini terdapat beberapa kekurangan seperti struktur borang yang masih statis, yang berarti dibutuhkan perubahan pada sisi kode program untuk mengubah borang akreditasi. Dan juga tidak adanya otomasi pengambilan data untuk memudahkan editor dalam melakukan pengisian borang.

Kebutuhan fungsional aplikasi juga diambil dari fitur eksisting aplikasi SPMI dengan mengganti beberapa nama role yang sebelumnya bernama Prodi menjadi Editor, Admin menjadi Creator dan Admin dengan perbedaan Admin dapat membuat query pada database sedangkan Creator hanya sebagai pembuat borang. Terakhir Asesor menjadi Auditor sehingga keseluruhan kebutuhan perangkat lunak ialah sebagai berikut:

1. Aplikasi dapat menangani proses login user dengan berbagai role, diantaranya sebagai: admin, creator, editor, dan auditor.
2. Role creator dapat membuat borang akreditasi dan indikator nya.
3. Role admin dapat mengelola user / menambah hak akses, yaitu memasukkan dan mengeluarkan user dari grup.
4. Role editor dapat mengisi borang akreditasi.
5. Role auditor dapat mengevaluasi borang akreditasi.
6. Sistem dapat melakukan pengisian borang secara otomatis berdasarkan API yang ada.
7. Sistem dapat melakukan otomasi penilaian berdasarkan indikator yang sudah ditentukan.

Role yang muncul dari kebutuhan analisis tersebut misalnya ialah petugas penjaminan mutu ITS sebagai Creator,

departemen menjadi role Editor, dan dosen-dosen yang terpilih sebagai penilai akan menjadi Auditor. Admin sendiri pada sistem ini akan dikelola oleh admin Departemen Pengelola Teknologi Sistem Informasi (DPTSI) yang mengelola database dari ITS. Penjelasan lebih lanjut untuk setiap role akan dibahas pada setiap Use Case.

4.1.2. Use Case Diagram

4.1.2.1. Creator

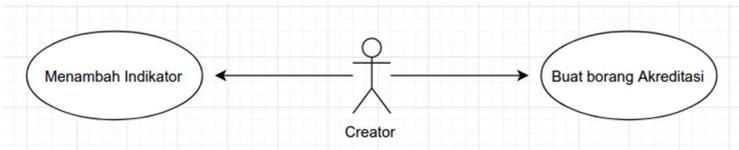


Diagram 4.1 Use case untuk role Creator

Creator dapat melakukan pengelolaan terhadap hak akses user seperti memasukkan user sebagai salah satu role pada periode tertentu dan juga mengubah periode pengisian borang untuk role tertentu. Creator juga dapat membuat borang, section, kolom, tabel, dan juga indikator pada borang akreditasi.

4.1.2.2. Admin

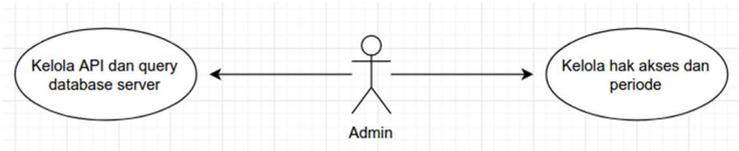


Diagram 4.2 Use case untuk role Admin

Admin berhak untuk menambah query API untuk menghubungkan tabel dengan database lain seperti database akademik maupun kemahasiswaan.

4.1.2.3. Editor

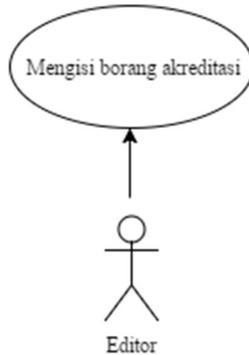


Diagram 4.3 Use case untuk role *Editor*

Role editor dapat mengisi borang yang telah dibuat oleh creator.

4.1.2.4. Auditor

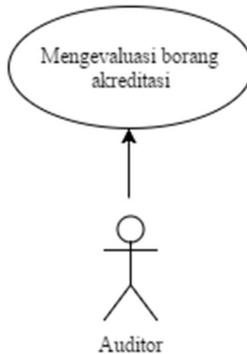


Diagram 4.4 Use case untuk role *Auditor*

Role auditor yang melakukan evaluasi terhadap isian dari editor berdasarkan indikator yang dibuat oleh creator.

4.1.2.5. Sistem

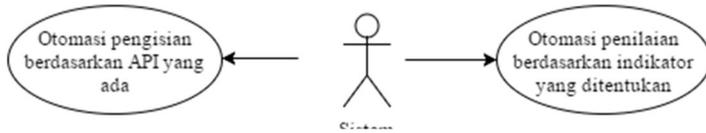


Diagram 4.5 Use case untuk role Sistem

Kemudian sistem yang akan berperan melakukan otomasi pengisian bagi role editor untuk table-table yang sudah dihubungkan dengan API oleh creator pada tahap sebelumnya.

4.2. Pemetaan navigasi / sitemap

4.2.1. Borang



Gambar 4.2 Menu navigasi pada halaman borang

Terdapat 4 menu pada halaman borang, yaitu:

- View : untuk melihat struktur borang
- Isian : untuk melihat isian borang
- Nilai : untuk melihat nilai
- Edit : untuk mengubah struktur borang/indicator (khusus *role Creator*)

4.2.2. Akses



Gambar 4.3 Menu navigasi pada header

Menu akses hanya dapat diakses oleh role creator. Terdapat 4 menu pada menu akses, yaitu:

- Periode/Season : untuk mengubah masa tenggat bagi editor / auditor untuk mengisi / mengevaluasi borang.
- Unit : untuk menambah dan mengubah unit (level).
- Grup : untuk menambah dan mengubah grup.
- Server : untuk menambah dan mengubah server maupun API.

4.3. Desain

4.3.1. Desain Database

Aplikasi web yang akan digunakan oleh pihak distributor menggunakan database SQL Server yang terdiri dari beberapa tabel untuk menyimpan data. Pada bagian ini akan dibahas mengenai tabel-tabel yang akan digunakan untuk aplikasi borang akreditasi.

Skema database dari aplikasi borang akreditasi dapat dilihat pada gambar 4.6. Skema database ini merupakan visualisasi dari phpMyAdmin designer.

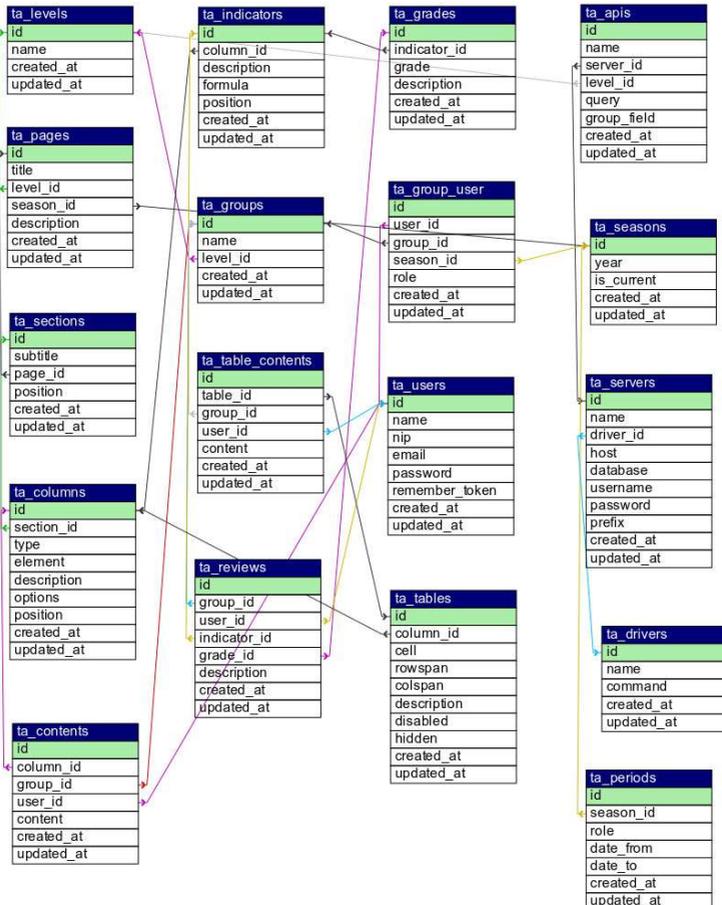


Diagram 4.2 Relationship antar tabel

Adapun rincian setiap entitas yang ada pada diagram tabel diatas dapat dilihat dalam tabel 4.1 berikut.

Tabel 4.1 Rincian keterangan atribut tiap entitas

No.	Entitas	Atribut	Keterangan
1	User	- ID - Name - NIP - Email - Password	Tabel ini digunakan untuk menyimpan informasi <i>user</i> aplikasi
2	Season	- ID - Year - Current	Tabel ini digunakan untuk menyimpan informasi season (periode)
3	Period	- ID - Season_ID - Role - Date_form - Date_to	Tabel ini digunakan untuk menghubungkan season dengan periode dibuka nya masing-masing role.
4	Level	- ID - Name	Tabel ini digunakan untuk menyimpan informasi level yang ada. (s1, s2, s3, upt)
5	Group	- ID - Name - Level_ID	Tabel ini digunakan untuk menyimpan informasi grup.
6	GroupUser	- ID - User_ID - Group_ID - Season_ID - Role	Tabel ini digunakan sebagai penghubung tabel user dan tabel grup.
7	Page	- ID - Title - Level_ID - Season_ID - Description	Tabel ini digunakan untuk menyimpan informasi laman borang.
8	Section	- ID - Subtitle - Page_ID - Position	Tabel ini digunakan untuk menyimpan informasi bagian pada laman borang.
9	Column	- ID - Section_ID - Type	Tabel ini digunakan untuk menyimpan

		<ul style="list-style-type: none"> - Element - Description - Option - Position 	informasi kolom pada tiap bagian borang.
10	Indicator	<ul style="list-style-type: none"> - ID - Column_ID - Description - Position 	Tabel ini digunakan untuk menyimpan informasi mengenai indikator keberhasilan dari tiap kolom borang.
11	Grade	<ul style="list-style-type: none"> - ID - Indicator_ID - Grade - Description 	Tabel ini digunakan untuk menyimpan informasi list nilai dan kriteria dari tiap indikator.
12	Table	<ul style="list-style-type: none"> - ID - Column_ID - Cell - Rowspan - Colspan - Description - Disabled 	Tabel ini digunakan untuk menyimpan informasi tabel dari tiap kolom borang.
13	Content	<ul style="list-style-type: none"> - ID - Column_ID - Group_ID - User_ID - Content 	Tabel ini digunakan untuk menyimpan isian borang.
14	TableContent	<ul style="list-style-type: none"> - ID - Table_ID - Group_ID - User_ID - Content 	Tabel ini digunakan untuk menyimpan informasi isian tabel borang.
15	Review	<ul style="list-style-type: none"> - ID - Group_ID - User_ID - Indicator_ID - Grade_ID - Description 	Tabel ini digunakan untuk menyimpan informasi penilaian terhadap isian borang.
16	Driver	<ul style="list-style-type: none"> - ID - Name - Command 	Tabel ini digunakan untuk menyimpan jenis driver SQL yang bisa

			digunakan untuk koneksi ke server.
17	Server	<ul style="list-style-type: none"> - ID - Name - Driver_ID - Host - Database - Username - Password - Prefix 	Tabel ini digunakan untuk menyimpan list server beserta koneksi nya ke server.
18	API	<ul style="list-style-type: none"> - ID - Name - Server_ID - Query - Group_field 	Tabel ini digunakan untuk menyimpan list API beserta query nya yang terhubung dengan server.

