



**IMPLEMENTASI TEKNOLOGI RESTFUL WEB SERVICE DALAM
PENGEMBANGAN SISTEM INFORMASI PEREKAMAN
PRESTASI MAHASISWA BERBASIS WEBSITE**

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Wiku Galindra Wardhana

NIM: 165150701111020



**PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2020

PENGESAHAN

IMPLEMENTASI TEKNOLOGI *RESTFUL WEB SERVICE* DALAM PENGEMBANGAN
SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA BERBASIS *WEBSITE*
(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Wiku Galindra Wardhana
NIM: 165150701111020

Skripsi ini telah diuji dan dinyatakan lulus pada
17 April 2020
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Issa Arwani, S.Kom., M.Sc.
NIP. 19830922 201212 1 003

Dosen Pembimbing II



Bayu Rahayudi, S.T., M.T.
NIP. 19740712 200604 1 001

Mengetahui

Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.

NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 April 2020



Wiku Galindra Wardhana

NIM: 165150701111020



ABSTRAK

Wiku Galindra Wardhana, Implementasi Teknologi *Restful Web Service* Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis *Website* (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Pembimbing: Issa Arwani, S.Kom., M.Sc. dan Bayu Rahayudi, S.T., M.T.

Selain dituntut ahli di bidang akademik, mahasiswa juga dituntut untuk memiliki prestasi. Fakultas Teknologi Pertanian Universitas Brawijaya berupaya memberikan pengakuan terhadap prestasi yang diperoleh mahasiswanya dengan melakukan perekaman prestasi mahasiswa. Namun, perekaman prestasi tersebut belum bisa dilakukan secara maksimal karena masih dilakukan secara manual. Oleh karena itu, diperlukan sebuah sistem informasi yang dapat melakukan perekaman prestasi mahasiswa pada Fakultas Teknologi Pertanian dengan tujuan untuk mempermudah pengumpulan data prestasi mahasiswa. Sistem ini dikembangkan dalam bentuk *website* dengan teknologi *RESTFUL web service*. Pengembangan sistem dilakukan dengan metode *prototyping* yang dimulai dari analisa kebutuhan. Selanjutnya, dibuat *prototype* sistem dan dilakukan evaluasi oleh ketua PSIK Fakultas Teknologi Pertanian. Dari hasil evaluasi dilakukan perbaikan dan dilakukan presentasi *prototype* ulang hingga *prototype* disetujui. Hasil dari analisa kebutuhan digambarkan ke dalam *use case diagram*, *use case scenario*, dan *activity diagram*. Perancangan sistem digambarkan ke dalam *sequence diagram*, *class diagram*, dan *physical data model*. Implementasi *website* menggunakan *framework* Laravel dan implementasi REST API menggunakan *framework* Lumen. Kemudian dilakukan pengujian terhadap sistem menggunakan *blackbox testing* dengan presentase hasil pengujian sebesar 100% yang menandakan sistem telah sesuai dengan spesifikasi. Selain itu, dilakukan pengujian *usability* menggunakan *System Usability Scale* dengan nilai 78. Berdasarkan nilai yang didapat, sistem masuk ke dalam kategori baik dengan skala nilai C dan rentang penerimaan berada pada kategori dapat diterima.

Kata kunci: prestasi mahasiswa, metode *prototyping*, *RESTFUL web service*, *laravel*, *lumen*, *system usability scale* (SUS)



ABSTRACT

Wiku Galindra Wardhana, *Implementation of Restful Web Service Technology in Development of Website Based Student Achievement Recording Information System (Case Study: Faculty of Agricultural Technology Brawijaya University)*

Supervisors: Issa Arwani, S.Kom., M.Sc. and Bayu Rahayudi, S.T., M.T.

Besides being demanded by experts in the academic field, students are also required to have achievements. The Faculty of Agricultural Technology Brawijaya University seeks to to give recognition of the achievements of its students by recording student achievements. However, the recording of these achievements can not be done optimally because it is still done manually. Therefore, an information system that can record student achievement at the Faculty of Agricultural Technology is needed with the aim to facilitate the collection of student achievement data. This system was developed on a website platform with RESTFUL web service technology. System development is done by prototyping method that starts from the requirement analysis. Next, a prototype system was made and an evaluated by the head of the information system planning and public relation Faculty of Agriculture Technology. From the evaluation results were carried out repairs and re-prototype presentations until the prototype was approved. The results of the requirement analysis are illustrated by use case diagram, use case scenario, and activity diagram. The system design is illustrated by sequence diagram, class diagram, and physical data model. The website implementation uses the Laravel framework and the REST API implementation uses the Lumen framework. Then the system is tested using blackbox testing with a percentage of test results is 100% which indicates the system complies with specifications. In addition, usability testing was done using the System Usability Scale with a value of 78. Based on the values obtained, the system categorized on Good category with C on value scale and the acceptability range is on Acceptable category.

Keywords: student achievement, prototyping methods, RESTFUL Web service, laravel, lumen, system ssability scale (SUS)



DAFTAR ISI

IMPLEMENTASI TEKNOLOGI <i>RESTFUL WEB SERVICE</i> DALAM PENGEMBANGAN SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA BERBASIS <i>WEBSITE</i>	i
PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR <i>SOURCE CODE</i>	xvi
DAFTAR LAMPIRAN.....	1
BAB 1 PENDAHULUAN.....	2
1.1 Latar Belakang.....	2
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat.....	4
1.4.1 Bagi Peneliti.....	4
1.4.2 Bagi Fakultas Teknologi Pertanian Universitas Brawijaya.....	4
1.4.3 Bagi Mahasiswa.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN PUSTAKA.....	6
2.1 Fakultas Teknologi Pertanian Universitas Brawijaya.....	6
2.2 Prestasi Mahasiswa.....	6
2.3 Sistem Informasi.....	7
2.4 Teknologi Pendukung.....	7
2.5 <i>Web Service</i>	8
2.6 <i>Representational State Transfer (REST)</i>	8
2.6.1 <i>REST Constraint</i>	9



DAFTAR TABEL

Tabel 2.1 HTTP verb dan aksi yang dilakukan.....	10
Tabel 2.2 Kode Status HTTP.....	10
Tabel 2.3 Perbandingan antara XML-RPC/SOAP dan REST.....	11
Tabel 2.4 Komponen <i>use case diagram</i>	14
Tabel 2.5 Komponen <i>activity diagram</i>	15
Tabel 2.6 Komponen <i>sequence diagram</i>	16
Tabel 2.7 Komponen <i>class diagram</i>	17
Tabel 2.8 Daftar Pernyataan SUS.....	19
Tabel 4.1 Identifikasi aktor.....	24
Tabel 4.2 Kebutuhan fungsional sistem informasi perekaman prestasi mahasiswa.....	25
Tabel 4.3 Kebutuhan non fungsional sistem informasi perekaman prestasi mahasiswa.....	26
Tabel 4.4 Use case scenario <i>login</i>	33
Tabel 4.5 <i>Use case scenario</i> mahasiswa tambah prestasi.....	34
Tabel 4.6 <i>Use case scenario</i> mahasiswa lihat prestasi.....	35
Tabel 4.7 <i>Use case scenario</i> mahasiswa hapus prestasi.....	35
Tabel 4.8 <i>Use case scenario</i> mahasiswa unduh <i>draft</i>	35
Tabel 4.9 <i>Use case scenario</i> kemahasiswaan tambah kegiatan.....	36
Tabel 4.10 <i>Use case scenario</i> verifikasi.....	37
Tabel 4.11 <i>Use case scenario</i> lihat daftar prestasi.....	37
Tabel 4.12 <i>Use case scenario</i> unduh daftar prestasi.....	38
Tabel 4.13 <i>Use case scenario</i> <i>logout</i>	38
Tabel 5.1 Perancangan fitur <i>register</i> API Prestasi.....	58
Tabel 5.2 Perancangan fitur <i>auth</i> API Prestasi.....	59
Tabel 5.3 Perancangan fitur tambah prestasi API Prestasi.....	59
Tabel 5.4 Perancangan fitur lihat prestasi API Prestasi.....	60
Tabel 5.5 Perancangan fitur hapus prestasi API Prestasi.....	60
Tabel 5.6 Perancangan fitur unduh <i>draft</i> API Prestasi.....	61
Tabel 5.7 Perancangan fitur tambah kegiatan API Prestasi.....	62



Tabel 5.8 Perancangan fitur verifikasi prestasi API Prestasi.....	62
Tabel 5.9 Perancangan fitur tolak prestasi API Prestasi	63
Tabel 5.10 Perancangan fitur lihat daftar prestasi API Prestasi	63
Tabel 5.11 Perancangan fitur unduh daftar prestasi API Prestasi	64
Tabel 7.1 Penjelasan pengujian fungsi register API prestasi	84
Tabel 7.2 Penjelasan pengujian fungsi auth API prestasi	85
Tabel 7.3 Penjelasan pengujian fungsi tambah prestasi API prestasi.....	86
Tabel 7.4 Penjelasan pengujian fungsi lihat prestasi mahasiswa API prestasi	87
Tabel 7.5 Penjelasan pengujian fungsi hapus prestasi API prestasi	88
Tabel 7.6 Penjelasan pengujian fungsi unduh <i>draft</i> API prestasi	89
Tabel 7.7 Penjelasan pengujian fungsi tambah kegiatan API prestasi.....	90
Tabel 7.8 Penjelasan pengujian fungsi verifikasi prestasi API prestasi.....	91
Tabel 7.9 Penjelasan pengujian fungsi tolak prestasi API prestasi	92
Tabel 7.10 Penjelasan pengujian fungsi lihat prestasi verifikasi API prestasi.....	93
Tabel 7.11 Penjelasan pengujian unduh daftar prestasi verifikasi API prestasi ...	94
Tabel 7.12 Pengujian <i>login</i>	96
Tabel 7.13 Pengujian mahasiswa tambah prestasi.....	96
Tabel 7.14 Pengujian mahasiswa lihat prestasi	97
Tabel 7.15 Pengujian mahasiswa hapus prestasi.....	97
Tabel 7.16 Pengujian mahasiswa unduh <i>draft</i>	97
Tabel 7.17 Pengujian kemahasiswaan tambah kegiatan.....	98
Tabel 7.18 Pengujian verifikasi (verifikasi).....	98
Tabel 7.19 Pengujian verifikasi (tolak).....	99
Tabel 7.20 Pengujian lihat daftar prestasi	99
Tabel 7.21 Pengujian unduh daftar prestasi.....	99
Tabel 7.22 Pengujian <i>logout</i>	100
Tabel 7.23 Hasil Pengujian <i>System Usability Scale (SUS)</i>	101
Tabel 7.24 Hasil perhitungan <i>System usability scale</i>	101



DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Fakultas Teknologi Pertanian Universitas Brawijaya	6
Gambar 2.2 contoh <i>Unique Resource Identifier</i>	10
Gambar 2.3 Contoh format JSON	13
Gambar 2.4 Alur pengembangan sistem informasi dengan metode <i>prototyping</i>	13
Gambar 2.5 Rentang nilai SUS	20
Gambar 3.1 Metodologi Penelitian	21
Gambar 4.1 Aturan penomoran sistem informasi perekaman prestasi mahasiswa	25
Gambar 4.2 Rancangan halaman login karyawan	27
Gambar 4.3 Rancangan halaman login mahasiswa	27
Gambar 4.4 Rancangan halaman mahasiswa	28
Gambar 4.5 Rancangan form tambah prestasi	28
Gambar 4.6 Rancangan halaman kemahasiswaan	29
Gambar 4.7 Rancangan form tambah kegiatan	30
Gambar 4.8 Rancangan halaman akademik	30
Gambar 4.9 Rancangan pilihan tingkat kejuaraan	31
Gambar 4.10 Rancangan pilihan posisi/jabatan	31
Gambar 4.11 Rancangan pilihan jenis prestasi	32
Gambar 4.12 <i>Use case diagram</i> sistem informasi perekaman prestasi mahasiswa	33
Gambar 4.13 <i>Activity diagram</i> login mahasiswa	39
Gambar 4.14 <i>Activity diagram</i> login karyawan	39
Gambar 4.15 <i>Activity diagram</i> mahasiswa tambah prestasi	40
Gambar 4.16 <i>Activity diagram</i> mahasiswa lihat prestasi	41
Gambar 4.17 <i>Activity diagram</i> mahasiswa hapus prestasi	41
Gambar 4.18 <i>Activity diagram</i> mahasiswa unduh <i>draft</i>	42
Gambar 4.19 <i>Activity diagram</i> kemahasiswaan tambah kegiatan	42
Gambar 4.20 <i>Activity diagram</i> verifikasi	43
Gambar 4.21 <i>Activity diagram</i> lihat daftar prestasi	43
Gambar 4.22 <i>Activity diagram</i> unduh daftar prestasi	44



Gambar 4.23 Activity diagram logout.....	44
Gambar 5.1 Arsitektur sistem informasi perekaman prestasi mahasiswa.....	45
Gambar 5.2 Sequence diagram login mahasiswa.....	46
Gambar 5.3 Sequence diagram login mahasiswa pada API prestasi.....	46
Gambar 5.4 Sequence diagram login karyawan.....	47
Gambar 5.5 Sequence diagram login karyawan pada API prestasi.....	47
Gambar 5.6 Sequence diagram mahasiswa tambah prestasi.....	48
Gambar 5.7 Sequence diagram mahasiswa tambah prestasi pada API prestasi	48
Gambar 5.8 Sequence diagram mahasiswa lihat prestasi.....	49
Gambar 5.9 Sequence diagram mahasiswa lihat prestasi pada API prestasi.....	49
Gambar 5.10 Sequence diagram mahasiswa hapus prestasi.....	50
Gambar 5.11 Sequence diagram mahasiswa hapus prestasi pada API prestasi	50
Gambar 5.12 Sequence diagram mahasiswa unduh draft.....	51
Gambar 5.13 Sequence diagram mahasiswa unduh draft pada API prestasi.....	51
Gambar 5.14 Sequence diagram kemahasiswaan tambah kegiatan.....	52
Gambar 5.15 Sequence diagram kemahasiswaan tambah kegiatan pada API prestasi.....	52
Gambar 5.16 Sequence diagram verifikasi.....	53
Gambar 5.17 Sequence diagram verifikasi bagian verifikasi pada API prestasi.....	53
Gambar 5.18 Sequence diagram verifikasi bagian tolak pada API prestasi.....	54
Gambar 5.19 Sequence diagram lihat daftar prestasi.....	54
Gambar 5.20 Sequence diagram lihat daftar prestasi pada API prestasi.....	54
Gambar 5.21 Sequence diagram unduh daftar prestasi.....	55
Gambar 5.22 Sequence diagram unduh daftar prestasi pada API prestasi.....	55
Gambar 5.23 Sequence diagram logout.....	56
Gambar 5.24 Class diagram website Sistem Informasi Perekaman Prestasi Mahasiswa.....	56
Gambar 5.25 Class Diagram API prestasi.....	57
Gambar 5.26 Physical data model sistem informasi perekaman prestasi mahasiswa.....	58
Gambar 6.1 Implementasi UI fitur login Mahasiswa.....	65
Gambar 6.2 Implementasi UI fitur Login Karyawan.....	68
Gambar 6.3 Implementasi UI fitur mahasiswa tambah prestasi.....	70



Gambar 6.4 Implementasi UI fitur mahasiswa lihat prestasi.....	73
Gambar 6.5 Implementasi UI fitur mahasiswa hapus prestasi.....	74
Gambar 6.6 Implementasi UI fitur mahasiswa unduh <i>draft</i>	75
Gambar 6.7 Implementasi UI fitur kemahasiswaan tambah kegiatan.....	76
Gambar 6.8 Implementasi UI fitur verifikasi (bagian verifikasi).....	78
Gambar 6.9 Implementasi UI fitur verifikasi (bagian tolak).....	79
Gambar 6.10 Implementasi UI fitur lihat daftar prestasi.....	80
Gambar 6.11 Implementasi UI fitur unduh daftar prestasi.....	81
Gambar 6.12 Implementasi UI fitur <i>logout</i>	82
Gambar 7.1 Pengujian fungsi <i>register</i> API prestasi.....	84
Gambar 7.2 pengujian fungsi <i>auth</i> API prestasi.....	85
Gambar 7.3 pengujian fungsi tambah prestasi API prestasi.....	86
Gambar 7.4 pengujian fungsi lihat prestasi mahasiswa API prestasi.....	87
Gambar 7.5 Pengujian fungsi hapus prestasi API prestasi.....	88
Gambar 7.6 Pengujian fungsi unduh <i>draft</i> API prestasi.....	89
Gambar 7.7 pengujian fungsi tambah kegiatan API prestasi.....	90
Gambar 7.8 Pengujian fungsi verifikasi prestasi API prestasi.....	91
Gambar 7.9 Pengujian fungsi tolak prestasi API prestasi.....	92
Gambar 7.10 pengujian fungsi lihat prestasi verifikasi API prestasi.....	92
Gambar 7.11 pengujian fungsi unduh daftar prestasi API prestasi.....	94



DAFTAR SOURCE CODE

Source Code 6.1 Implementasi base url pada website.....	65
Source Code 6.2 Implementasi kode fitur login Mahasiswa.....	67
Source Code 6.3 Implementasi kode fitur login mahasiswa pada API prestassi... 68	68
Source Code 6.4 Implementasi kode fitur login Karyawan.....	69
Source Code 6.5 Implementasi kode fitur login pada API prestassi.....	70
Source Code 6.6 Implementasi kode fitur mahasiswa tambah prestasi.....	71
Source Code 6.7 Implementasi kode fitur mahasiswa tambah prestasi pada API prestassi.....	72
Source Code 6.8 Implementasi kode fitur mahasiswa lihat prestasi.....	73
Source Code 6.9 Implementasi kode fitur mahasiswa lihat prestasi pada API prestassi.....	73
Source Code 6.10 Implementasi kode fitur mahasiswa hapus prestasi.....	74
Source Code 6.11 Implementasi kode fitur mahasiswa hapus prestasi pada API prestassi.....	74
Source Code 6.12 Implementasi kode fitur mahasiswa unduh draft.....	75
Source Code 6.13 Implementasi kode fitur mahasiswa unduh draft pada API prestassi.....	75
Source Code 6.14 Implementasi kode fitur kemahasiswaan tambah kegiatan.....	77
Source Code 6.15 Implementasi kode fitur kemahasiswaan tambah kegiatan pada API prestassi.....	78
Source Code 6.16 Implementasi kode fitur verifikasi (bagian verifikasi).....	79
Source Code 6.17 Implementasi kode fitur verifikasi (bagian verifikasi) pada API prestassi.....	79
Source Code 6.18 Implementasi kode fitur verifikasi (bagian tolak).....	80
Source Code 6.19 Implementasi kode fitur verifikasi (bagian tolak) pada API prestassi.....	80
Source Code 6.20 Implementasi kode fitur lihat daftar prestasi.....	81
Source Code 6.21 Implementasi kode fitur lihat daftar prestasi pada API prestassi.....	81
Source Code 6.22 Implementasi kode fitur unduh daftar prestasi.....	82
Source Code 6.23 Implementasi kode fitur unduh daftar prestasi pada API prestassi.....	82



DAFTAR LAMPIRAN

Lampiran 1 Dokumentasi wawancara analisis kebutuhan	107
Lampiran 2 Dokumentasi wawancara presentasi <i>prototype</i> pertama	110
Lampiran 3 Dokumentasi wawancara presentasi <i>prototype</i> kedua	113
Lampiran 4 Kuisisioner SUS responden 1	114
Lampiran 5 Kuisisioner SUS responden 2	115
Lampiran 6 Kuisisioner SUS responden 3	116
Lampiran 7 Kuisisioner SUS responden 4	117
Lampiran 8 Kuisisioner SUS responden 5	118
Lampiran 9 Kuisisioner SUS responden 6	119
Lampiran 10 Kuisisioner SUS responden 7	120
Lampiran 11 Kuisisioner SUS responden 8	121



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada Kamus Besar Bahasa Indonesia (KBBI), prestasi merupakan sebuah hasil yang didapat melalui sesuatu yang diusahakan atau dikerjakan baik dari hasil belajar, bekerja, maupun keterampilan pada suatu bidang. Dalam dunia perkuliahan, selain dituntut ahli dalam masalah akademik, mahasiswa juga dituntut untuk menjadi orang yang berprestasi. Prestasi yang didapat pada mahasiswa akan menjadi kualitas dari orang tersebut serta akan menjadi nilai tambah ketika nantinya berkecimpung di dunia kerja. Dalam mempersiapkan mahasiswa, tidak cukup hanya dengan *hard skill* berupa bekal secara akademik melainkan juga dengan menumbuhkan *soft skill* dari mahasiswa yang didapat dengan berbagai kegiatan kemahasiswaan. Pada kehidupan di kampus juga banyak terdapat berbagai jenis kegiatan yang dapat dijadikan sebagai media pengembangan diri bagi mahasiswa seperti peningkatan penalaran, minat dan bakat, perluasan wawasan bagi mahasiswa, serta kesatuan dan persatuan. Pengembangan *soft skill* bagi mahasiswa tersebut harus mendapatkan dukungan dari pihak perguruan tinggi khususnya bidang kemahasiswaan berupa kelembagaan di bidang kemahasiswaan dan dorongan kepada mahasiswa untuk mendapat prestasi kejuaraan (Ristekdikti, 2019).

Mengetahui pentingnya hal tersebut, Fakultas Teknologi Pertanian Universitas Brawijaya berupaya memberikan penghargaan berupa pengakuan terhadap prestasi – prestasi yang telah didapat oleh mahasiswa Fakultas Teknologi Pertanian dengan cara melakukan perekaman prestasi – prestasi mahasiswa Fakultas Teknologi Pertanian agar nantinya bisa dijadikan sebagai *curriculum vitae* yang valid dalam bentuk SKPI (Surat Keterangan Pendamping Ijazah). Berdasarkan wawancara dengan bapak Mas'ud Effendi, STP., MP. Selaku ketua PSIK Fakultas Teknologi Pertanian, pada pelaksanaan perekaman prestasi mahasiswa di Fakultas Teknologi Pertanian tidak bisa dilakukan secara maksimal dikarenakan beberapa kendala mulai dari beberapa prestasi yang didapat mahasiswa Fakultas Teknologi Pertanian yang belum terekam oleh pihak pimpinan Fakultas Teknologi Pertanian, mahasiswa Fakultas Teknologi Pertanian merasa kesulitan untuk melakukan pendataan prestasi yang telah didapat, hingga pihak akademik merasa kesulitan untuk melakukan rekap data prestasi yang telah terekam. Berdasarkan permasalahan yang ada, ketua PSIK Fakultas Teknologi Pertanian mengatakan bahwa pihak Fakultas Teknologi Pertanian merencanakan untuk merancang sistem informasi perekaman prestasi mahasiswa dalam bentuk *website* dengan tujuan mempermudah dalam pengumpulan data prestasi mahasiswa Fakultas Teknologi Pertanian, dan dengan adanya sistem ini maka akan mendatangkan manfaat dari berbagai *stakeholder*.

Dari permasalahan yang ada, maka diperlukan sebuah sistem informasi yang dapat melakukan perekaman prestasi mahasiswa Fakultas Teknologi Pertanian. Sistem ini akan dikembangkan dengan menggunakan teknologi *web service*. Hal



ini dikarenakan pihak Fakultas Teknologi Pertanian berencana agar sistem dapat dijalankan sebagai aplikasi *mobile*. Dari pihak Fakultas Teknologi Pertanian juga berharap agar nantinya pihak perusahaan luar juga dapat mengakses data riwayat prestasi mahasiswa Fakultas Teknologi Pertanian sebagai *cross check* ketika ada pelamar dari Fakultas Teknologi Pertanian.

Web service adalah teknologi yang membuat internet agar bisa saling melakukan komunikasi dengan pola *program-to-program* (Rohman, Soebroto, & Kharisma 2018). Tujuan dari penggunaan *web service* adalah *client* dari berbagai macam platform berupa *dekstop*, *website*, dan *mobile apps* dapat mengakses layanan pada *web service* (Rohman, Soebroto, & Kharisma 2018). REST merupakan arsitektur *web service* yang bersifat *client server* dimana *client* melakukan *request* kepada *server* dan *server* memproses *request* dan mengembalikan *response*. Restful *web service* merupakan sebutan untuk aplikasi web yang menggunakan arsitektur dari REST. Dalam menggunakan teknologi REST dibantu dengan sebuah *tools* yang bernama API (*Application Programming Interface*) berbasis *website*. API secara umum terdiri dari 2 bagian, yaitu *server* sebagai penyedia data dan *client* yang dapat melakukan request data. REST API adalah API berbasis *website* yang menggunakan teknologi REST dan menggunakan format JSON (*JavaScript Object Notation*), yaitu sebuah format pertukaran data yang bisa digunakan baik pada *front-end* maupun *back-end* dari aplikasi *website* maupun sebuah *service* (Cooksey B., 2014).

1.2 Rumusan Masalah

1. Bagaimana hasil dari analisis kebutuhan Sistem Informasi Perekaman Prestasi Mahasiswa sesuai dengan kebutuhan pengguna?
2. Bagaimana hasil dari perancangan Sistem Informasi Perekaman Prestasi Mahasiswa dan perancangan *web service* berupa REST API berdasarkan hasil analisis kebutuhan yang telah didefinisikan?
3. Bagaimana hasil dari implementasi Sistem Informasi Perekaman Prestasi Mahasiswa dan implementasi *web service* berupa REST API berdasarkan hasil perancangan yang telah dibuat?
4. Bagaimana hasil dari pengujian Sistem Informasi Perekaman Prestasi Mahasiswa dan pengujian *web service* berupa REST API berdasarkan hasil implementasi?

1.3 Tujuan

1. Melakukan analisa kebutuhan Sistem Informasi Perekaman Prestasi Mahasiswa berdasarkan kebutuhan pengguna.
2. Melakukan perancangan Sistem Informasi Perekaman Prestasi Mahasiswa dan perancangan *web service* berupa REST API berdasarkan analisis kebutuhan yang telah didefinisikan.
3. Melakukan implementasi Sistem Informasi Perekaman Prestasi Mahasiswa dan implementasi *web service* berupa REST API berdasarkan hasil perancangan yang telah dibuat.



4. Melakukan pengujian Sistem Informasi Perekaman Prestasi Mahasiswa dan pengujian *web service* berupa REST API yang telah diimplementasikan.

1.4 Manfaat

Dengan adanya penelitian ini diharapkan pihak-pihak yang berkaitan dengan penelitian yang dilakukan mendapat manfaat, diantaranya:

1.4.1 Bagi Peneliti

Sebagai sarana untuk menambah pengetahuan dan pengalaman ketika melakukan penelitian.

1.4.2 Bagi Fakultas Teknologi Pertanian Universitas Brawijaya

1. Pihak pimpinan dapat lebih mudah untuk mengetahui prestasi yang telah di dapat oleh mahasiswa Fakultas Teknologi Pertanian.
2. Mahasiswa Fakultas Teknologi Pertanian dapat lebih mudah untuk mengumpulkan prestasi yang telah didapat.
3. Pihak kemahasiswaan lebih mudah dalam melakukan validasi prestasi yang telah di dapat oleh mahasiswa Fakultas Teknologi Pertanian.

1.4.3 Bagi Mahasiswa

Sebagai sumber maupun referensi untuk penelitian terkait pengembangan sistem informasi perekaman prestasi, pengembangan sistem informasi berbasis *website*, maupun pengembangan sistem informasi dengan teknologi *Restful web service*.

1.5 Batasan Masalah

Bedasarkan latar belakang dan rumusan masalah yang ada, maka dibuat batasan masalah sebagai berikut:

1. Sistem yang dikembangkan berfokus pada proses perekaman prestasi mahasiswa Fakultas Teknologi Pertanian Universitas Brawijaya.
2. Metode pengembangan menggunakan metode *Prototyping* dengan menerapkan tahap analisis kebutuhan, *Prototyping*, perancangan, implementasi, dan pengujian.
3. Sistem yang dikembangkan merupakan sebuah *website* dan API yang dijalankan pada *local server*.

1.6 Sistematika Pembahasan

Sistematika pembahasan yang berisikan gambaran masing – masing bab pada laporan penelitian dibuat dengan tujuan mempermudah pemahaman laporan penelitian ini.



BAB 1 PENDAHULUAN

Menjelaskan mengenai latar belakang dalam pemilihan judul skripsi pada penelitian ini, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika pembahasan.

BAB 2 LANDASAN PUSTAKA

Menjelaskan mengenai beberapa teori yang berhubungan dengan penelitian berupa landasan teori yang mendasari proses pembuatan sistem.

BAB 3 METODOLOGI

Menjelaskan mengenai langkah dan metode yang diterapkan pada penelitian ini. Dimulai dengan studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, kemudian kesimpulan dan saran.

BAB 4 ANALISIS KEBUTUHAN

Menjelaskan tentang analisis kebutuhan dari pengguna yang hasilnya digunakan sebagai dasar dalam proses perancangan sistem.

BAB 5 PERANCANGAN SISTEM

Menjelaskan tentang proses perancangan sistem berdasarkan analisis kebutuhan.

BAB 6 IMPLEMENTASI SISTEM

Menjelaskan tentang implementasi sistem dalam bentuk *Source Code* yang pembuatannya mengacu pada perancangan sistem.

BAB 7 PENGUJIAN SISTEM

Menjelaskan tentang pengujian dari sistem yang telah diimplementasikan dengan tujuan untuk menyesuaikan dengan analisis kebutuhan.

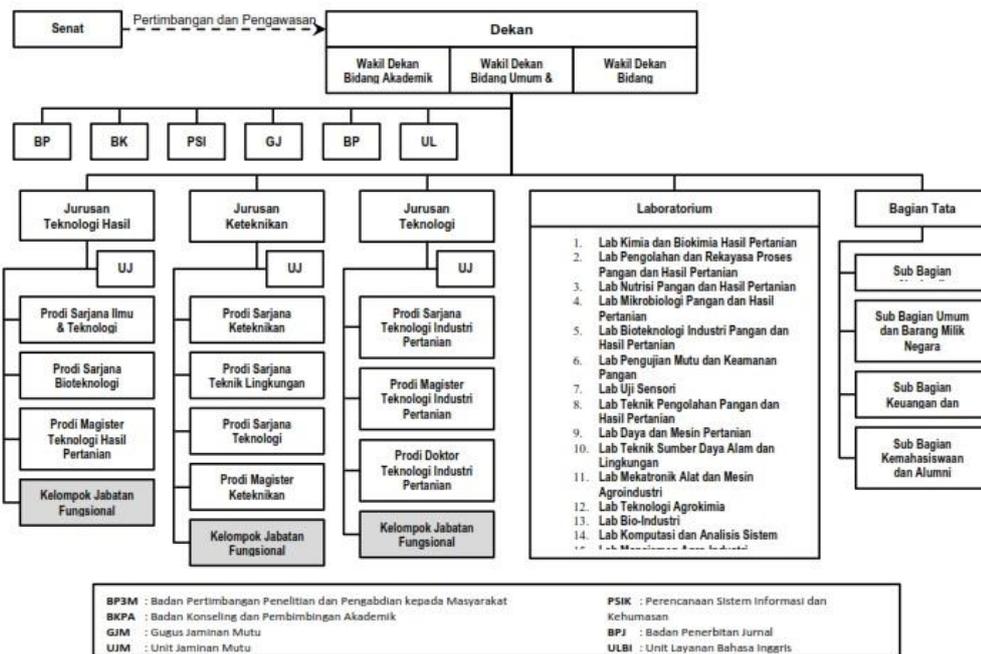
BAB 8 KESIMPULAN DAN SARAN

Menjelaskan tentang kesimpulan dari penelitian yang telah dibuat serta pemberian saran.