Adapun relasi antar entitas pada skema database dijabarkan pada tabel 4.2 berikut ini:

Tabel 4.2 Relasi antar entitas

Relationship Type	Constraint	Keterangan
(User, Group)	(N, N)	Setiap user dapat memiliki banyak group, dan satu grup dapat memiliki banyak user.
(Page, Level)	(N, 1)	Setiap page hanya memiliki satu level, dan satu level dapat dimiliki oleh banyak Page.
(Page, Section)	(1, N)	Setiap page dapat memiliki banyak section, dan satu section hanya dimiliki oleh satu Page.
(Section, Column)	(1, N)	Setiap page dapat memiliki banyak column, dan satu column hanya dimiliki oleh satu section.
(Column, Table)	(1, 1)	Setiap kolom hanya memiliki paling banyak satu tabel.
(Column, Indicator)	(1, N)	Setiap column dapat memiliki banyak indicator, dan satu

		indicator hanya dimiliki oleh satu column.
(Indicator, Grade)	(1, N)	Setiap indicator dapat memiliki banyak grade, dan satu grade hanya dimiliki oleh satu indicator.
(Column, Content)	(1, N)	Setiap column dapat memiliki banyak content, dan satu content hanya dimiliki oleh satu column.
(Content, ContentGrade)	(1, N)	Setiap content dapat memiliki banyak content_grade, dan satu content_grade hanya dimiliki oleh satu content.
(Server, Api)	(1, N)	Setiap server dapat memiliki banyak api, dan satu api hanya dimiliki oleh satu server.

4.3.2. Desain Interface

Interface aplikasi didesain menggunakan framework css bootstrap. Berikut adalah beberapa desain interface yang akan digunakan dalam pembuatan aplikasi ini:

4.3.2.1. Registrasi dan Login pada Borang Akreditasi

1. Register

Aplikasi borang akreditasi saat pertama kali dibuka akan menampilkan halaman register. User diminta untuk memasukkan username dan password.

2. Login

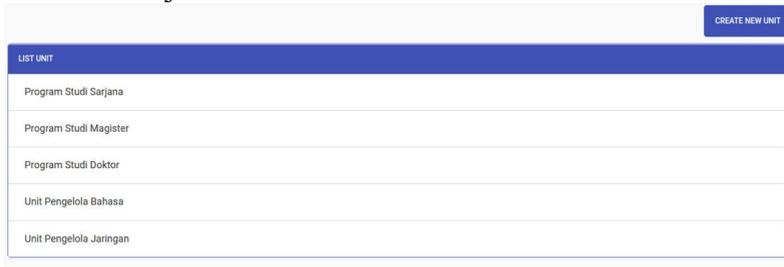
Halaman login digunakan untuk melakukan autentifikasi user yang sedang menggunakan aplikasi borang akreditasi berdasarkan rolanya.

3. Home

Halaman home akan menampilkan tampilan yang merujuk pada aktifitas utama user berdasarkan role yang sedang login pada aplikasi borang akreditasi.

4.3.2.2. Aplikasi Borang untuk Admin (role creator)

1. Manajemen user akses

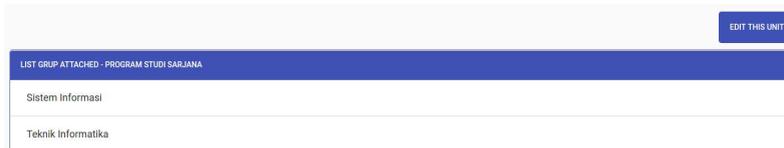


The screenshot shows a web interface for managing units. At the top right, there is a blue button labeled 'CREATE NEW UNIT'. Below it is a table with a blue header 'LIST UNIT'. The table contains five rows of unit names:

LIST UNIT
Program Studi Sarjana
Program Studi Magister
Program Studi Doktor
Unit Pengelola Bahasa
Unit Pengelola Jaringan

Gambar 4.4 Tampilan daftar Unit (level)

Halaman daftar unit ditujukan untuk mengetahui daftar unit yang telah dibuat dalam aplikasi borang akreditasi. Pada halaman daftar unit ditampilkan daftar unit yang dapat dipilih dan ditambahkan berupa list.



The screenshot shows a web interface for managing groups. At the top right, there is a blue button labeled 'EDIT THIS UNIT'. Below it is a table with a blue header 'LIST GRUP ATTACHED - PROGRAM STUDI SARJANA'. The table contains two rows of group names:

LIST GRUP ATTACHED - PROGRAM STUDI SARJANA
Sistem Informasi
Teknik Informatika

Gambar 4.5 Tampilan daftar Grup (jurusan)

Halaman grup akan muncul setelah unit dipilih. Pada halaman ini akan ditampilkan daftar jurusan/grup yang terdapat dalam setiap unit yang telah terdaftar di aplikasi borang akreditasi.



Assign user to group Sistem Informasi

EDIT THIS GROUP

Nama User Mimin

Role Editor

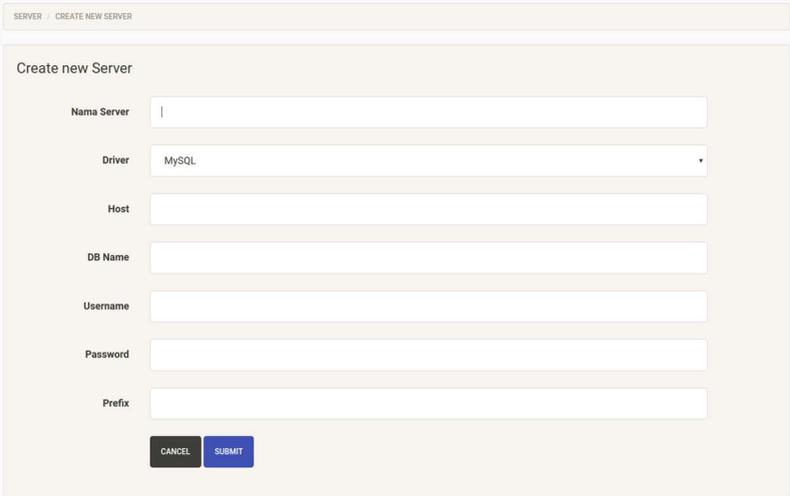
Season 2017

CANCEL SUBMIT

Gambar 4.6 Tampilan untuk menempatkan user ke grup

Setiap user memiliki hak akses pada grup yang berbeda. Oleh karena itu creator harus menempatkan setiap user pada grup yang seharusnya. Pada tampilan di atas terdapat 3 field yang terdiri dari nama user, role yang dimiliki dan masa berlakunya.

2. Menambah Server & API



SERVER / CREATE NEW SERVER

Create new Server

Nama Server

Driver MySQL

Host

DB Name

Username

Password

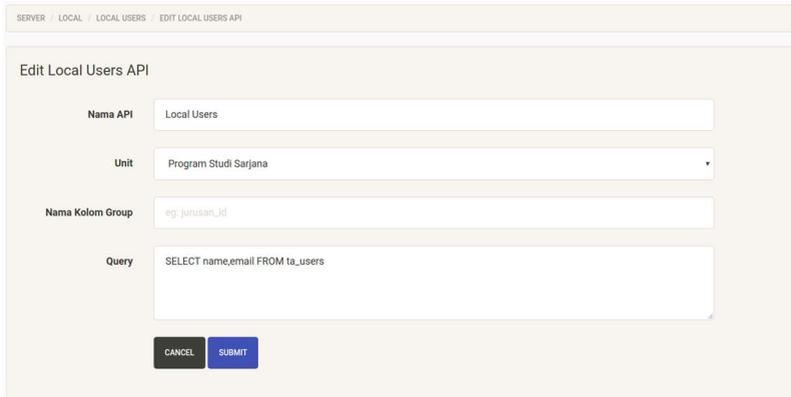
Prefix

CANCEL SUBMIT

Gambar 4.7 Tampilan untuk menambah server

Untuk menambahkan server pada aplikasi borang, diisikan beberapa isian konfigurasi seperti nama

server, driver yang digunakan, host, nama database yang digunakan, username, password dan prefix yang diinginkan pada aplikasi borang akreditasi.



The screenshot shows a web interface for editing a Local Users API. The breadcrumb trail at the top reads: SERVER / LOCAL / LOCAL USERS / EDIT LOCAL USERS API. The main heading is "Edit Local Users API". The form contains the following fields:

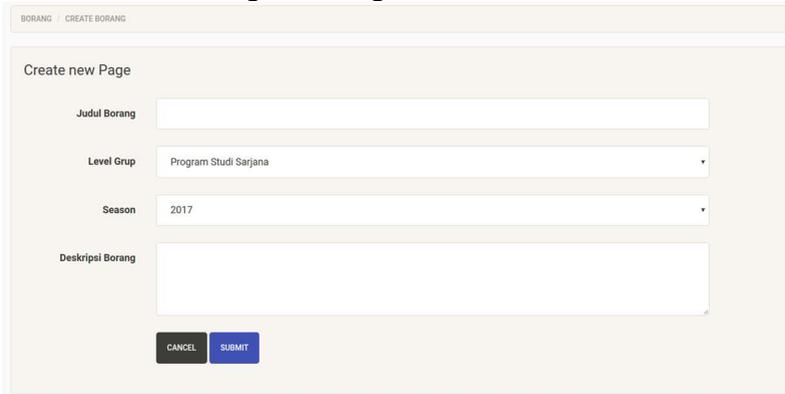
- Nama API:** A text input field containing "Local Users".
- Unit:** A dropdown menu with "Program Studi Sarjana" selected.
- Nama Kolom Group:** A text input field containing "eg: jurusan_id".
- Query:** A text area containing the SQL query "SELECT name,email FROM ta_users".

At the bottom of the form are two buttons: "CANCEL" and "SUBMIT".

Gambar 4.8 Tampilan untuk menambah API pada server

API pada server dapat ditambahkan melalui halaman API dengan mengisi nama API yang akan dibuat, untuk unit apa API tersebut dibuat, nama kolom grup dan query yang akan dieksekusi untuk menampilkan hasil data yang diinginkan.

3. Membuat bagian borang akreditasi



BORANG / CREATE BORANG

Create new Page

Judul Borang

Level Grup Program Studi Sarjana ▾

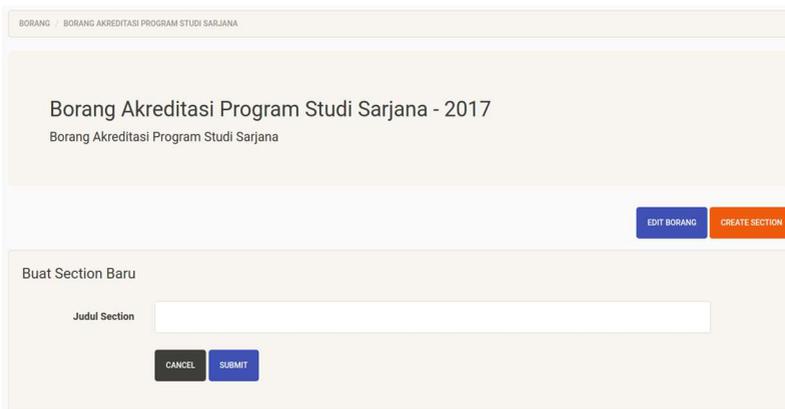
Season 2017 ▾

Deskripsi Borang

CANCEL SUBMIT

Gambar 4.9 Tampilan pembuatan borang (page)

Untuk memulai membuat borang dilakukan pada halaman baru borang dengan melakukan pembuatan setiap level yaitu level pertama berupa halaman. Pada halaman ini creator mengisi beberapa detail informasi seperti judul borang, level grup, season dan deskripsi borang.