BAB 2 LANDASAN PUSTAKA

2.1 Fakultas Teknologi Pertanian Universitas Brawijaya

Fakultas Teknologi Pertanian merupakan salah satu Fakultas di Universitas Brawijaya. Fakultas Teknologi Pertanian (FTP) didirikan pada tanggal 26 Januari 1998. Sampai sekarang, Fakultas Teknologi Pertanian telah memiliki tiga jurusan, diantaranya jurusan Teknologi Hasil Pertanian, jurusan Keteknikan Pertanian, dan jurusan Teknologi Industri Pertanian (PSIK TP UB, 2019). Adapun struktur organisasi dari Fakultas Teknologi Pertanian ditunjukkan pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Fakultas Teknologi Pertanian Universitas Brawijaya

Sumber: Website Fakultas Teknologi Pertanian Universitas Brawijaya

Dalam hal pelayanan kepada mahasiswa khususnya di bidang pelayanan non akademik, Fakultas teknologi Pertanian mempunyai sub bagian Kemahasiswaan dan Alumni. Selain itu, sub bagian Kemahasiswaan juga bertugas dalam proses pendataan prestasi mahasiswa yang ada di Fakultas Teknologi Pertanian. Pada prestasi yang telah didata oleh sub bagian Kemahasiswaan maka selanjutnya dilakukan rekapitulasi oleh sub bagian akademik. Serta dalam hal perencanaan dan pelayanan teknologi, Fakultas Teknologi Pertanian dibantu oleh bagian Perencanaan Sistem Informasi dan Kehumasan (PSIK).

2.2 Prestasi Mahasiswa

Sebagai mahasiswa, Selain diharapkan dapat menguasai *hard skill* berdasarkan bidang yang ditekuni juga diharapkan menguasai *soft skill* sebagai penunjang



keberhasilan di masa mendatang dengan menciptakan karakter unggul pada mahasiswa. Hal ini dapat didukung oleh pihak perguruan tinggi khususnya bidang kemahasiswaan. Bentuk dukungan bisa berupa kelembagaan di bidang kemahasiswaan dan juga dorongan kepada mahasiswa untuk mendapat prestasi kejuaraan. Pada buku panduan simkatmawa (sistem informasi manajemen pemeringkatan kemahasiswaan), prestasi terbagi menjadi 3 aspek berdasarkan kegiatannya yaitu (Ristekdikti, 2019):

1. Kegiatan non lomba/pengakuan/rekognisi
2. Prestasi kegiatan ko dan ekstrakurikuler mandiri
3. Prestasi kegiatan ko dan ekstrakurikuler belmawa

2.3 Sistem Informasi

Sebuah sistem informasi dapat diartikan sebagai suatu kumpulan komponen yang mengumpulkan, menyimpan, dan mendistribusikan informasi yang digunakan sebagai pendukung dalam kegiatan pengambilan keputusan dan mengatur suatu organisasi. Selain itu, sistem informasi juga digunakan sebagai pendukung dalam pengambilan keputusan, penkoordinasian, dan kontrol, sistem informasi juga membantu pengguna dalam menganalisa masalah, menggambarkan objek yang kompleks, dan membuat produk baru (Laudon, 2014).

2.4 Teknologi Pendukung

Dalam mengembangkan sebuah sistem, dapat dibantu dengan teknologi bernama *web service*. Tujuan dari penggunaan *web service* adalah *client* dari berbagai macam platform berupa *desktop*, *website*, dan *mobile apps* dapat mengakses layanan pada *web service* (Rohman, Soebroto, & Kharisma 2018). Pada penelitian Mumbaikar dan Padiya (2013) dijelaskan bahwa *web service* adalah platform yang bersifat independen. Terdapat 2 tipe arsitektur dari *web service*, yaitu SOAP (*Simple Object Access Protocol*) dan REST (*REpresentational State Transfer*). REST merupakan arsitektur *web service* yang bersifat *client server* dimana *client* melakukan *request* kepada *server* dan *server* memproses *request* dan mengembalikan *response*. Restful *web service* merupakan sebutan untuk aplikasi web yang menggunakan arsitektur dari REST.

Dalam menggunakan teknologi REST dibantu dengan sebuah *tools* yang bernama API (*Application Programming Interface*) berbasis *website*. API secara umum terdiri dari 2 bagian, yaitu *server* sebagai penyedia data dan *client* yang dapat melakukan *request* data. Pada API berbasis *website* dibagi menjadi dua yaitu REST API dan SOAP API. REST API adalah API berbasis *website* yang menggunakan teknologi REST dan menggunakan format JSON (*JavaScript Object Notation*), yaitu sebuah format pertukaran data yang bisa digunakan baik pada *front-end* maupun *back-end* dari aplikasi *website* maupun sebuah *service* (Cooksey B., 2014).



2.5 Web Service

Web service adalah sebuah teknologi yang membuat internet agar bisa saling melakukan komunikasi dengan pola *program-to-program* (Rohman, Soebroto, & Kharisma 2018). Web service memungkinkan komunikasi antar *client – server* melalui *Hyper Text Transfer Protocol* (HTTP) dalam pertukaran data pada beberapa aplikasi berbeda *platform* (Bhuvanewari, dkk., 2011).

Pada penelitian christanto dan kurniawati (2015) mennyebutkan beberapa keuntungan dalam menggunakan teknologi *web service*, diantaranya:

1. Fungsi yang dapat diakses melalui jaringan
2. Dapat digunakan kembali (*reuse*)
3. Mudah dalam pengaksesan data
4. Dapat menghubungkan berbagai jenis aplikasi
5. *Protocol* standar yang digunakan untuk melakukan komunikasi

2.6 Representational State Transfer (REST)

Representational State Transfer (REST) adalah sebuah gaya arsitektur untuk pendistribusian sistem *hypermedia*. Arsitektur REST adalah arsitektur klien – server dimana klien mengirim *request* pada *server* dan server memproses *request* dan mengembalikan sebuah *response* (transaksi). Setiap transaksi bersifat independen dan tidak terkait dengan transaksi lainnya (*stateless*). Hal ini yang membuat aplikasi REST sederhana dan ringan. *RESTful web service* adalah sebutan untuk aplikasi *web* yang menggunakan arsitektur REST. *RESTful web service* menggunakan metode *http GET, POST, PUT, dan DELETE* untuk menerima, membuat, memperbarui dan menghapus *resource* (Mumbaikar S., & Padiya P., 2013). Ketika menggunakan Arsitektur REST akan meningkatkan beberapa aspek, diantaranya (Doglio, 2015):

1. Performa, gaya komunikasi yang digunakan REST sederhana dan efisien, sehingga dapat meningkatkan performa dari sistem yang menggunakan teknologi ini.
2. Skalabilitas komponen dalam berinteraksi, setiap sistem terdistribusi harus bisa mengatasi hal ini, dan interaksi sederhana dari REST lebih mempermudah dalam mengatasi hal tersebut.
3. Kemampuan modifikasi pada komponen, dengan REST yang memudahkan komponen untuk dimodifikasi secara independen dengan biaya dan risiko yang minim.
4. Portabilitas, REST bersifat *technology and language agnostic* yang artinya dapat diimplementasi dan digunakan teknologi apapun.
5. Handal, sifat *stateless* REST memudahkan dalam hal perbaikan ketika terjadi kesalahan sistem



6. Visibilitas, sifat *stateless* REST memudahkan dalam hal visibilitas yang mana kita tidak perlu melihat lebih dari sebuah *request* untuk menentukan keadaan dari *request* tersebut.

Dapat disimpulkan bahwa keuntungan menggunakan REST diantaranya:

1. Desain yang berorientasi dengan komponen membuat sistem sangat toleran dengan kesalahan. Ketika terdapat satu kesalahan komponen maka tidak akan mempengaruhi stabilitas keseluruhan sistem.
2. Menghubungkan komponen cukup mudah, meminimalisir risiko ketika menambah fitur maupun memperbesar dan memperkecil sistem.
3. Sistem yang didesain menggunakan REST akan mudah diakses oleh orang luas.

2.6.1 REST Constraint

Dalam mendefinisikan arsitektur dari REST, sistem didefinisikan dalam keadaan null tanpa pembatas yang kemudian diberikan *constraint* atau pembatas secara satu per satu (Doglio, 2015).

1. Client – Server

Pembatasan pertama adalah melakukan pemisahan antara klien sebagai representasi pemrosesan antarmuka dari sistem dari server yang berisi penyimpanan dan pemrosesan data pada bagian server.

2. Stateless

Pembatasan kedua adalah komunikasi antar klien dan server harus bersifat *stateless*, yang mana tiap *request* dari klien harus melakukan *request* informasi yang dimiliki oleh server tanpa perlu bergantung dengan data lain yang disimpan diluar server (independen).

3. Cacheable

Pembatasan ketiga adalah data yang dikirim oleh respon harus dapat disimpan dalam *cache*, yang membuat beberapa interaksi dapat melakukan *bypass* ketika data terdapat pada *cache*. Yang perlu diperhatikan bahwa ada kemungkinan data yang disimpan dalam *cache* menjadi basi.

4. Uniform Interface

Pembatasan keempat adalah dengan membuat komponen memiliki antarmuka yang seragam, dapat menyederhanakan pekerjaan klien ketika melakukan interaksi dengan cara memberi peraturan yang jelas untuk diikuti.

5. Layered System

Pembatasan kelima adalah penggunaan konsep lapisan (*layer*) dengan cara memisahkan komponen berdasarkan lapisan dan memperbolehkan tiap lapisan melakukan memindahkan *output* yang didapat dari lapisan bawah kepada lapisan atasnya, dengan hal ini dapat menyederhanakan kompleksitas dari keseluruhan sistem dan menjaga komponen agar tidak *coupling* ketika



dicek. Hal ini akan sangat membantu untuk sistem besar yang akan terus berkembang.

2.6.2 Resource Identifier

Resource identifier dapat menyediakan cara yang unik mengidentifikasi setiap momen dan dapat menyediakan *full path* kepada *resource*. Penggunaan ID numerik sederhana tidak bisa dilakukan karena harus menyediakan *full path* nya, dan karena penggunaan REST berbasis HTTP, cara mengakses *resource* adalah dengan menyediakan URI (*Unique Resource Identifier*) secara penuh. Gambar 2.2 merupakan contoh dua buah URI yang memiliki *identifier* yang bersifat *unique* (Doglio, 2015).

```
GET /api/v1/books/j-k-rowling/harry-potter-and-the-deathly-hollows
GET /api/v1/books/j-k-rowling/harry-potter-and-the-half-blood-prince
```

Gambar 2.2 contoh *Unique Resource Identifier*

Sumber: Doglio (2015)

2.6.3 Actions

Karena REST menggunakan protokol HTTP sebagai acuan, yang mana tersedia kumpulan dari kata kerja yang dapat digunakan sebagai referensi tipe dari aksi yang dilakukan kepada *resource*. Tabel 2.1 merupakan penjelasan dari aksi HTTP (Doglio, 2015).

Tabel 2.1 HTTP verb dan aksi yang dilakukan

Sumber: Doglio (2015)

HTTP Verb	Aksi
GET	Mengakses <i>resource</i> pada mode <i>read only</i>
POST	Mengirim <i>resource</i> kepada server
PUT	Memperbarui <i>resource</i> yang dipilih
DELETE	Menghapus <i>resource</i>

2.6.4 Status Codes

Kode status yang merupakan dari HTTP merupakan sebuah angka yang merangkum *response* yang berkaitan. Kode status sangat membantu klien menafsirkan *response*. Pada tabel 2.2 merupakan penjelasan kode status yang biasa dijumpai (Doglio, 2015).

Tabel 2.2 Kode Status HTTP

Sumber: Doglio (2015)

Kode status	Arti
200	OK. <i>Request</i> dapat dijalankan dan konten yang di <i>request</i> telah dikembalikan



201	<i>Created.</i> <i>Resource</i> berhasil dibuat dan <i>server</i> mengakui hal tersebut
400	<i>Bad Request.</i> <i>Request</i> bermasalah seperti kesalahan parameter yang dibutuhkan
401	<i>Unauthorized.</i> Digunakan ketika mengakses <i>resource</i> yang tidak bisa diakses, biasa digunakan dalam autentikasi
403	<i>Forbidden.</i> <i>Resource</i> tidak bisa diakses
404	<i>Not found.</i> URL yang diberikan tidak cocok dengan <i>resource</i> manapun yang tersedia
405	<i>Method not Allowed.</i> Muncul ketika HTTP <i>verb</i> digunakan pada <i>resource</i> yang berbeda, menggunakan POST pada <i>read only resource</i>
500	<i>Internal Server Error.</i> Kesalahan kode ketika kondisi tidak terduga terjadi pada bagian server

2.6.5 REST dan pendahulunya

Sebelum REST dikembangkan terdapat sebuah teknologi yang memungkinkan interkoneksi antar sistem yang menggunakan bahasa universal. Pada tahun 1998 dikembangkan sebuah teknologi yang dapat melakukan interkoneksi antar sistem menggunakan RPC (*Remote Produce Call*) dan menggunakan XML sebagai bahasa perpindahan utama, teknologi ini bernama XML-RPC. Seiring berjalannya waktu teknologi XML-RPC semakin berkembang dan bermutasi menjadi SOAP (*Simple Object Access Protocol*) yang merupakan versi formal dan terstandarisasi dari prinsip teknologi yang sama. Meskipun menggunakan bahasa universal namun antara kedua teknologi ini memiliki karakteristik masing – masing yang dijelaskan pada tabel 2.3 (Doglio, 2015).

Tabel 2.3 Perbandingan antara XML-RPC/SOAP dan REST

Sumber: Doglio (2015)

XML-RPC/SOAP	REST
Klien dari SOAP harus dibuat berdasarkan setiap bahasa pemrograman. Klien baru harus dikodekan terlebih dahulu untuk menguraikan WSDL untuk memahami bagaimana <i>service</i> bekerja	REST hanya membutuhkan bahasa pemrograman yang mampu mengatasi konektivitas dari protokol yang dipilih
Klien harus tahu semua tentang <i>service</i> sebelum interaksi diinisiasi	Klien hanya butuh tau <i>endpoint root</i> utama, dengan <i>response</i> pada <i>hypermedia</i> membuat pendeteksian secara otomatis dapat terjadi



Karena *service* digunakan dalam klien *source code* dan memanggil fungsi secara spesifik dari kode server, terjadinya *coupling* antara dua sistem terbilang besar.

Implementasi dari antarmuka bersifat independen, dapat dilakukan *rewrite* pada kode server tanpa perlu merubah antarmuka dari API

2.7 REST API

Application Programming Interface (API) adalah sebuah *tools* yang membuat beberapa sistem dapat saling terhubung. Pada API terdapat dua bagian, yaitu server yang berfungsi sebagai penyedia dari API dan klien yang berbentuk sebuah program yang mengetahui data apa yang tersedia pada API dan dapat memanipulasi data tersebut sesuai *request* pengguna. Contohnya adalah sebuah aplikasi *smartphone* yang tersinkron dengan *website*. Ketika ditekan tombol *refresh* pada aplikasi, aplikasi akan meminta server melalui API dan menerima informasi terbaru (Cooksey B., 2014).

API berbasis *website* dibagi menjadi dua yaitu REST API dan SOAP API. REST API adalah API berbasis *website* yang menggunakan teknologi REST dan menggunakan format JSON (*JavaScript Object Notation*). REST API terdiri dari (Cooksey B., 2014):

1. *Resource* yang dibutuhkan.
2. URL untuk mengakses *resource* tersebut.
3. Aksi yang diperbolehkan untuk klien dalam mengelola *resource* (berupa GET, POST, PUT, DELETE).
4. Data yang diperlukan dalam melakukan aksi dan format yang harus dibawa.

2.8 Javascript Object Notation (JSON)

JSON adalah sebuah format pertukaran data yang bersifat independen dari bahasa pemrograman lain (Afsari, dkk., 2017). Afsari dan Lacouture (2017) mengatakan bahwa tidak seperti XML, JSON adalah sebuah format yang mudah untuk dipindah dan *generate*, selain itu juga mudah dibaca oleh manusia dan gambar 2.3 merupakan contoh dari format JSON. Data pada JSON terdiri dari 3 bagian yang diantaranya (Christanto dan Kurniawati, 2015):

1. Pasangan *key* dan *value* yang dibatasi dengan tanda “:”
2. JSON *arrays* yang merupakan sekumpulan JSON *object* yang direpresentasikan dengan “[]”
3. JSON *object* yang terdiri dari beberapa JSON *object* atau JSON *array* yang direpresentasikan dengan “{ }”

```

{
  "type": "Feature",
  "id": "f1",
  "geometry": {...},
  "properties": {...},
  "title": "Example Feature"
}

```

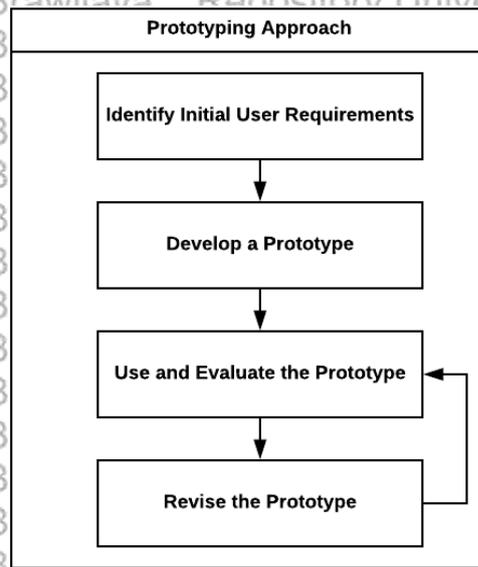
Gambar 2.3 Contoh format JSON

Sumber: Afsari, dkk (2017)

2.9 Metode Prototyping

Prototype pada sistem informasi berarti sebuah versi awal dari sistem yang menunjukkan fitur – fitur operasional penting dari sistem nantinya. Nauman dan Jenkins (1982) mengatakan bahwa *prototype* sistem informasi adalah sebuah sistem yang menangkap fitur penting dari sistem, *prototype* dibuat dalam bentuk belum jadi yang nantinya akan dimodifikasi, ditambah, maupun digantikan. Tiap *prototype* diuji kepada *user* dan dikembalikan kepada tim pengembang untuk dikerjakan ulang dan nantinya *prototype* akan berubah menjadi produk final, memberikan keuntungan kepada *user* untuk meninjau ulang kebutuhan dan melihat hasil implementasi perangkat lunak (CASEMaker, 2000).

Pada metode *prototyping* menyajikan sebuah *base line*. Yaitu sebuah informasi dasar yang digambarkan pada *prototype* yang telah dibuat yang nantinya digunakan sebagai poin referensi baik untuk pengguna maupun pengembang dimana dapat mengidentifikasi lebih awal masalah dan peluang pada proses pengembangan. Selain itu *prototyping* merupakan cara yang efektif untuk menggambarkan dan mengklarifikasikan kebutuhan pengguna (Alavi M., 1984). Untuk alur pengembangan sistem informasi dengan metode *prototyping* dapat dilihat pada Gambar 2.3.



Gambar 2.4 Alur pengembangan sistem informasi dengan metode *prototyping*

Sumber: Alavi M. (1984)



2.10 Unified Modelling Language

Unified Modeling Language (UML) merupakan sebuah standar bahasa permodelan untuk pengembangan sistem dan perangkat lunak. Dari permasalahan yang kompleks dilakukan sebuah permodelan dengan tujuan menyederhanakan permasalahan dan juga agar mudah dipelajari (Hendini, 2016). Terdapat enam keuntungan utama pada pendekatan ini, antara lain (Miles, 2006):

1. Bahasa yang digunakan adalah bahasa formal: setiap bahasa yang digunakan mempunyai arti yang kuat, jadi ketika memodelkan suatu sistem tidak akan ditemukan kesalahpahaman.
2. Ringkas: setiap bahasa yang dipakai menggunakan notasi yang sederhana dan terus terang.
3. Lengkap: menjelaskan keseluruhan aspek penting dari sistem.
4. Scalable: ketika diperlukan, bahasa formal sudah cukup untuk mengatasi permodelan proyek sistem secara masif, namun juga harus mencakup proyek kecil.
5. Dibuat berdasarkan pelajaran yang telah dipelajari: UML adalah puncak dari *best practice* pada komunitas *object-oriented* selama lima belas tahun terakhir.
6. Menjadi standar: UML dikontrol oleh grup standar dengan kontribusi aktif dari vendor dan akademik yang mendunia.

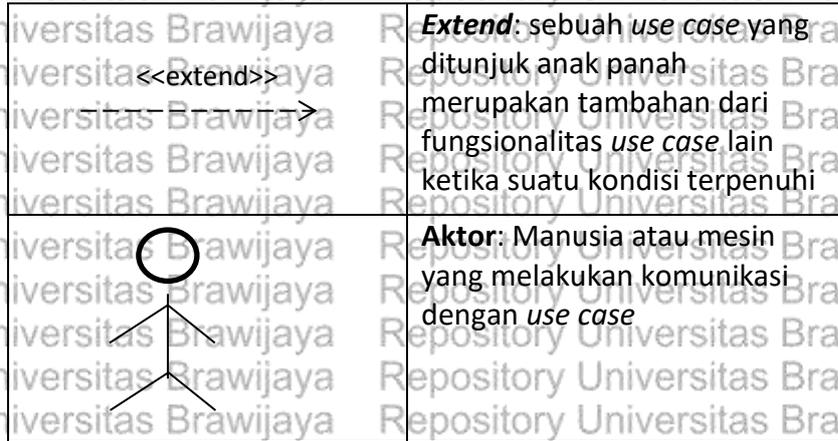
2.10.1 Use Case Diagram

Pada *use case diagram* lebih difokuskan untuk menggambarkan fungsionalitas dari sistem berdasarkan kebutuhan pengguna. Pada *use case diagram* mempresentasikan interaksi antar aktor dengan sistem. Aktor adalah sebuah entitas yang bisa terdiri dari manusia yang berinteraksi dengan sistem dalam melakukan suatu pekerjaan tertentu (Dharwiyanti S., & Wahono R.S., 2003). Adapun Penjelasan komponen dari *use case diagram* bisa dilihat pada Tabel 2.1.