BORANG / BORANG AKREDITASI PROGRAM STUDI SARJANA

Borang Akreditasi Program Studi Sarjana - 2017

Borang Akreditasi Program Studi Sarjana

EDIT BORANG CREATE SECTION

Buat Section Baru

Judul Section

CANCEL SUBMIT

Gambar 4.10 Tampilan pembuatan section baru

Level kedua pada tahap pembuatan borang yaitu membuat section dari halaman yang telah dibuat. Pada tahap ini creator hanya perlu mengisi judul section dari halaman tersebut.

BORANG / BORANG AKREDITASI PROGRAM STUDI SARJANA / IDENTITAS PENGISI BORANG

Borang Akreditasi Program Studi Sarjana

Borang Akreditasi Program Studi Sarjana

Identitas Pengisi Borang VIEW UBAH NILAI EDIT

Edit Section

Judul Section

IDENTITAS PENGISI BORANG

Nama Short answer + x

NIDN Short answer + x

NIDN Short answer + x

Jabatan Short answer + x

Tanggal Pengisian Date + x

Tanda Tangan Paragraph + x

Identitas Pengisi Borang + x

	A	B	C	D	E
1	Nama	NIDN	Jabatan	Tanggal Pengisian	Tanda Tangan
2	{{ name }}	{{ email }}			
3					
4					
5					
6					

[Change API](#)

Gambar 4.11 Tampilan pembuatan kolom (field) pada section

Level ketiga dari proses pembuatan borang yaitu membuat kolom. Kolom ini dapat disesuaikan dengan kebutuhan pembuatan borang. Setiap kolom dapat ditentukan apa tipe kolom yang diinginkan, yaitu dapat berupa jawaban singkat, opsi, check box dan jawaban panjang

4. Menambah indikator untuk tiap bagian

BORANG / BORANG AKREDITASI PROGRAM STUDI SARJANA / STANDAR 1. VISI, MISI, TUJUAN DAN SASARAN, SERTA STRATEGI PENCAPAIAN

Borang Akreditasi Program Studi Sarjana
Borang Akreditasi Program Studi Sarjana

STANDAR 1. Visi, Misi, Tujuan dan Sasaran, serta strategi Pencapaian

VIEW ISIAN NILAI EDIT

STANDAR 1. VISI, MISI, TUJUAN DAN SASARAN, SERTA STRATEGI PENCAPAIAN

1.1 Visi, Misi, Tujuan, dan Sasaran serta Strategi Pencapaian

Elemen Penilaian

1.1 Kejelasan dan kerealistikan visi, misi, tujuan, dan sasaran, serta strategi pencapaian sasaran Program Studi

Deskripsi indikator	Grade	Deskripsi
1.1.a Kejelasan dan kerealistikan visi, misi, tujuan, dan sasaran Program Studi	Sangat Baik	Memiliki visi, misi, tujuan, dan sasaran yang sangat jelas dan sangat realistik.
Variables null	Baik	Memiliki visi, misi, tujuan, dan sasaran jelas dan realistik.
	Cukup	Memiliki visi, misi, tujuan, dan sasaran yang cukup jelas namun kurang realistik.

Gambar 4.12 Tampilan pembuatan indikator

Masing masing bagian memiliki indikator penilaian yang berbeda. *Creator* dapat mengisi elemen penilaian yang ingin dibuat. Pada setiap indikator dapat ditetapkan *grade* serta deskripsinya.

4.3.2.3. Aplikasi Borang untuk Editor (role editor)

1. Pengisian borang akreditasi

BORANG / BORANG AKREDITASI PROGRAM STUDI SARJANA | STANDAR 2. TATA PAMONG, KEPEMIMPINAN, SISTEM PENGELOLAAN, DAN PENJAMINAN MUTU

Borang Akreditasi Program Studi Sarjana
Borang Akreditasi Program Studi Sarjana

STANDAR 2. Tata Pamong, Kepemimpinan, Sistem Pengelolaan, dan Penjaminan Mutu VIEW ISIAN NILAI EDIT -

STANDAR 2. TATA PAMONG, KEPEMIMPINAN, SISTEM PENGELOLAAN, DAN PENJAMINAN MUTU

2.1 Sistem Tata Pamong

Sistem tata pamong berjalan secara efektif melalui mekanisme yang disepakati bersama, serta dapat memelihara dan mengakomodasi semua unsur, fungsi, dan peran dalam program studi. Tata pamong didukung dengan budaya organisasi yang dicerminkan dengan ada dan tegaknya aturan, tatacara pemilihan pimpinan, etika dosen, etika mahasiswa, etika tenaga kependidikan, sistem penghargaan dan sanksi serta pedoman dan prosedur pelayanan (administrasi, perpustakaan, laboratorium, dan studio). Sistem tata pamong (input, proses, output dan outcome serta lingkungan eksternal yang menjamin terlaksananya tata pamong yang baik) harus diformulasikan, disosialisasikan, dilaksanakan, dipantau dan dievaluasi dengan peraturan dan prosedur yang jelas.

Uraikan secara ringkas sistem dan pelaksanaan tata pamong di Program Studi untuk membangun sistem tata pamong yang kredibel, transparan, akuntabel, bertanggung jawab dan adil.

Gambar 4.13 Tampilan isian borang

Laman pengisian borang menggunakan base yang sama seperti dengan tampilan pembuatan kolom, hanya saja lebih sederhana tanpa menu untuk mengubah kolom, yaitu berupa judul kolom dan field untuk input isian.

4.3.2.4. Aplikasi Borang untuk Auditor (role auditor)

1. Penilaian borang akreditasi

STANDAR 1. Visi, Misi, Tujuan dan Sasaran, serta strategi Pencapaian VIEW ISIAN NILAI EDIT -

STANDAR 1. VISI, MISI, TUJUAN DAN SASARAN, SERTA STRATEGI PENCAPAIAN Auditor - Sistem Informasi

1.1 Visi, Misi, Tujuan, dan Sasaran serta Strategi Pencapaian

Elemen Penilaian	Deskriptor	Harkat dan Peringkat
1.1 Kejelasan dan kerealistikan visi, misi, tujuan, dan sasaran, serta strategi pencapaian sasaran Program Studi	1.1.a Kejelasan dan kerealistikan visi, misi, tujuan, dan sasaran Program Studi	Kurang - Memiliki visi, misi, tujuan, dan sasaran jelas dan realistik.
	1.1.b Strategi pencapaian sasaran dengan rentang waktu yang jelas dan didukung oleh dokumen.	Kurang - Strategi pencapaian sasaran: (1) dengan tahapan waktu yang jelas, dan realistik (2) didukung dokumen yang lengkap.

Gambar 4.14 Tampilan laman evaluasi

Laman evaluasi berisi borang isian dari editor dan indikator yang dibuat oleh creator, untuk selanjutnya dinilai oleh auditor. Auditor hanya perlu memilih harkat dan peringkat dalam bentuk dropdown list.

4.4. Jenis & Struktur Penyajian Tabel dalam Aplikasi Borang

Dalam aplikasi borang terdapat beberapa data dalam bentuk tabel yang ditujukan untuk memudahkan pengguna dalam proses pengisian borang. Setiap informasi memiliki struktur penyajian tabel yang berbeda-beda sesuai kebutuhan.

Berikut ini merupakan penjabaran dari berbagai jenis tabel dalam aplikasi borang.

4.4.1. Tabel Induk (Master Table)

Tabel induk ialah jenis penyajian tabel yang menampilkan data yang tersedia secara terperinci, sehingga pengguna dapat memperoleh setiap informasi yang diinginkan. Dalam hal ini, tabel induk (summary table) merupakan ringkasan hasil dari berbagai variabel dan dimuat dalam satu tabel sehingga sangat efisien.

Dalam aplikasi borang, terdapat beberapa jenis penyajian data yang disajikan dalam tabel induk yaitu sebagai berikut:

- a. Tabel Umpan Balik (Standar 1) - Buku 3A

Tabel 4.3 Contoh tabel umpan balik

Umpan Balik dari (1)	Isi Umpan Balik (2)	Tindak Lanjut (3)
Dosen		
Mahasiswa		
Alumni		
Pengguna lulusan		

Tabel diatas terdiri dari 2 variabel yaitu isi umpan balik dan tindak lanjut, variabel pertama dan kedua memiliki constraint tipe data text.

b. Tabel Pencapaian mahasiswa (Standar 3)- Buku 3A

Tabel 4.4 Contoh tabel pencapaian mahasiswa

No.	Nama Kegiatan dan Waktu Penyelenggaraan	Tingkat (Lokal, Wilayah, Nasional, atau Internasional)	Prestasi yang Dicapai
(1)	(2)	(3)	(4)

Tabel diatas terdiri dari 3 variabel dengan constraint tipe data yang sama yaitu text

c. Tabel Layanan Mahasiswa (Standar 3)- Buku 3A

Tabel 4.5 Contoh tabel layanan mahasiswa

No.	Jenis Pelayanan kepada Mahasiswa	Bentuk kegiatan, Pelaksanaan dan Hasilnya
(1)	(2)	(3)
1	Bimbingan dan konseling	
2	dan bakat (ekstra kurikuler)	
3	pinaan <i>soft skills</i>	
4	Beasiswa	
5	Kesehatan	

Tabel diatas hanya terdiri dari 2 variabel yang berisi penjelasan setiap jenis pelayanan kepada mahasiswa dengan type data text.

d. Tabel Data dosen tetap (Standart 4)-Buku 3A

Tabel 4.6 Contoh tabel data dosen tetap

No.	Nama Dosen Tetap	NIDN**	Tgl. Lahir	Jabatan Akademik***	Gelar Akademik	Pendidikan S1, S2, S3 dan Asal PT*	Bidang Keahlian untuk Setiap Jenjang Pendidikan
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

Tabel diatas merupakan data dosen tetap yang keahliannya di dalam PS ataupun diluar PS. Tabel memiliki 8 variabel dengan tipe data berikut:

- (1) No: integer
 - (2) Nama dosen tetap: varchar
 - (3) NIDN: integer
 - (4) Tanggal lahir: date
 - (5) Jabatan Akademik: varchar
 - (6) Gelar akademik: varchar
 - (7) Pendidikan: text
 - (8) Bidang keahlian: text
- e. Tabel aktivitas mengajar dosen (Standart 4) - Buku 3A

Tabel 4.7 Contoh tabel aktivitas mengajar dosen

No.	Nama Dosen Tetap	Bidang Keahlian	Kode Mata Kuliah	Nama Mata Kuliah	Jumlah Kelas	Jumlah Pertemuan yang Direncanakan	Jumlah Pertemuan yang Dilaksanakan
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Jumlah							

Tabel diatas terdiri dari 8 variabel isian dari aktivitas mengajar yang dilakukan oleh setiap dosen tetap maupun tidak tetap

- f. Tabel Program tugas belajar (Standart 4) - Buku 3A

Tabel 4.8 Contoh tabel program tugas belajar

No.	Nama Dosen	Jenjang Pendidikan Lanjut	Bidang Studi	Perguruan Tinggi	Negara	Tahun Mulai Studi
(1)	(2)	(3)	(4)	(5)	(6)	(7)

Tabel diatas merupakan tabel peningkatan kemampuan dosen tetap melalui program tugas belajar dalam bidang yang sesuai dengan bidang PS. Tabel ini memiliki 7 variabel yang diisi sesuai dengan header tabel.

g. Kegiatan dosen tetap

Tabel 4.9 Contoh tabel kegiatan dosen tetap

No.	Nama Dosen	Jenis Kegiatan*	Tempat	Waktu	Sebagai	
					Penyaji	Peserta
(1)	(2)	(3)	(4)	(5)	(6)	

Tabel merupakan tabel kegiatan dosen tetap dengan 6 variabel dan variabel terakhir yang memiliki opsi karakteristik data

h. Tabel pencapaian reputasi

Tabel 4.10 Contoh tabel pencapaian reputasi

No.	Nama Dosen	Prestasi yang Dicapai*	Waktu Pencapaian	Tingkat (Lokal, Nasional, Internasional)
(1)	(2)	(3)	(4)	(5)

Tabel merupakan tabel pencapaian reputasi dan prestasi dengan 5 variabel

i. Tabel sks program studi

Tabel 4.11 Contoh tabel sks program studi

Jenis Mata Kuliah	sks	Keterangan
(1)	(2)	(3)
Mata Kuliah Wajib		
Mata Kuliah Pilihan		
Jumlah Total		

Tabel merupakan tabel sks program studi dengan 3 variabel

4.4.2. Tabel Silang/ Nested

Tabel silang ialah penyajian yang ditujukan untuk menampilkan data ringkas dan seefektif mungkin. Tabel ini berfungsi sebagai data/informasi untuk topik yang akan didiskusikan dlm text dari persilangan antara 2 variabel atau lebih serta bertujuan untuk membandingkan 2 atau lebih variabel yang terkait. Dalam aplikasi borang, terdapat beberapa jenis penyajian data yang disajikan dalam tabel silang yaitu sebagai berikut:

- a. Tabel Profil Mahasiswa & Lulusan (Standar 3) Buku 3A

Tabel 4.12 Contoh tabel profil mahasiswa & lulusan

Tahun Akademik	Daya Tampung	Jumlah Calon Mahasiswa Reguler		Jumlah Mahasiswa Baru		Jumlah Total Mahasiswa		Jumlah Lulusan		IPK Lulusan Reguler		
		Ikut Seleksi	Lulus Seleksi	Reguler bukan Transfer	Transfer	Reguler bukan Transfer	Transfer	Reguler bukan Transfer	Transfer	Min	Rata	Maks
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
TS-4												
TS-3												
TS-2												
TS-1												
TS												
Jumlah												

Tabel diatas terdiri dari 5 parameter dan 13 variabel dalam 5 atribut kelompok (nested) yang seluruhnya memiliki tipe data integer, begitu juga dengan data mahasiswa non reguler di bawah ini yang memiliki parameter yang sama

Tabel 4.13 Contoh tabel profil mahasiswa & lulusan

Tahun Akademi	Daya Tampung	Jumlah Calon Mahasiswa		Jumlah Mahasiswa Baru		Jumlah Total Mahasiswa	
		Ikut Seleksi	Lulus Seleksi	Non-Reguler	Transfer ⁽³⁾	Non-Reguler	Transfer ⁽³⁾
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
TS-4							
TS-3							
TS-2							
TS-1							
TS							

b. Tabel sks rata-rata per semester

Tabel 4.14 Contoh tabel sks rata-rata per semester

No.	Nama Dosen Tetap	sks Pengajaran pada			sks Pene-litian	sks Pengabdian kepada Masyarakat	sks Manajemen**		Jumlah sks
		PS Sendiri	PS Lain PT Sen-diri	PT Lain			PT Sen-diri	PT Lain	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Jumlah									
Rata-rata*									

Tabel diatas terdiri dari 10 variabel dengan 2 nested atribut.

c. Tabel struktur kurikulum

Tabel 4.15 Contoh tabel struktur kurikulum

Smt	Kode MK	Nama Mata Kuliah*	Bobot sks	sks MK dalam Kurikulum		Bobot Tugas**	Kelengkapan****			Unit/ Jur/ Fak Penyelenggara
				Inti*	Institusional		Deskripsi	Silabus	SA P	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
I										
II										
Ds t										
Total sks										

Pada tabel diatas berisi 11 variabel dengan 2 nested atribut pada 3 atau lebih parameter yang digunakan

Tabel diatas merupakan jenis tabel kumulatif dengan menyertakan jumlah tenaga pendidik dari 4 atau lebih parameter dari 11 jenis variabel

2. Tabel Distribusi Frekuensi Persentase

Tabel 4.18 Contoh tabel distribusi frekuensi persentase

No.	Jenis Kemampuan	Tanggapan Pihak Pengguna				Rencana Tindak Lanjut oleh Program Studi
		Sangat Baik	Baik	Cukup	Kurang	
		(%)	(%)	(%)	(%)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	Integritas (etika dan moral)					
2	Keahlian berdasarkan bidang ilmu (profesionalisme)					
3	Bahasa Inggris					
4	Penggunaan Teknologi Informasi					
5	Komunikasi					
6	Kerjasama tim					
7	Pengembangan diri					
Total		(a)	(b)	(c)	(d)	

Tabel diatas merupakan jenis tabel persentase dengan isian yang menggunakan satuan % dalam perhitungannya.

BAB V

IMPLEMENTASI

Pada bab ini, akan dijelaskan mengenai implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi akan menjelaskan mengenai lingkungan implementasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

5.1. Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi pada tabel berikut:

Table 5.1 Lingkungan implementasi

Prosesor	Intel® Core™ i3-4030U CPU @ 1.90GHz × 4
Memory	4096 MB RAM
Sistem Operasi	Ubuntu 16.04 (64-bit)

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti editor, database, server; bahasa pemrograman, dan library yang disajikan dalam tabel berikut:

Table 5.2 Perangkat lunak yang digunakan

Websserver	Apache 2.4
Bahasa Pemrograman	PHP 7.1, Javascript ECMAS
Database	<ul style="list-style-type: none">- MySQL (development)- SQL Server (testing)

Editor (IDE)	Cscpro Panel (Beta)
Browser	Google Chrome 56
Library	<ul style="list-style-type: none"> - jQuery 2.1 (Javascript lib) - Bootstrap v3.3.6 (Layout) - Composer (php package manager) - Laravel 5.4 (PHP Framework)

Server yang digunakan dalam pengembangan dan uji coba aplikasi ini ialah virtual private server DigitalOcean dengan spesifikasi berikut.

Table 5.3 Spesifikasi *server* yang digunakan

Sistem Operasi	Ubuntu 16.04 Server (64-bit)
CPU	Xeon
RAM	796 MB RAM
Internal Memory	20 GB
Lokasi	Singapore

5.2. Pengembangan Sistem

Desain sistem adalah salah satu implementasi desain arsitektur aplikasi dari implementasi kebutuhan fungsional dan analisis data di tahap sebelumnya. Berikut ini merupakan implementasi desain sistem untuk memenuhi setiap kebutuhan fungsional yang telah dideskripsikan

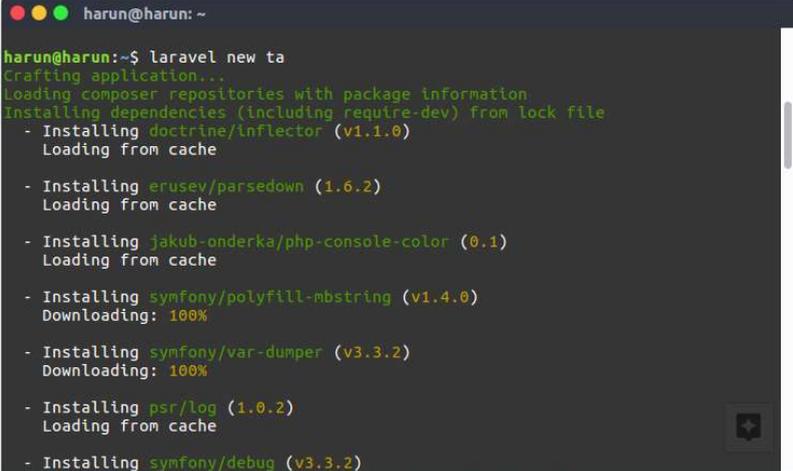
(Iterative Development)

5.2.1. Iterasi Pertama

Implementasi (Fase Construction)

5.2.1.1. Inisiasi laravel project

Inisiasi dilakukan dengan membuat project baru dengan framework laravel menggunakan command line berikut: “*laravel new ta*” atau “*composer create-project --prefer-dist laravel/laravel ta*”



```

harun@harun: ~
harun@harun:~$ laravel new ta
Crafting application...
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
- Installing doctrine/inflector (v1.1.0)
  Loading from cache
- Installing erusev/parsedown (1.6.2)
  Loading from cache
- Installing jakub-ondarka/php-console-color (0.1)
  Loading from cache
- Installing symfony/polyfill-mbstring (v1.4.0)
  Downloading: 100%
- Installing symfony/var-dumper (v3.3.2)
  Downloading: 100%
- Installing psr/log (1.0.2)
  Loading from cache
- Installing symfony/debug (v3.3.2)

```

Kode Program 5.1 Command line untuk inisiasi project laravel

Setelah selesai inisiasi maka akan muncul pesan “Application Ready! ..”. Dan project siap untuk digunakan.

5.2.1.2. Pembuatan database

Skema tabel pada database dibuat menggunakan fitur *database migration* pada framework laravel dengan command “*php artisan make:migration create_pages_table*” yang kemudian akan membuat file migration pada folder database/migrations. Lalu edit file tersebut dan tambahkan skema tabel seperti pada kode program 5.2 berikut:

```

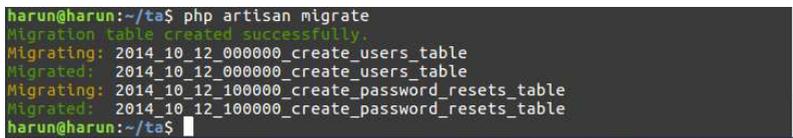
Schema::create( 'pages', function ( Blueprint $table ) {
    $table->increments( 'id' );
    $table->string( 'title' );
    $table->integer( 'level_id' )->unsigned();
    $table->integer( 'season_id' )->unsigned();
    $table->text( 'description' )->nullable();
    $table->timestamps();

    $table->foreign( 'level_id' )
        ->references( 'id' )
        ->on( 'levels' )
        ->onDelete( 'cascade' )
    ;
    $table->foreign( 'season_id' )
        ->references( 'id' )
        ->on( 'seasons' )
        ->onDelete( 'cascade' )
    ;
} );

```

Kode Program 5.2 Contoh skema tabel *file migration*

Tabel yang akan dibuat sesuai dengan daftar tabel yang terdapat pada tabel 4.1. Setelah seluruh tabel Kemudian dilakukan *migration* dengan *command line* seperti berikut:



```

harun@harun:~/ta$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
harun@harun:~/ta$

```

Kode Program 5.3 Command line untuk melakukan migration

5.2.3. Pembuatan otentikasi dan role (iterasi pertama)

5.2.3.1 Pembuatan otentikasi user

Laravel memiliki fitur pre-built untuk otentikasi (user login, register, forgot password, dll) cukup menggunakan command line berikut: “*php artisan make:auth*” seperti yang tertera pada kode program 5.4

```
harun@harun:~/ta$ php artisan make:auth
Authentication scaffolding generated successfully.
harun@harun:~/ta$ ls
```

Kode Program 5.4 Command line untuk membuat otentikasi

Kemudian tabel user, sistem login, forgot password, registrasi hingga tampilannya akan secara otomatis digenerate.