Tabel 2.4 Komponen *use case diagram*

Sumber: Booch, et al (1998)

Simbol	Keterangan
	Asosiasi: sebuah penghubung antara aktor dengan <i>use case</i>
	Include: sebuah <i>use case</i> yang ditunjuk anak panah membutuhkan fungsionalitas <i>use case</i> lain



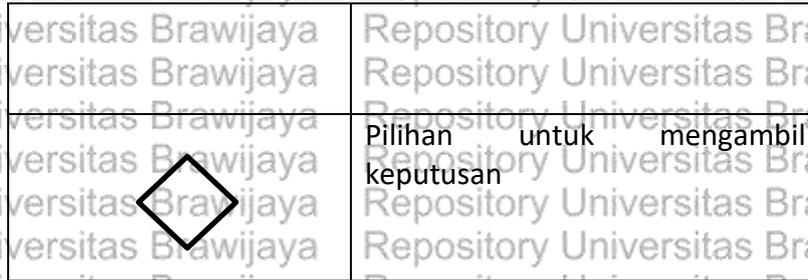
2.10.2 Activity Diagram

Activity diagram adalah sebuah diagram yang menggambarkan aktivitas dari suatu fungsi pada sistem dan transisi yang dipicu oleh sumber (aktor). Diagram ini berfokus pada proses internal. Penjelasan komponen dari diagram *activity* dapat dilihat pada Tabel 2.2.

Tabel 2.5 Komponen *activity diagram*

Sumber: Booch, et al (1998)

Simbol	Keterangan
	<p>Activity: Menunjukkan interaksi pada antarmuka masing – masing <i>class</i></p>
	<p>Action: State dari sistem yang menunjukkan eksekusi dari suatu aksi</p>
	<p>Start State: awalan dari sebuah <i>activity</i></p>
	<p>State Transition: State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan</p>
	<p>End State: awalan dari sebuah <i>activity</i></p>



2.10.3 Sequence Diagram

Sequence diagram merupakan sebuah diagram yang tentang interaksi pada sistem baik di dalam maupun di sekitar yang digambarkan berupa pesan yang dikirim. Pada *sequence diagram* juga menggambarkan rangkaian atau skenario dari *event* yang menghasilkan sebuah *output*. Dimulai dengan pemicu aktivitas, proses berupa perubahan internal yang terjadi, hingga *output* yang dihasilkan (Dharwiyanti S., & Wahono R.S., 2003). Pada Tabel 2.3 menjelaskan tentang komponen dari *sequence diagram*.

Tabel 2.6 Komponen *sequence diagram*

Sumber: Hendini (2016)

Simbol	Keterangan
	Aktor: menggambarkan orang yang melakukan interaksi dengan sistem
	Entity Class: berisi kumpulan kelas berupa entitas yang membentuk gambaran awal sistem
	Boundary Class: berisi kumpulan kelas yang menjadi <i>interface</i> seperti tampilan form
	Control Class: berisi kumpulan logika aplikasi contoh seperti kalkulasi dan aturan bisnis yang melibatkan objek lain



Activation: sebuah eksekusi operasi dari objek

Message: simbol mengirim pesan antar kelas

2.10.4 Class Diagram

Class diagram merupakan spesifikasi yang ketika diinstansiasi akan menghasilkan sebuah objek. Pada *class diagram* menjelaskan tentang kondisi yang digambarkan berupa atribut serta metode/fungsi pada tiap *class* yang saling berelasi (Dharwiyanti S., & Wahono R.S.). Tabel 2.4 menjelaskan tentang komponen dari *class diagram*.

Tabel 2.7 Komponen *class diagram*

Sumber: Booch, et al (1998)

Simbol	Keterangan
	Generalization: Penghubung antar <i>class</i> guna berbagi perilaku dan struktur dari objek yang ada
	Class: Himpunan dari objek-objek yang berbagi atribut serta metode/fungsi yang sama

2.11 MySQL

MySQL merupakan sistem untuk manajemen *database* yang dapat digunakan secara gratis atau biasa disebut *open source*. MySQL menggunakan bahasa SQL (*Structured Query Language*) yang merupakan bahasa yang umum digunakan untuk pengaksesan *database* dan juga termasuk *relational database* yang artinya dalam penyimpanan suatu data dilakukan dengan membagi berdasarkan tabel yang terpisah dengan aturan relasi antar tabelnya (MySQL, 2020). Pada penelitian ini perancangan pada *database* digambarkan dengan menggunakan *physical data model* berdasarkan hasil perancangan pada *class diagram*.



2.12 Laravel

Laravel adalah sebuah *framework* PHP perangkat lunak berbasis web. Pada *framework* Laravel berfokus meringankan beban dalam proses pengembangan *website* pada beberapa aspek, seperti autentikasi, *routing*, *session*, dan *caching* (Laravel, 2019). *Framework* Laravel memiliki beberapa keuntungan, diantaranya:

1. Laravel membuat kode program lebih jelas dan sederhana.
2. Laravel memudahkan pengembang untuk mempelajari kode program.
3. Laravel menyediakan seluruh ekosistem untuk pengembangan aplikasi *website*.
4. Laravel hanya membutuhkan sedikit konfigurasi.

2.13 Lumen

Lumen adalah sebuah *framework* yang didesain untuk menulis API. Lumen menggunakan bagian komponen yang sama dari Laravel. Lumen juga mempunyai sebagian dari *console tools* yang ada pada Laravel dan juga beberapa fitur dari Laravel yaitu *database migration*, *Eloquent Models (ORM Package)*, *job queues*, dan *scheduled jobs*. Pada pengalaman pengembangan Lumen dan Laravel bisa dibilang sama. Yang berarti akan mempermudah pengembang jika mengadopsi *framework* Laravel dan Lumen (Redmond Paul, 2016).

2.14 Pengujian Perangkat Lunak

Pengujian pada perangkat lunak dilakukan kepada sistem yang telah dikembangkan dengan tujuan untuk menemukan *error* yang dibuat secara tidak sengaja sesuai dengan apa yang dirancang dan dibangun (Pressman, 2010). Ada 2 jenis pengujian, diantaranya adalah *white box testing* dan *black box testing*.

2.14.1 Black Box Testing

Black box Testing merupakan sebuah metode pengujian perangkat lunak dimana penguji tidak mengetahui baik struktur internal/ rancangan/ pengembangan dari item yang akan diuji. Biasanya digunakan untuk melakukan pengujian fungsional dari suatu perangkat lunak (*softwaretestingfundamentals*, 2019).

Pada pengujian *black box* bertujuan untuk mencoba menemukan *error* dengan beberapa kategori sebagai berikut (*softwaretestingfundamentals*, 2019):

1. Kesalahan maupun fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data atau pada akses *database* dari luar.
4. Kesalahan performa atau behavioral.
5. Kesalahan inialisai dan terminasi.



2.14.2 Usability Testing

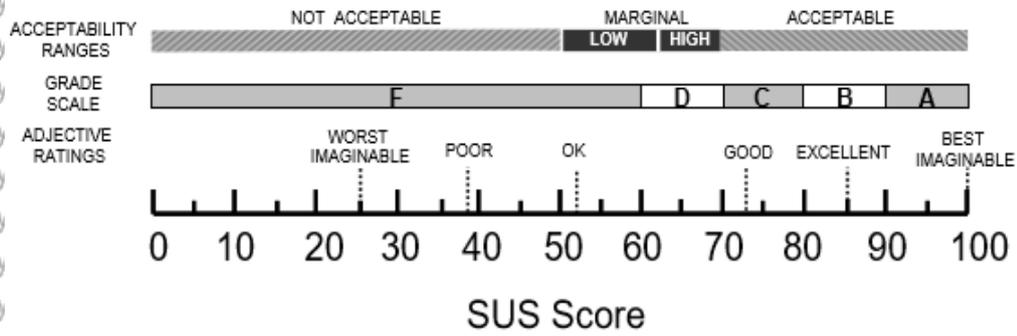
Usability Testing merupakan sebuah pengujian yang dilakukan untuk mengevaluasi tingkat keefektifan pengguna dalam berinteraksi dengan sistem serta bagaimana sistem memandu pengguna ketika dioperasikan (Pressman, 2010). Pengujian ini dilakukan dengan mengukur interaksi antara sistem dan pengguna sistem dengan menggunakan metode *System Usability Scale* (SUS).

System Usability Scale (SUS) terdiri dari sepuluh pernyataan sederhana yang dapat dilihat pada Tabel 2.5. Dimana pada setiap pernyataan, responden diminta untuk menunjukkan tingkat kesetujuan dari 5 skala (tidak setuju hingga sangat setuju). Lalu dilakukan perhitungan skor SUS. Setiap item pernyataan bernilai anantara 0 – 4. Untuk item ganjil maka nilai akan dikurangi 1 dan untuk item genap nilai 5 akan dikurangi dengan nilai yang didapat. Setelah itu hasil penjumlahan dari seluruh item dikali dengan 2,5. Jika terdapat lebih dari satu responden maka dicari rata – rata dari skor SUS keseluruhan responden. Skor SUS mempunyai rentang dari 0 – 100 dengan klarifikasi pada tiap rentang yang ada yang ditunjukkan melalui rentang penerimaan, skala penilaian, serta penilaian adjektif (Brooke J., 1986). Berikut adalah rentang nilai dan penerimaan dalam SUS yang bisa dilihat pada Gambar 2.4 (Bangor A. dkk, 2009).

Tabel 2.8 Daftar Pernyataan SUS

Sumber: Sharfina Z., dkk (2016)

No.	Pernyataan
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.
3	Saya merasa sistem ini mudah untuk digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.



Gambar 2.5 Rentang nilai SUS

Sumber : Bangor A., dkk (2009)

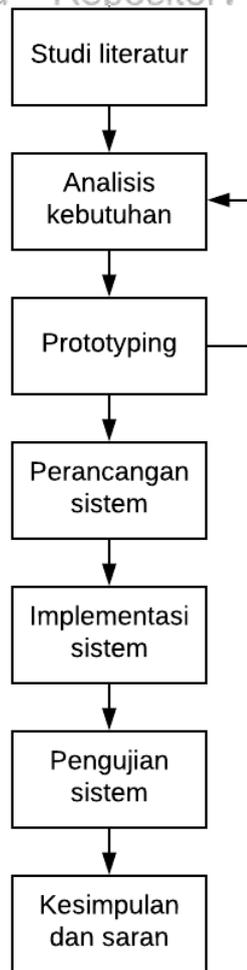
2.15 Postman

Postman merupakan sebuah platform kolaborasi untuk pengembangan API. Fitur dari Postman menyederhanakan setiap langkah dari pembangunan API yang dapat membuat pengembangan API menjadi lebih cepat. Beberapa hal yang dapat dilakukan oleh Postman diantaranya adalah (Postman, 2020):

1. Bertindak sebagai klien API dan melakukan request REST, SOAP, maupun GraphQL melalui Postman.
2. Pengujian yang terotomatisasi
3. Simulasi dengan *endpoint* secara langsung tanpa harus mengotak atik *backend server*.
4. Dokumentasi API.
5. Pemantauan performa dan waktu respon dari API.
6. Menyediakan konteks berbagi dalam *workspace* dalam membangun dan menggunakan API secara *real-time*.