Gambar 5.1 Tampilan halaman register

5.2.3.2 Pembuatan *CRUD* untuk Unit (*level*)

Unit adalah hirarki pertama dalam manajemen hak akses pada aplikasi ini. Susunannya yaitu Unit → Grup → User. Pertama-tama buat resource controller untuk melakukan Create Read Update Delete (CRUD) untuk unit seperti kode program 5.5 berikut:

```
harun@harun:~/ta$ php artisan make:controller LevelController --model=Level --resource
controller created successfully.
harun@harun:~/ta$
```

Kode Program 5.5 Command line untuk membuat *controller*

Kemudian tambahkan route (penghubung antara URL dengan *controller* pada Laravel) yang mengarah ke *controller* seperti berikut, di laman routes/web.php:

```
Route::resource( 'level', 'LevelController' );
Route::post( 'group/{group}/user', 'GroupController@addUser' )->name( 'group.adduser' );
Route::delete( 'group/{group}/user', 'GroupController@delUser' )->name( 'group.deluser' );
Route::resource( 'group', 'GroupController' );
```

Kode Program 5.6 Contoh file route untuk controller

Lalu cek route list dari *command line* untuk melihat *controller* yang sudah terdaftar dengan *command* “*php artisan route:list*”, hasilnya akan tampak seperti gambar 5.2 berikut:

POST	level	level.store	App\Http\Controllers\LevelController@store
GET HEAD	level	level.index	App\Http\Controllers\LevelController@index
GET HEAD	level/create	level.create	App\Http\Controllers\LevelController@create
PUT PATCH	level/{level}	level.update	App\Http\Controllers\LevelController@update
GET HEAD	level/{level}	level.show	App\Http\Controllers\LevelController@show
DELETE	level/{level}	level.destroy	App\Http\Controllers\LevelController@destroy
GET HEAD	level/{level}/edit	level.edit	App\Http\Controllers\LevelController@edit

Gambar 5.2 Command line untuk melihat route yang terdaftar

Terlihat *method* untuk index, create, show, update dan destroy pada controller level sudah dibuat. Selanjutnya tinggal membuat *view* dan *processing* pada *controller* nya.

Berikut merupakan isi method dari *GrupController*, untuk method `__construct` dan `index` yang mengambil model dari tabel *Group*, kemudian menampilkannya dalam view *group.index* seperti terlihat pada kode program 5.7 berikut.

```

12 class GroupController extends Controller
13 {
14     /**
15      * Instantiate a new controller instance.
16      *
17      * @return void
18      */
19     public function __construct()
20     {
21         $this->middleware( 'auth' );
22         $this->middleware( 'auth.isAdmin' );
23     }
24
25     /**
26      * Display a listing of the resource.
27      *
28      * @return \Illuminate\Http\Response
29      */
30     public function index()
31     {
32         $group = Group::all();
33         return view( 'group.index', compact( 'group' ) );
34     }

```

Kode Program 5.7 Contoh isi *method* index pada controller

Berikut isi method dari create dan store untuk *LevelController* seperti terlihat pada kode program 5.8:

```

36 ▾ /**
37  * Show the form for creating a new resource.
38  *
39  * @return \Illuminate\Http\Response
40  */
41 public function create()
42 ▾ {
43     $is_edit = false;
44     $group = new Group;
45     $level = Level::all();
46     return view( 'group.store', compact( 'is_edit', 'group', 'level' ) );
47 }
48
49 ▾ /**
50  * Store a newly created resource in storage.
51  *
52  * @param \Illuminate\Http\Request $request
53  * @return \Illuminate\Http\Response
54  */
55 public function store(Request $request)
56 ▾ {
57     $this->validate( $request, [
58         'name' => 'required|unique:groups,name',
59         'level_id' => 'required|numeric|exists:levels,id',
60     ] );
61
62     $group = new Group;
63     return $this->storeUpdate( $request, $group );
64 }
65

```

Kode Program 5.8 Contoh isi method create pada controller

Untuk method edit dan update kurang lebih sama seperti diatas.

Untuk *View*, aplikasi ini pada umumnya menggunakan 3 view untuk masing-masing controller nya. Dengan rincian sebagai berikut. Kode program 5.9 berikut adalah view untuk index: (level.index)

```

1 |@extends('layouts.app')
2 |
3 |@section('content')
4 |<ul class="breadcrumb">
5 |<li><a href="{{ url( '/' ) }}">Home</a></li>
6 |<li class="active"><a href="{{ route( 'level.index' ) }}">Unit</a></li>
7 |</ul>
8 |
9 |<div class="text-right pb-3">
10 |<a href="{{ route( 'level.create' ) }}" class="btn btn-primary">Create new Unit</a>
11 |</div>
12 |
13 |<div class="panel panel-primary">
14 |<div class="panel-heading">
15 |<h3 class="panel-title">List Unit</h3>
16 |</div>
17 |<div class="list-group">
18 |@foreach ( $level as $level )
19 |<a href="{{ route( 'level.show', $level->id ) }}" class="list-group-item">
20 |    {{ $level->name }}
21 |</a>
22 |
23 |@empty
24 |<div class="text-center list-group-item">
25 |    There was no unit currently.
26 |</div>
27 |
28 |@endforeach
29 |</div>
30 |</div>
31 |
32 |@endsection

```

Kode Program 5.9 Contoh isi index pada view

Kemudian view untuk show: (level.show) dapat dilihat pada kode program 5.10 berikut:

```

1 |@extends('layouts.app')
2 |
3 |@section('content')
4 |<ul class="breadcrumb">
5 |<li><a href="{{ route( 'level.index' ) }}">Unit</a></li>
6 |<li class="active"><a href="{{ route( 'level.show', $level->id ) }}">{{ $level->name }}</a></li>
7 |</ul>
8 |
9 |<div class="text-right pb-3">
10 |<a href="{{ route( 'level.edit', $level->id ) }}" class="btn btn-primary">Edit this unit</a>
11 |</div>
12 |
13 |<div class="panel panel-primary">
14 |<div class="panel-heading">
15 |<h3 class="panel-title">List Grup Attached - {{ $level->name }}</h3>
16 |</div>
17 |<div class="list-group">
18 |@foreach ( $groups as $group )
19 |<a href="{{ route( 'group.show', $group->id ) }}" class="list-group-item">
20 |    {{ $group->name }}
21 |</a>
22 |
23 |@empty
24 |<div class="text-center list-group-item">
25 |    There was no grup assigned currently.
26 |</div>
27 |
28 |@endforeach
29 |</div>
30 |</div>
31 |
32 |@endsection

```

Kode Program 5.10 Contoh isi show pada view

Lalu view untuk store seperti terlihat pada kode program 5.11 berikut:

```

1 @extends('layouts.app')
2
3 @section('content')
4 <ul class="breadcrumb">
5 <li><a href="{{ route( 'level.index' ) }}">Unit</a></li>
6 @if ( $is_edit )
7 <li><a href="{{ route( 'level.show' , $level->id ) }}">Edit {{ $level->name }}</a></li>
8 @else
9 <li><a href="{{ route( 'level.store' ) }}">Create Unit</a></li>
10 @endif
11 </ul>
12
13 <form method="post" action="{{ ( $is_edit ? route( 'level.show' , $level->id ) : route( 'level.index' ) ) }}" class="form-horizontal well">
14 <legend>{{ $is_edit ? 'Edit ' . $level->name : 'Create new' }} unit</legend>
15 <div class="form-group">
16 <label for="level-name" class="col-md-2 control-label">Nama Unit</label>
17 <div class="col-md-9">
18 <input type="text" class="form-control" name="name" id="level-name" value="{{ old( 'name' , $level->name ) }}" required>
19 </div>
20 </div>
21 <div class="form-group">
22 <div class="col-md-9 col-md-offset-2">
23 <button onClick="window.history.go(-1); return false;" class="btn btn-default">Cancel</button>
24 <button type="submit" class="btn btn-primary">Submit</button>
25 </div>
26 </div>
27 {{ $is_edit ? method_field( 'PUT' ) : '' }}
28 {{ csrf_field() }}
29 </form>
30
31 @endsection

```

Kode Program 5.11 Contoh isi store pada view

a. Pembuatan grup

CRUD pada controller grup kurang lebih sama dengan CRUD pada controller unit (level). Yang berbeda hanyalah grup merupakan bagian dari unit (reversed one-to-many).

b. Pembuatan role (manajemen akses)

Role disini merupakan kolom pivot dari table Group, User dan Season. Seperti pada gambar 5.3 berikut:

Assign user to group Sistem Informasi

Nama User

Role

Season

LIST USER ATTACHED - SISTEM INFORMASI

User	Role	Season
Editor SI	2	2017
Auditor SI	3	2017

Gambar 5.3 Tampilan manajemen akses (memasukkan role ke user)

5.2.4. Pembuatan borang awal

a. Pembuatan laman borang

Hirarki pada borang yaitu dari Page (Laman) → Section (Bagian) → Column / Table → Content (Isian) → Review (Nilai). Pembuatan laman borang menggunakan CRUD yang kurang lebih sama seperti CRUD pada LevelController dan GroupController.

b. Pembuatan section / kolom / tabel

CRUD pada Section cukup berbeda dibanding fitur lainnya. Karena disinilah letak dinamis nya. Contohnya pada kolom type yang memiliki banyak varian input, belum lagi pada bagian tabel yang memiliki table database sendiri.

View pada bagian ini cenderung dilakukan pada sisi client-side (JavaScript), salah satu contohnya seperti berikut ini:

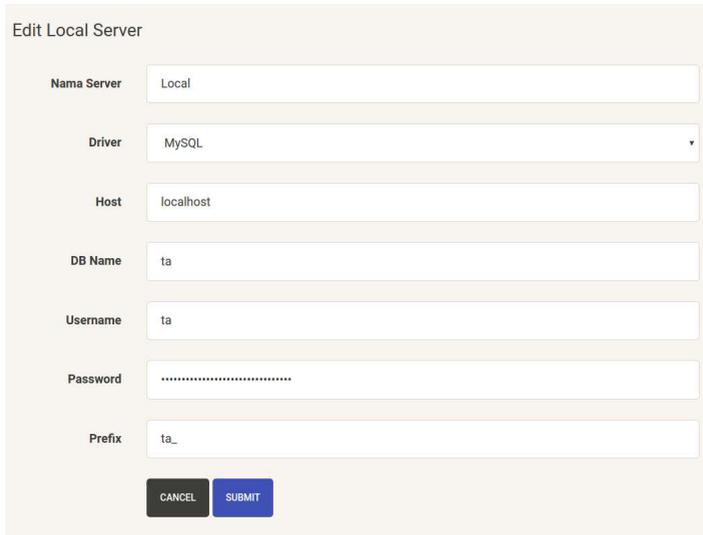
```

492 ▾ /* Vars */
493 var loading = '<div class="loading text-center">1, ratio<5 ); (3+ratio)/2 ) |
|                                                                | Sangat Baik | =IF( ratio >= 5 )                           |
|                                                                |             | Add grade                                   |
|                                                                |             | Add more indicator                          |

**Gambar 5.14** Tampilan pengisian formulasi pada indikator

### 5.2.3.3. Server & API

Server pada aplikasi ini merupakan penghubung / koneksi ke database server lain. Jadi creator bisa menghubungkan isian pada tabel dengan database terkait, seperti kemahasiswaan ataupun kepegawaian. Gambar 5.15 merupakan tampilan untuk penambahan server, dengan memasukkan tipe driver (MySQL / SQL Server / Postgres / SQL Lite).



Edit Local Server

Nama Server

Driver

Host

DB Name

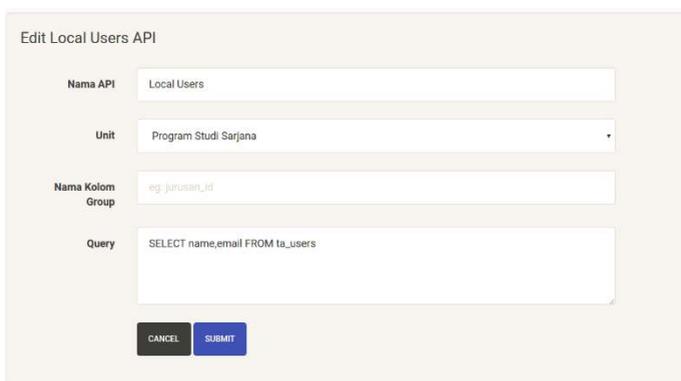
Username

Password

Prefix

**Gambar 5.15 Tampilan pembuatan server**

Sedangkan API (Application Programming Interface) berisi data query SQL yang dilakukan melalui server yang dipilih sebelumnya. Query inilah yang akan mengambil data tertentu pada server tersebut. Tampilan pembuatan API terlihat seperti pada gambar 5.16



Edit Local Users API

Nama API

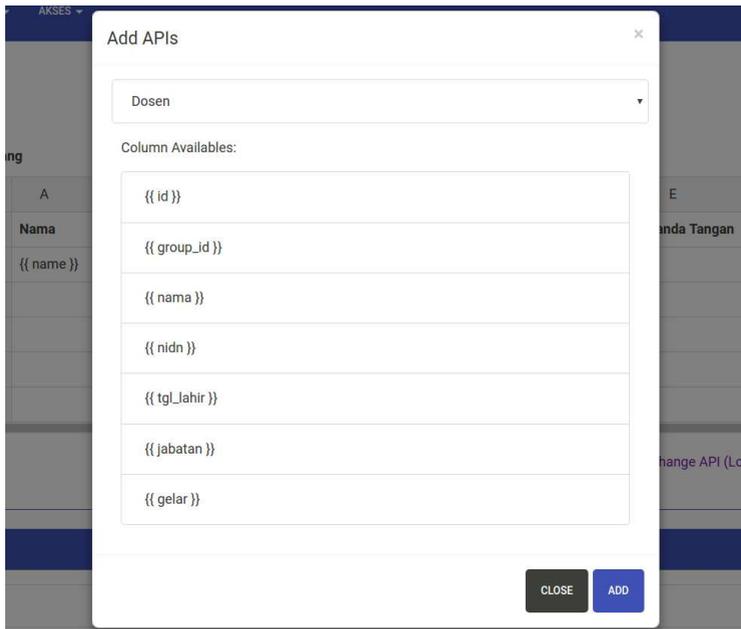
Unit

Nama Kolom Group

Query

**Gambar 5.16 Tampilan pembuatan API**

Kemudian API yang sudah dibuat dapat dihubungkan dengan tabel borang oleh creator melalui button add/change API seperti yang terlihat pada gambar 5.17.



**Gambar 5.17 Tampilan menghubungkan API dengan tabel borang**

Lalu pada tabel isian borang saat diisi editor akan otomatis terisi data sesuai dengan yang dideskripsikan pada API sebelumnya, seperti yang terlihat pada gambar 5.18

Bagi PS yang dibina oleh Departemen Pendidikan Nasional, sebutkan nama dosen tetap institusi yang terdaftar sebagai dosen 034/DIKTI/Kep/2002, dalam tabel di bawah ini.

| No. | Nama Dosen Tetap | NIDN**      | Tgl. Lahir | Jabatan Akademik   | Gelar Akademik |
|-----|------------------|-------------|------------|--------------------|----------------|
| 1   | Dosen satu       | 11234567890 | 1978-09-13 | Ketua Jurusan      | S.Kom. MSc.    |
| 2   | Dosen Duwa       | 11234567894 | 1977-06-15 | Sekretaris Jurusan | S.Kom. M.Kom.  |
| 3   | Dosen Tiga       | 11234567838 | 1978-04-25 | Dosen              | S.Kom. M.Si.   |

**Gambar 5.18 Tampilan pada tabel isian borang**

*[halaman ini sengaja dikosongkan]*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini, akan dijelaskan hasil dari pengujian aplikasi serta pembahasan terhadap hasil pengujian yang dilakukan. Berikut adalah penjabaran untuk hasil dan pembahasan pengujian yang dilakukan.

#### **6.1. Hasil Pengujian**

Pengujian yang dilakukan pada aplikasi dijabarkan pada sub bab ini. Testing yang dilakukan yaitu functional testing (dilakukan sendiri), non-functional testing (menggunakan tools Apache JMeter) dan user acceptance test (survey dan demo langsung ke end-user).

##### **6.1.1. Pengujian Fungsional (*Functional Testing*)**

Pengujian sistem dilakukan untuk mengetahui apakah fitur yang dibuat pada aplikasi dapat berjalan dengan baik. Semua fitur akan dicoba, tetapi dokumentasi pada buku ini hanya pada fitur-fitur utama aplikasi saja. Tabel berikut merupakan daftar fitur yang diuji.

**Tabel 6.1 Daftar fitur utama aplikasi yang akan diuji**

| <b>No</b> | <b>Fitur</b>         |
|-----------|----------------------|
| 1         | Register             |
| 2         | Login                |
| 3         | Manajemen User Akses |
| 4         | Tambah Server & API  |

|    |                              |
|----|------------------------------|
|    |                              |
| 5  | Membuat Borang               |
| 6  | Membuat Section              |
| 7  | Membuat Kolom                |
| 8  | Membuat Table                |
| 9  | Membuat Formulasi pada Table |
| 10 | Membuat Indikator            |
| 11 | Pengisian Borang             |
| 12 | Penilaian Borang             |

Berikut adalah hasil pengujian fitur yang terangkum dalam tabel 6.2 berikut:

**Tabel 6.2** Tabel *Functional Testing*

| <b>Process</b>              | <b>Input</b> | <b>Expected Result</b> | <b>Actual Result</b> |
|-----------------------------|--------------|------------------------|----------------------|
| <b>Manajemen User Akses</b> |              |                        |                      |
| <b>Membuat</b>              | Nama unit    | Unit                   | Unit                 |

|                                |                                                                             |                     |                     |
|--------------------------------|-----------------------------------------------------------------------------|---------------------|---------------------|
| <b>Unit Baru</b>               |                                                                             | bertambah           | bertambah           |
|                                | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                     |                     |
| <b>Memperbarui Unit</b>        | Nama unit                                                                   | Nama unit berubah   | Nama unit berubah   |
|                                | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                     |                     |
| <b>Membuat Grup</b>            | Nama grup                                                                   | Grup bertambah      | Grup bertambah      |
|                                | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                     |                     |
| <b>Memperbarui Grup</b>        | Nama grup                                                                   | Nama grup berubah   | Nama grup berubah   |
|                                | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                     |                     |
| <b>Memasukkan User ke Grup</b> | Nama user, role, season                                                     | User role bertambah | User role bertambah |
|                                | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                     |                     |
| <b>Tambah Server &amp; API</b> |                                                                             |                     |                     |

|                           |                                                                                             |                                   |                                   |
|---------------------------|---------------------------------------------------------------------------------------------|-----------------------------------|-----------------------------------|
| <b>Menambah Server</b>    | Nama server, driver, localhost, nama database, username database, password database, prefix | Server bertambah, Tidak ada error | Server bertambah, Tidak ada error |
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.                 |                                   |                                   |
| <b>Memperbarui Server</b> | Nama server, driver, localhost, nama database, username database, password database, prefix | Server berubah, Tidak ada error   | Server berubah, Tidak ada error   |
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.                 |                                   |                                   |
| <b>Manajemen Borang</b>   |                                                                                             |                                   |                                   |
| <b>Membuat Borang</b>     | Judul borang, Level grup, Season, Deskripsi                                                 | Borang bertambah                  | Borang bertambah                  |

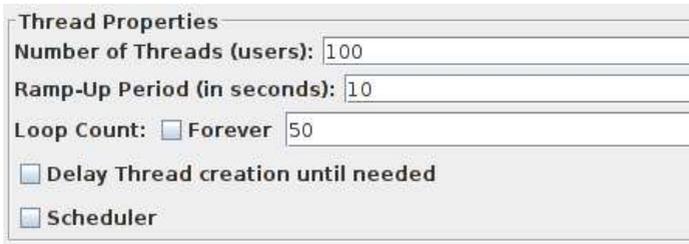
|                                     |                                                                             |                                           |                                           |
|-------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------|
|                                     | Borang                                                                      |                                           |                                           |
|                                     | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                           |                                           |
| <b>Membuat Section</b>              | Judul section                                                               | Section bertambah                         | Section bertambah                         |
|                                     | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                           |                                           |
| <b>Membuat Kolom</b>                | Tipe kolom, judul kolom                                                     | Kolom bertambah                           | Kolom bertambah                           |
|                                     | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                           |                                           |
| <b>Membuat Table</b>                | Jumlah baris, jumlah kolom, isi kolom                                       | Table bertambah                           | Table bertambah                           |
|                                     | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                           |                                           |
| <b>Formulasi pada Table</b>         |                                                                             |                                           |                                           |
| <b>Membuat Formulasi pada Table</b> | Formula [sum, max, avg, min]                                                | Formula berhasil diparsing ke dalam table | Formula berhasil diparsing ke dalam table |

|                           |                                                                             |                                              |                                              |
|---------------------------|-----------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------|
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                              |                                              |
| <b>Validasi formula</b>   | Formula                                                                     | Formula sesuai dengan rumus yang dimasukkan  | Formula sesuai dengan rumus yang dimasukkan  |
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                              |                                              |
| <b>Membuat Indikator</b>  | Elemen penilaian, Deskripsi Indikator, Variable, Grade, Deskripsi           | Indikator berhasil diparsing ke tiap section | Indikator berhasil diparsing ke tiap section |
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                              |                                              |
| <b>Validasi Indikator</b> | Indikator                                                                   | Penilaian (review) sesuai dengan Indikator   | Penilaian (review) sesuai dengan Indikator   |
|                           | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                              |                                              |
| <b>Pengisian Borang</b>   |                                                                             |                                              |                                              |

|                         |                                                                             |                                                     |                                                     |
|-------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| <b>Pengisian Borang</b> | Isian borang                                                                | Isian berhasil dimasukkan sesuai dengan grup editor | Isian berhasil dimasukkan sesuai dengan grup editor |
|                         | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                                     |                                                     |
| <b>Penilaian Borang</b> |                                                                             |                                                     |                                                     |
| <b>Penilaian Borang</b> | Nilai Borang                                                                | Nilai berhasil dimasukkan sesuai dengan ID user     | Nilai berhasil dimasukkan sesuai dengan ID user     |
|                         | Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error. |                                                     |                                                     |

### 6.1.2. Pengujian Non Fungsional (*Non-Functional Testing*)

Pada testing ini menggunakan tool Apache JMeter, untuk mengukur tingkat reliability aplikasi. Disimulasikan terdapat 100 user yang mengakses secara bersamaan dengan durasi waktu 10 detik, dan diulang sebanyak 50 kali.



**Gambar 6.1 Rules pada Apache JMeter**

Testing dilakukan dengan asumsi worst-case scenario, yaitu melakukan testing pada halaman terberat pada aplikasi, dalam hal ini digunakan pada tiga halaman, yaitu:

Pertama testing untuk halaman section ke-5 yang berisikan 6 tabel dengan 2 tabel formulasi, 4 query local, dan juga merupakan laman edit borang yang cukup memakan resource yang banyak. (url: /section/5/edit)

Yang kedua dilakukan pada laman json yang mengambil data dari server lain dan memerlukan beberapa 5 query local dan 2 query ke database remote nya . (url: /section/2/content/1/get)

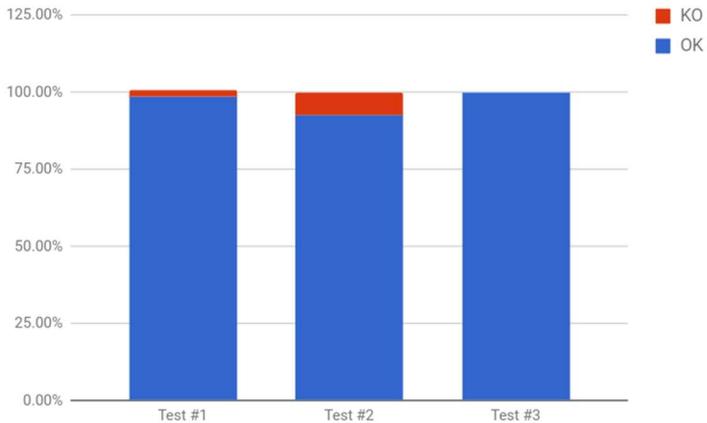
Yang ketiga dilakukan pada laman review yang memiliki 6 tabel dengan 2 tabel formulasi tidak hanya pada tabel, namun juga formulasi pada indikator, dan juga menggunakan 7 query local. (url: /section/5/content/1/review/edit)

Inisiasi testing dilakukan dengan command berikut “./jmeter -n -t ta.intip.in.jmx -l result -e -o output” seperti terlihat pada gambar 6.2.