BAB 3 METODOLOGI

Penelitian yang dilakukan termasuk ke dalam jenis penelitian implementatif dengan tipe pengembangan. Penelitian berlokasi pada Fakultas Teknologi Pertanian Universitas Brawijaya. Pengumpulan data dilakukan dengan melakukan wawancara dengan bapak Mas'ud Effendi STP, MP, selaku ketua PSIK Fakultas Teknologi Pertanian. Metodologi penelitian dimulai dari Identifikasi masalah hingga kesimpulan dan saran. Gambar 3.1 menunjukkan metodologi pengerjaan dari penelitian ini.



Gambar 3.1 Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan metode pemahaman dan kajian teori dengan mencari dan mempelajari literatur dari laporan penelitian, buku, artikel, maupun jurnal. Dilakukan studi literatur dengan tujuan untuk mengumpulkan informasi yang berkaitan dengan pengembangan sistem dan digunakan untuk menunjang penulisan laporan. Pada penelitian ini teori – teori yang perlu dipelajari meliputi teori tentang:



1. Fakultas Teknologi Pertanian Universitas Brawijaya
2. Prestasi mahasiswa
3. Sistem informasi
4. Teknologi pendukung
5. *Web service*
6. *Representational State Transfer (REST)*
7. REST API
8. *JavaScript Object Notation (JSON)*
9. Metode *Prototyping*
10. *Unified Modelling Language (UML)*
11. MySQL
12. Laravel
13. Lumen
14. Pengujian perangkat lunak
15. Postman

3.2 Analisis Kebutuhan

Tahap ini dimulai dengan melakukan analisa kondisi di Fakultas Teknologi Pertanian saat ini dengan melakukan wawancara kepada bapak Mas'ud Effendi, STP., MP. Selaku ketua PSIK Fakultas Teknologi Pertanian untuk mengetahui permasalahan yang sedang dialami dan harapan dengan dikembangkannya sistem yang dapat menyelesaikan permasalahan tersebut, gambaran umum dari sistem, aktor yang akan menggunakan sistem, serta daftar kebutuhan dari sistem. Setelah itu dibuat rancangan *prototype* berdasarkan daftar kebutuhan dan dilakukan presentasi *prototype*. Jika terdapat penambahan atau perubahan maka akan dilakukan pembaharuan pada *prototype* dan dilakukan presentasi kembali. Hal ini dilakukan hingga ketua PSIK menyetujui *prototype* yang ditunjukkan. Hasil dari analisis kebutuhan dan *prototyping* digambarkan dalam bentuk *use case diagram*, *use case scenario*, serta *activity diagram*.

3.3 Perancangan Sistem

Ketika kebutuhan sistem telah teridentifikasi melalui tahap analisis kebutuhan. Hasil analisis kebutuhan selanjutnya dimodelkan menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Hasil pemodelan digambarkan dalam bentuk *sequence diagram* dan *class diagram*. selanjutnya dilakukan perancangan *database* dalam bentuk *Physical Data Model*. Setelah itu dilakukan perancangan REST API.



3.4 Implementasi Sistem

Dari hasil perancangan yang telah dibuat, selanjutnya dilakukan implementasi dari sistem ke dalam bentuk program. Pada tahap ini sistem diimplementasikan menggunakan *framework* Laravel untuk *website* dan Lumen untuk REST API.

3.5 Pengujian Sistem

Setelah sistem selesai diimplementasikan, maka dilakukan pengujian sistem dengan tujuan membandingkan sistem yang telah dibuat apakah sesuai dengan kebutuhan yang telah dijabarkan sebelumnya. Pada tahap ini dilakukan pengujian REST API, pengujian *black box* serta pengujian *Usability* pada beberapa aktor dengan menggunakan pengujian *System Usability Scale* (SUS).

3.6 Kesimpulan dan Saran

Pada tahapan ini akan ditarik kesimpulan dari tahapan – tahapan yang sebelumnya telah dilakukan. Pada tahap ini juga dikumpulkan kritik dan saran yang digunakan untuk meningkatkan kualitas dari pembuatan sistem.



BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem informasi perekaman prestasi mahasiswa adalah sebuah sistem berbasis *website* yang dibuat untuk melakukan perekaman prestasi mahasiswa Fakultas Teknologi Pertanian Universitas Brawijaya. Perekaman dilakukan dengan cara mahasiswa melakukan tambah prestasi yang telah didapat beserta bukti berupa foto prestasi atau sertifikat penghargaan. Setelah itu dilakukan verifikasi oleh pihak kemahasiswaan Fakultas Teknologi Pertanian Universitas Brawijaya. Setelah terverifikasi maka prestasi dapat dilihat oleh pihak akademik Fakultas Teknologi Pertanian. Selain itu, pihak akademik Fakultas Teknologi Pertanian juga bisa mengunduh daftar prestasi dalam format Tabel (*excel*). Mahasiswa juga bisa mengunduh *draft* prestasi mahasiswa. *Draft* prestasi mahasiswa berupa daftar prestasi yang telah dimasukkan dan telah diverifikasi oleh kemahasiswaan Fakultas Teknologi Pertanian, *draft* akan diunduh dalam format pdf.

4.2 Identifikasi Aktor

Berdasarkan hasil wawancara yang telah dilakukan, maka dapat dilakukan identifikasi aktor pada sistem informasi perekaman prestasi mahasiswa. Tabel 4.1 menunjukkan identifikasi aktor serta penjelasan mengenai tugas dan peran masing – masing terhadap sistem.

Tabel 4.1 Identifikasi aktor

No.	Identifikasi Aktor	Karakteristik Aktor
1	Mahasiswa	Mahasiswa adalah aktor yang dapat melakukan tambah prestasi yang telah didapat oleh mahasiswa pada sistem. Mahasiswa juga dapat mengunduh <i>draft</i> prestasi yang telah diverifikasi dalam bentuk pdf.
2	Kemahasiswaan	Kemahasiswaan adalah aktor yang dapat melakukan verifikasi terhadap prestasi yang telah dimasukkan oleh mahasiswa. Kemahasiswaan juga dapat melakukan tambah prestasi berdasarkan kegiatan (memasukkan beberapa nama mahasiswa sekaligus).
3	Akademik	Akademik adalah aktor yang dapat melihat dan mengunduh daftar prestasi mahasiswa dalam bentuk <i>excel</i> .



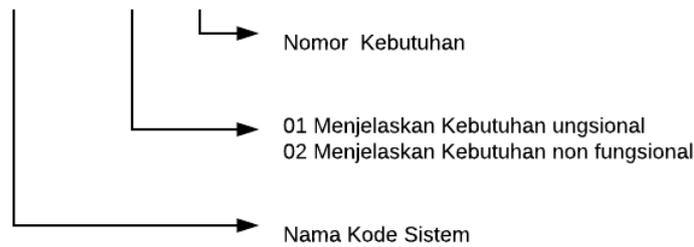
4.3 Daftar Kebutuhan Sistem

Dilakukan pendataan daftar kebutuhan dengan tujuan mendapatkan informasi mengenai kebutuhan apa saja yang dibutuhkan oleh pengguna terhadap sistem informasi perekaman prestasi mahasiswa. Sebelum dilakukan pendataan sistem dilakukan pembuatan kode dan aturan penomoran. Tujuan dari pembuatan kode dan aturan penomoran adalah untuk mempermudah dalam membedakan satu kebutuhan dengan yang lain.

4.3.1 Aturan Penomoran

Format pada aturan penomoran yaitu adalah (Nama Kode Sistem) (Jenis Kebutuhan) (Nomor Kebutuhan). Contoh dan Penjelasan aturan penomoran dapat dilihat pada Gambar 4.1.

SKPI-01-001



Gambar 4.1 Aturan penomoran sistem informasi perekaman prestasi mahasiswa

Kebutuhan fungsional merupakan kebutuhan utama dari sistem informasi perekaman prestasi mahasiswa. Pada Tabel 4.2 disebutkan dan dijelaskan kode, nama fungsi, serta deskripsi dari kebutuhan fungsional. Pada Tabel 4.3 disebutkan dan dijelaskan kode, nama fungsi, serta deskripsi dari kebutuhan non fungsional.

Tabel 4.2 Kebutuhan fungsional sistem informasi perekaman prestasi mahasiswa

No.	Kode	Nama Fungsi	Deskripsi
1	SKPI-01-001	Mahasiswa Login	Mahasiswa, Kemahasiswaan, dan akademik dapat melakukan Login menggunakan nomor induk dan password
2	SKPI-01-002	Mahasiswa Tambah prestasi	Mahasiswa dapat memasukkan data prestasi yang telah didapat (beserta bukti berupa sertifikat maupun dokumentasi)
3	SKPI-01-003	Mahasiswa Lihat Prestasi	Mahasiswa dapat melihat riwayat prestasi yang telah dimasukkan



4	SKPI-01-004	Mahasiswa Hapus Prestasi	Mahasiswa dapat menghapus prestasi yang telah dimasukkan dengan status menunggu
5	SKPI-01-005	Mahasiswa Unduh Draft	Mahasiswa dapat mengunduh Draft prestasi yang telah dimasukkan dan telah diverifikasi
6	SKPI-01-006	Kemahasiswaan Tambah kegiatan	Kemahasiswaan dapat memasukkan data beberapa peraih prestasi dalam satu kegiatan
7	SKPI-01-007	Verifikasi	Kemahasiswaan dapat melakukan verifikasi prestasi yang telah dimasukkan
8	SKPI-01-008	Lihat Daftar Prestasi	Akademik dapat melihat daftar prestasi yang telah terverifikasi
9	SKPI-01-009	Unduh Daftar Prestasi	Akademik dapat mengunduh daftar prestasi dalam format Tabel (<i>excel</i>)
10	SKPI-01-010	Logout	Mahasiswa, kemahasiswaan, dan akademik dapat keluar dari sistem

Tabel 4.3 Kebutuhan non fungsional sistem informasi perekaman prestasi mahasiswa

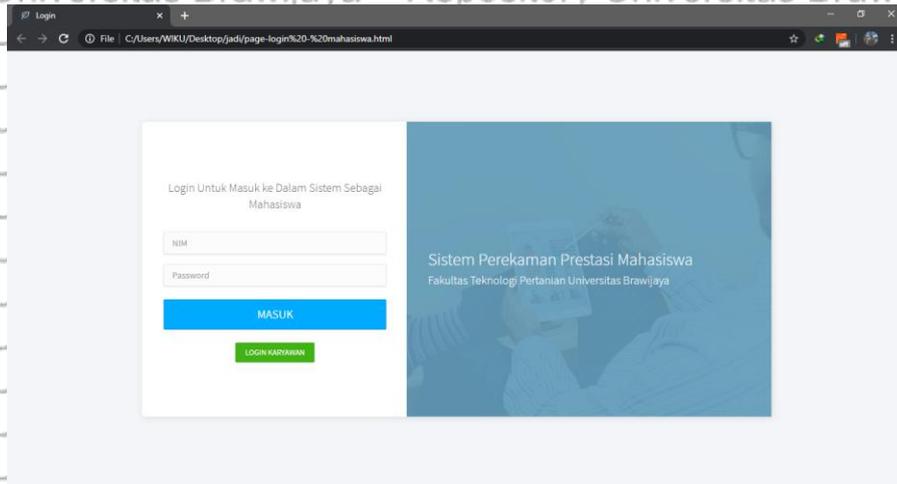
No.	Kode	Nama Fungsi	Deskripsi
1	SKPI-02-001	Usability	Mahasiswa, Kemahasiswaan, dan akademik dapat dengan mudah saat mengoperasikan sistem informasi perekaman prestasi mahasiswa

4.4 Prototype

Setelah dilakukan analisa kebutuhan dan didapatkan gambaran umum sistem, identifikasi aktor, dan daftar kebutuhan sistem. Dilakukan perancangan *user interface (prototype)* yang nantinya ditunjukkan pada ketua PSIK Fakultas Teknologi Pertanian. Pada tahap ini presentasi *prototype* yang telah dibuat sebanyak dua kali dan disetujui pada tahap ke dua.