```
harun@harun:~/apache-jmeter-3.2/bin$./jmeter -n -t ta.intip.in.jmx -l result -e
-o output
Creating summariser <summary>
Created the tree successfully using ta.intip.in.jmx
Starting the test @ Wed Jun 21 10:29:02 WIB 2017 (1498015742308)
```

**Gambar 6.2 Perintah eksekusi testing**

Dari testing yang dilakukan, berikut dijabarkan penjelasan hasilnya:



**Gambar 6.3** Persentase keberhasilan *request* dari *testing*

Pada testing pertama 98.08% request berhasil dikembalikan. Pada testing kedua 92.7% request berhasil dikembalikan. Sedangkan pada testing ketiga 100% request berhasil dieksekusi.

**Tabel 6.3** *Response time (ms)* dari masing-masing *testing*

| HTTP Request | Average | Min | Max    |
|--------------|---------|-----|--------|
| Test #1      | 4072.79 | 123 | 130456 |
| Test #2      | 7939.85 | 0   | 157603 |
| Test #3      | 3609.74 | 768 | 66054  |

Pada testing pertama, dari 5000 request yang dikirimkan, hanya sekitar 96 yang gagal kembali (1.92%) Dengan response time paling cepat 123 milisecond, dan response time terlama hingga 130.5 detik.

Pada testing kedua, dari 5000 request yang dikirimkan, sekitar 365 gagal dikembalikan (7.30%) Dengan response time paling cepat 0 milisecond, dan response time terlama hingga 157.6 detik.

Pada testing ketiga, dari 5000 request yang dikirimkan, sekitar 365 gagal dikembalikan (7.30%) Dengan response time paling cepat 768 milisecond, dan response time terlama hingga 66 detik.

### 6.1.3. User Acceptance Test

Dilakukan pengujian aplikasi oleh 5 orang yang mewakili seluruh role pada aplikasi ini, dengan 3 orang melakukan pengujian pada masing-masing role, dan sisanya masing-masing untuk role editor dan auditor.

Berikut penjelasan latar belakang dari penguji yang melakukan testing:

**Tabel 6.4 Latar belakang penguji aplikasi**

| No | Nama Penguji          | Latar Belakang                                | Role yg diuji                                      |
|----|-----------------------|-----------------------------------------------|----------------------------------------------------|
| 1  | Widjonarko<br>Roestam | Mantan Petinggi<br>Lembaga<br>Penjaminan Mutu | Semua role (admin,<br>creator, editor,<br>auditor) |

|   |                  |                                                 |                                              |
|---|------------------|-------------------------------------------------|----------------------------------------------|
| 2 | Murni R          | Ketua Lembaga Penjaminan Mutu                   | Semua role (admin, creator, editor, auditor) |
| 3 | Nisfu Asrul Sani | Mantan Penanggung Jawab Lembaga Penjaminan Mutu | Semua role (admin, creator, editor, auditor) |
| 4 | Staff PJM 1      | Staff Lembaga Penjaminan Mutu                   | Editor                                       |
| 5 | Staff PJM 2      | Staff Lembaga Penjaminan Mutu                   | Auditor                                      |

Pengujian menggunakan metode System Usability Scale (SUS) seperti yang dijelaskan pada bab 2.2.10. Digunakan daftar pertanyaan sebagai tertera pada table 6.5.

**Tabel 6.5 Daftar Pertanyaan yang diajukan**

| No | Pertanyaan                                                                               | Negasi |
|----|------------------------------------------------------------------------------------------|--------|
| 1  | Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin                   | -      |
| 2  | Saya berpikir sistem ini sangat rumit dan tidak perlu                                    | V      |
| 3  | Saya berpikir sistem ini mudah digunakan                                                 | -      |
| 4  | Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini | V      |
| 5  | Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik                 | -      |
| 6  | Saya berpikir terlalu banyak inkonsistensi dari sistem ini                               | V      |

|    |                                                                                        |   |
|----|----------------------------------------------------------------------------------------|---|
| 7  | Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat | - |
| 8  | Saya berpikir bahwa sistem ini sangat sulit untuk digunakan                            | V |
| 9  | Saya merasa sangat percaya diri dalam menggunakan sistem ini                           | - |
| 10 | Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini                     | V |

Pertanyaan yang bersifat negasi berarti nilai yang semakin rendah semakin baik, berbeda dengan pertanyaan lainnya yang semakin tinggi semakin baik, dengan range penilaian 1-5. Kemudian range diubah ke 0-4. Jawaban dari masing-masing pertanyaan tersebut kemudian dibuat di jumlah dan dirata-ratakan seperti terlihat pada gambar 6.8.

| Participant            | q1         | q2         | q3         | q4         | q5         | q6       | q7         | q8         | q9         | q10      | SUS Score   |
|------------------------|------------|------------|------------|------------|------------|----------|------------|------------|------------|----------|-------------|
| Widjonarko Roestam     | 3          | 4          | 4          | 0          | 3          | 3        | 3          | 3          | 3          | 1        | 67.5        |
| Murni R                | 3          | 3          | 3          | 3          | 3          | 4        | 4          | 4          | 3          | 4        | 85          |
| Nisfu Asrul Sani       | 2          | 3          | 3          | 3          | 4          | 3        | 3          | 3          | 3          | 2        | 72.5        |
| Staff PJM 1 (editor)   | 3          | 2          | 2          | 0          | 3          | 1        | 3          | 3          | 3          | 0        | 50          |
| Staff PJM 2 (auditor)  | 3          | 4          | 4          | 3          | 4          | 4        | 4          | 4          | 4          | 3        | 92.5        |
| <b>Total Usability</b> | <b>2.8</b> | <b>3.2</b> | <b>3.2</b> | <b>1.8</b> | <b>3.4</b> | <b>3</b> | <b>3.4</b> | <b>3.4</b> | <b>3.2</b> | <b>2</b> | <b>73.5</b> |

**Gambar 6.4 Hasil Perhitungan SUS**

Penjelasan hasil usability testing dapat dilihat pada tabel 6.4. Dengan kesimpulan diambil jika skor lebih atau sama dengan 2 maka dianggap setuju, dan jika dibawah 2 maka dianggap tidak setuju.

**Tabel 6.3 Pertanyaan setelah di negasi dan hasil usability test**

| No | Pertanyaan                                                             | Skor | Kesimpulan |
|----|------------------------------------------------------------------------|------|------------|
| 1  | Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin | 2.8  | Setuju     |

|    |                                                                                                |     |              |
|----|------------------------------------------------------------------------------------------------|-----|--------------|
| 2  | Saya berpikir sistem ini tidak rumit dan sangat diperlukan                                     | 3.2 | Setuju       |
| 3  | Saya berpikir sistem ini mudah digunakan                                                       | 3.2 | Setuju       |
| 4  | Saya berpikir saya tidak akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini | 1.8 | Tidak Setuju |
| 5  | Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik                       | 3.4 | Setuju       |
| 6  | Saya berpikir sangat sedikit inkonsistensi dari sistem ini                                     | 3   | Setuju       |
| 7  | Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat         | 3.4 | Setuju       |
| 8  | Saya berpikir bahwa sistem ini sangat mudah untuk digunakan                                    | 3.4 | Setuju       |
| 9  | Saya merasa sangat percaya diri dalam menggunakan sistem ini                                   | 3.2 | Setuju       |
| 10 | Saya tidak perlu belajar banyak hal sebelum dapat menggunakan sistem ini                       | 2   | Setuju       |

## 6.2. Pembahasan

Pada subbab ini akan dibahas dan disimpulkan hasil dari pengujian fungsional, non-fungsional dan user acceptance test dari aplikasi borang akreditasi.

Pada pengujian fungsional seluruh fitur yang diujikan berhasil dilakukan sesuai dengan ekspektasi (expected result) dan tidak ada error yang ditemukan.

Pada pengujian non-fungsional, hasil uji coba menggunakan aplikasi JMeter dapat disimpulkan laman terberat terdapat pada halaman “/section/2/content/get” yaitu pada pengujian kedua dengan 92.7% request yang dikirimkan berhasil dieksekusi dengan baik. Dari 5000 request yang dikirimkan, terdapat sekitar 365 request yang gagal kembali (7.30%). Dengan response time paling cepat 123 milisecond, dan response time terlama hingga 130.5 detik.

Pada user acceptance test menghasilkan rata-rata skor System Usability Scale (SUS) 73.5 dengan skor pertanyaan paling rendah pada pertanyaan ke-4 (1.8) yaitu pengguna membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini, dan skor pertanyaan paling tinggi pada pertanyaan ke-5, ke-7, dan ke-8 (3.4) yaitu seluruh fungsi yang telah terintegrasi dengan sangat baik; banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat; dan sistem ini sangat mudah untuk digunakan.

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Pada bab ini dijabarkan mengenai kesimpulan yang didapat dari hasil penelitian yang telah dilakukan. Kesimpulan ini diharapkan dapat menjawab tujuan yang telah ditetapkan diawal penelitian. Saran diberikan untuk digunakan pada penelitian selanjutnya.

#### **7.1. KESIMPULAN**

Dari pelaksanaan penelitian tugas akhir ini, dapat diambil kesimpulan sebagai berikut.

1. Pembuatan aplikasi borang akreditasi menggunakan role-based system berhasil dibuat dalam tugas akhir ini.
2. Aplikasi borang akreditasi memiliki tiga role, dari sebelumnya empat role. Yaitu creator, editor, dan auditor. Role owner tidak diimplementasi karena fungsinya yang merupakan bagian dari role creator.
3. Aplikasi borang akreditasi dapat digunakan langsung oleh creator tanpa perlu mengubah source code aplikasi seperti pada aplikasi sejenis sebelumnya, SPMI-ITS.
4. Aplikasi borang akreditasi dapat langsung terhubung dengan database lain, misalnya database kemahasiswaan maupun database kepegawaian tanpa perlu mengubah source code aplikasi.
5. Aplikasi borang dapat melakukan formulasi pada table dan indikator sesuai dengan formula yang ditentukan oleh role creator, dan dapat secara otomatis terformulasi untuk role editor pada bagian table, dan juga untuk role auditor pada bagian indikator. Sehingga tugas editor dan auditor menjadi lebih sederhana yaitu hanya mengisi sesuai format yang ada dan juga menilai sesuai indikator yang ditentukan menggunakan dropdown list.
6. Karena desain struktur borang yang dinamis, tugas akhir ini dapat diaplikasikan untuk unit-unit lain selain

Program Studi Sarjana, seperti Program Studi Doktor; Program Studi Magister; bahkan hingga unit pengelola (UPT).

## **7.2. SARAN**

Tugas akhir ini memiliki potensi yang besar untuk dikembangkan kedepannya agar lebih baik dan lebih bermanfaat, untuk itu diberikan beberapa saran yang dapat dipertimbangkan yaitu:

1. Agar aplikasi dapat mendukung untuk tipe form lain, seperti format borang ASEAN University Network (AUN) hanya diperlukan perubahan pada penilaian, yaitu mengubah sistem bobot per indikator dengan sistem lain sesuai format ketentuannya.
2. Penilaian hasil evaluasi tidak diimplementasikan pada tugas akhir ini, jadi untuk penelitian selanjutnya bisa melengkapi dengan menambahkan score penilaian berdasarkan bobot dari masing-masing indikator.
3. Untuk penelitian selanjutnya juga agar dapat mengembangkan dashboard untuk role baru, misalnya tim pemantau untuk melihat skor penilaian di masing-masing grup (jurusan), dan evaluasi / rekomendasi untuk bagian mana yang perlu ditingkatkan agar nilai grup tersebut dapat meningkat kedepannya.

## DAFTAR PUSTAKA

- [1] P. d. P. K. I. I. Lembaga Penjaminan Mutu, BORANG - Sistem Penjaminan Mutu Internal, Surabaya: Institut Teknologi Sepuluh Nopember, 2016.
- [2] D. P. Nasional, Akreditasi Program Studi Sarjana, Jakarta: Badan Akreditasi Nasional Perguruan Tinggi, 2008.
- [3] D. F. F. a. D. R. Kuhn, "Role-Based Access Controls," *15th National Computer Security Conference (1992), Baltimore MD*, vol. Technology Administration, pp. 554 - 563, 1992.
- [4] R. Awaludin, Menyelami Framework Laravel, Rahmat Awaludin, 2014.
- [5] Mike, "What's up with SQL Server 2008 Express editions," 7 August 2008.
- [6] D. A. Cockburn, "Using Both Incremental and Iterative Development," *Humans and Technology*, 2008.
- [7] "Pengenalan JSON," [Online]. Available: <http://www.json.org/json-id.html>. [Accessed 20 June 2017].
- [8] A. S. Foundation, "Apache Jmeter," Apache Software Foundation, [Online]. Available: <http://jmeter.apache.org/>. [Accessed 21 Juni 2017].
- [9] P. Sevcik, "Network Forecast Sevcik - Defining The Application Performance Index," [Online]. Available:

- [http://www.apdex.org/docs/Defining\\_The\\_Application\\_Performance\\_Index.pdf](http://www.apdex.org/docs/Defining_The_Application_Performance_Index.pdf). [Accessed 19 Desember 2019].
- [10] Usability.gov, "System Usability Scale (SUS)," [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Accessed 19 Desember 2016].
- [11] J. Sauro, "Measuring Usability With The System Usability Scale (SUS)," MeasuringU, 2 Februari 2011. [Online]. Available: <http://www.measuringu.com/sus.php>. [Accessed 19 Desember 2016].
- [12] Adriana Martin, "RUP Practices That Could Help Scrum" Scrum Expert, [Online]. Available: <https://intip.in/wUfr>. [Accessed 4 July 2017].

## BIODATA PENULIS



Penulis bernama lengkap Harun Rizal lahir di Payakumbuh pada tanggal 16 Mei 1995. Merupakan anak ketiga dari 4 bersaudara dan telah menempuh pendidikan formal yaitu; SD Muhammadiyah 08 Plus Jakarta Timur, SMPI Raudhatul Jannah Payakumbuh, dan SMA Pembangunan Bukittinggi.

Pada tahun 2013 melanjutkan pendidikan di Jurusan Sistem Informasi FTIF - Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 5213100125.

Selama menjadi mahasiswa penulis telah mengikuti berbagai kegiatan kemahasiswaan, baik berupa kepanitiaan ditingkat Institut maupun ditingkat Nasional, salah satunya adalah dalam acara big event milik ITS yaitu ITS EXPO dan juga acara Information System Expo (ISE) pada tahun 2015 sebagai staff ahli multimedia.

Pada tahun keempat karena penulis tertarik dengan bidang diseminasi data, sehingga mengambil bidang minat Laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI).

*[halaman ini sengaja dikosongkan]*

## LAMPIRAN 1

### A.1 Pertanyaan Kuesioner

Berikut ini ialah contoh pertanyaan kuesioner yang akan diberikan kepada user.

#### SYSTEM USABILITY SCALE FORM

Instruksi : Isikan setiap jawaban pernyataan dengan memberikan tanda checklist (√) atau cross (x) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.

1 = Sangat tidak setuju

5 = Sangat setuju

| Pernyataan                                                             | 1 | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------|---|---|---|---|---|
| Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin |   |   |   |   |   |
| Saya berpikir sistem ini sangat rumit dan tidak perlu                  |   |   |   |   |   |
| Saya berpikir sistem ini mudah digunakan                               |   |   |   |   |   |
| Saya berpikir saya akan membutuhkan bantuan                            |   |   |   |   |   |

|                                                                                        |  |  |  |  |  |
|----------------------------------------------------------------------------------------|--|--|--|--|--|
| seseorang untuk dapat menggunakan sistem ini                                           |  |  |  |  |  |
| Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik               |  |  |  |  |  |
| Saya berpikir terlalu banyak inkonsistensi dari sistem ini                             |  |  |  |  |  |
| Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat |  |  |  |  |  |
| Saya berpikir bahwa sistem ini sangat sulit untuk digunakan                            |  |  |  |  |  |
| Saya merasa sangat percaya diri dalam menggunakan sistem ini                           |  |  |  |  |  |
| Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini                     |  |  |  |  |  |

## Bagian II Evaluasi

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? Sebutkan
2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan
3. Saran anda untuk perbaikan ke depannya?

## A.2 Hasil Kuesioner

Berikut ini ialah hasil kuesioner yang telah diisi oleh user.

**SYSTEM USABILITY SCALE FORM**

Tanggal: Jumat, 16 Juni 2017  
 Nama Lengkap: MURANI, P  
 Role: Creator  
 Tanda Tangan: MP

**Instruksi** : Isikan setiap jawaban pernyataan dengan memberikan tanda checklist (✓) atau cross (x) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.  
 1 = Sangat tidak setuju  
 5 = Sangat setuju

| Pernyataan                                                                               | 1 | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------------|---|---|---|---|---|
| Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin                   |   |   |   | ✓ |   |
| Saya berpikir sistem ini sangat rumit dan tidak perlu                                    |   | ✓ |   |   |   |
| Saya berpikir sistem ini mudah digunakan                                                 |   |   |   | ✓ |   |
| Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini |   | ✓ |   |   |   |
| Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik                 |   |   |   | ✓ |   |
| Saya berpikir terlalu banyak inkonsistensi dari sistem ini                               | ✓ |   |   |   |   |
| Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat   |   |   |   |   | ✓ |
| Saya berpikir bahwa sistem ini sangat sulit untuk digunakan                              | ✓ |   |   |   |   |
| Saya merasa sangat percaya diri dalam menggunakan sistem ini                             |   |   |   | ✓ |   |
| Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini                       | ✓ |   |   |   |   |

**Bagian II Evaluasi**

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? Sebutkan

tidak dengan isian sama.

2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan

- tombol upload agar bisa menyimpan bukti fisik yg berupa softfile.

3. Saran anda untuk perbaikan ke depannya?

- tampilan diperantik supaya tdk membosankan  
 - ditambah buku pedoman pengisian

## SYSTEM USABILITY SCALE FORM

Tanggal: 13 Juni 2017

Nama Lengkap: Widjanarko Roeston

Role: Auditor, Editor, Creator

Tanda Tangan:



**Instruksi** : Isikan setiap jawaban pernyataan dengan memberikan tanda checklist (√) atau cross (X) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.

1 = Sangat tidak setuju

5 = Sangat setuju

| Pernyataan                                                                               | 1 | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------------|---|---|---|---|---|
| Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin                   |   |   |   | √ |   |
| Saya berpikir sistem ini sangat rumit dan tidak perlu                                    | √ |   |   |   |   |
| Saya berpikir sistem ini mudah digunakan                                                 |   |   |   |   | √ |
| Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini |   |   |   | √ |   |
| Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik                 |   |   |   | √ |   |
| Saya berpikir terlalu banyak inkonsistensi dari sistem ini                               |   | √ |   |   |   |
| Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat   |   |   |   | √ |   |
| Saya berpikir bahwa sistem ini sangat sulit untuk digunakan                              |   | √ |   |   |   |
| Saya merasa sangat percaya diri dalam menggunakan sistem ini                             |   |   |   | √ |   |
| Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini                       |   |   |   | √ |   |

**Bagian II Evaluasi**

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? Sebutkan

Tidak -

2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan

Ada fungsi self evaluator bagi editor untuk mengetahui total score

3. Saran anda untuk perbaikan ke depannya?

Help menu yg lebih<sup>2</sup> pesan

## SYSTEM USABILITY SCALE FORM

Tanggal: 9 Juni 2017

Nama Lengkap: Nisa Azzahra Sani

Role: Creator

Tanda Tangan: 

**Instruksi** : Isikan setiap jawaban pernyataan dengan memberikan tanda checklist (√) atau cross (x) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.

1 = Sangat tidak setuju

5 = Sangat setuju

| Pernyataan                                                                               | 1 | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------------|---|---|---|---|---|
| Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin                   |   |   | √ |   |   |
| Saya berpikir sistem ini sangat rumit dan tidak perlu                                    |   | √ |   |   |   |
| Saya berpikir sistem ini mudah digunakan                                                 |   |   |   | √ |   |
| Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini |   | √ |   |   |   |
| Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik                 |   |   |   |   | √ |
| Saya berpikir terlalu banyak inkonsistensi dari sistem ini                               |   | √ |   |   |   |
| Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat   |   |   |   | √ |   |
| Saya berpikir bahwa sistem ini sangat sulit untuk digunakan                              |   | √ |   |   |   |
| Saya merasa sangat percaya diri dalam menggunakan sistem ini                             |   |   |   | √ |   |
| Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini                       |   |   | √ |   |   |

**Bagian II Evaluasi**

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? Sebutkan

kejelasan dalam memotai / keterangan

2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan

3. Saran anda untuk perbaikan ke depannya?

## LAMPIRAN B

### B.1 Skema database

Berikut ini ialah gambar skema database yang digunakan dalam kegiatan operasional aplikasi.