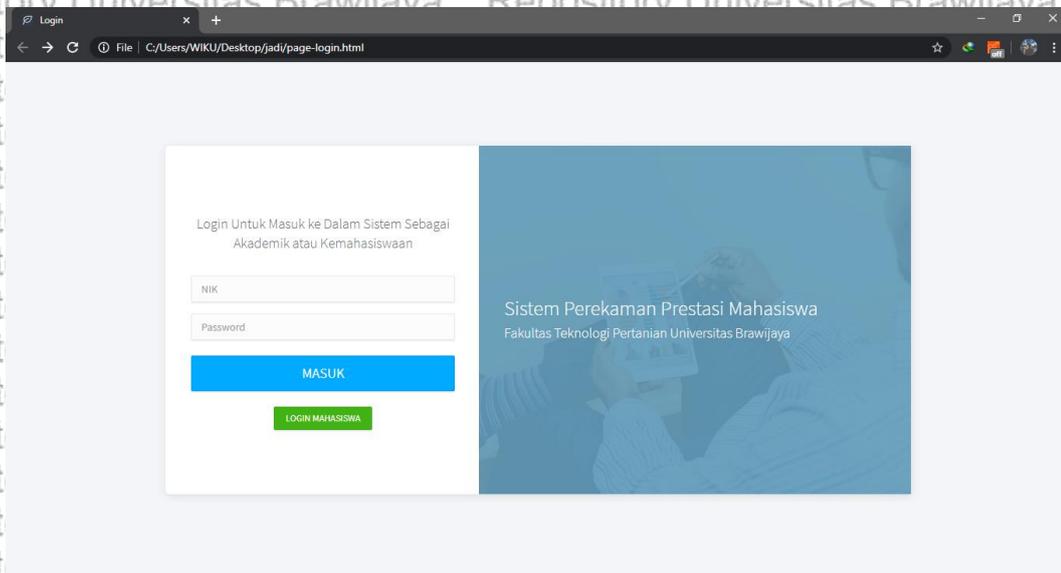
4.4.1 Presentasi Prototype Pertama

Pada presentasi *prototype* pertama, dibuat perancangan *user interface* berdasarkan analisis kebutuhan yang telah didapatkan, setelah itu dilakukan evaluasi oleh pihak PSIK Fakultas Teknologi Pertanian dari rancangan yang telah dibuat. Evaluasi dilakukan pada setiap halaman yang telah dibuat.



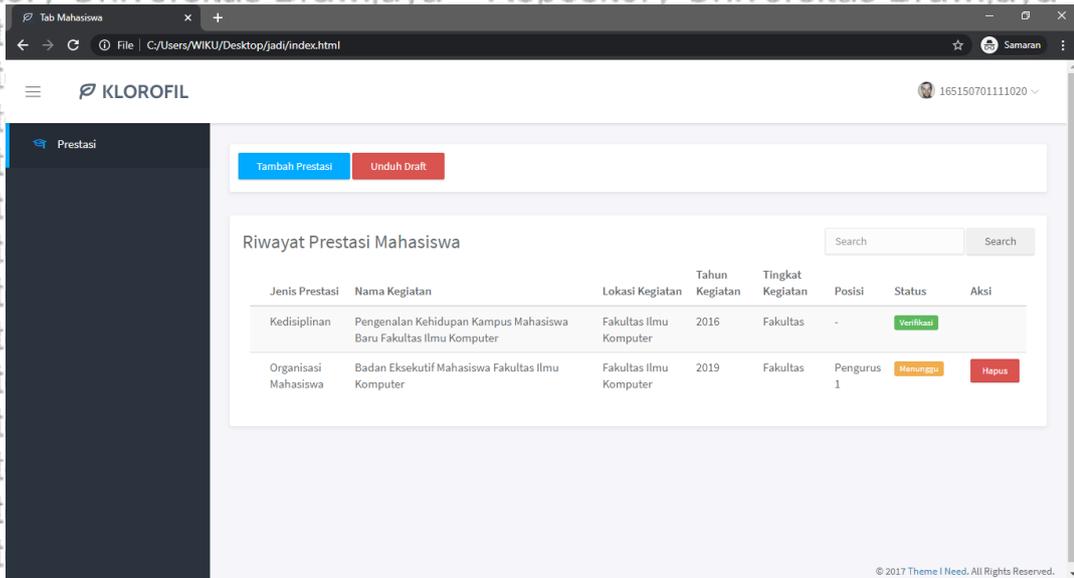
Gambar 4.2 Rancangan halaman login karyawan

Pada perancangan halaman *login* karyawan yang terdiri dari akademik dan kemahasiswaan yang ditunjukkan pada Gambar 4.2 dimana terdapat komponen seperti form untuk memasukkan NIK dan *password*, tombol untuk masuk, dan tombol *login* mahasiswa yang mengarah pada halaman *login* mahasiswa. Pada perancangan halaman *login* karyawan tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.



Gambar 4.3 Rancangan halaman login mahasiswa

Pada perancangan halaman *login* mahasiswa yang ditunjukkan pada Gambar 4.3 dimana terdapat komponen seperti form untuk memasukkan NIM dan *password*, tombol untuk masuk, dan tombol *login* karyawan yang mengarah pada halaman *login* karyawan. Pada perancangan halaman *login* karyawan tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.



Gambar 4.4 Rancangan halaman mahasiswa

Pada perancangan halaman mahasiswa yang ditunjukkan pada Gambar 4.4 dimana terdapat komponen seperti tombol untuk menampilkan form tambah prestasi dan unduh *draft*, dan Tabel yang berisikan daftar prestasi yang telah dimasukkan oleh mahasiswa yang masuk ke dalam sistem beserta statusnya. Untuk prestasi dengan status menunggu terdapat tombol hapus untuk menghapus prestasi yang telah dimasukkan mahasiswa. Pada perancangan halaman mahasiswa tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.

Tambah Prestasi

Nama Kegiatan
Nama Kegiatan

Lokasi Kegiatan
Lokasi Kegiatan

Tahun Kegiatan
Tahun Kegiatan

Tingkat Kejuaraan
Internasional

Posisi/Jabatan
Juara 1

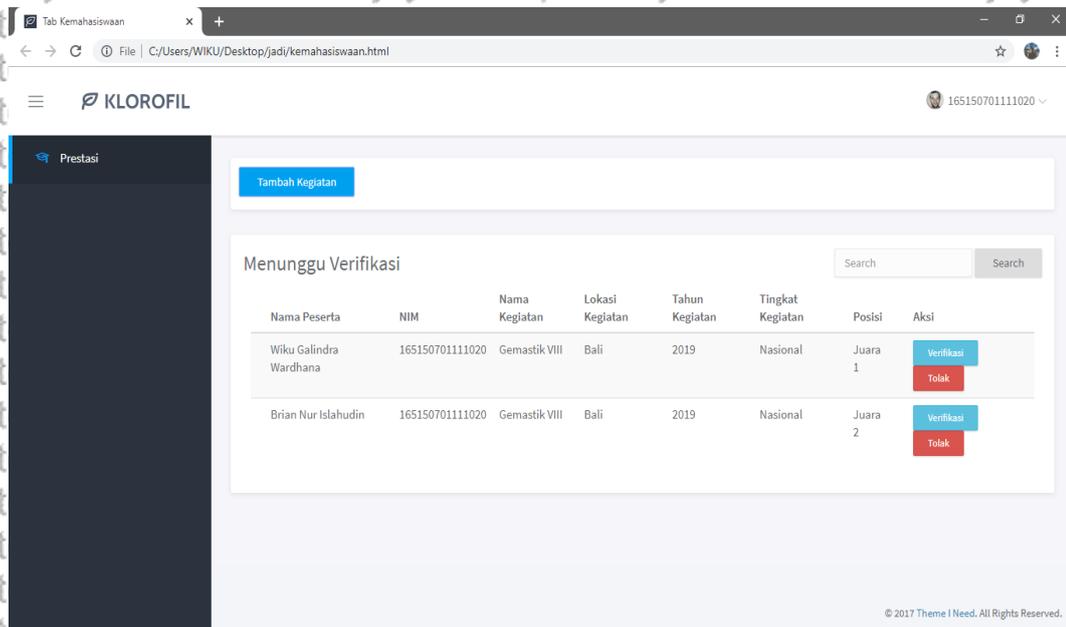
Uraian Kegiatan/Prestasi
Informasi tambahan mengenai kegiatan

File Gambar
Pilih File Tidak ada file yang dipilih

Simpan

Gambar 4.5 Rancangan form tambah prestasi

Gambar 4.5 merupakan perancangan form tambah prestasi dimana terdapat komponen *field* untuk memasukkan informasi prestasi berupa nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kejuaraan, posisi/jabatan, uraian kegiatan/prestasi, dan tombol untuk mengunggah Gambar sertifikat maupun dokumentasi kegiatan, selain itu terdapat tombol simpan. Pada perancangan form tambah prestasi terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian yaitu perlu ditambahkan masukkan untuk memilih jenis prestasi yang didapat agar prestasi yang telah dimasukkan secara langsung terbagi kedalam jenis – jenis yang telah disediakan, jenis prestasi dibagi menjadi karya ilmiah, prestasi akademik, pengabdian masyarakat, organisasi mahasiswa, religi, kesenian, olahraga, pelatihan/sertifikasi, kedisiplinan, dan lain – lain.



Gambar 4.6 Rancangan halaman kemahasiswaan

Pada perancangan halaman kemahasiswaan yang ditunjukkan pada Gambar 4.6 dimana terdapat komponen seperti tombol tambah kegiatan yang jika ditekan akan menampilkan form untuk memasukkan prestasi dengan beberapa nama mahasiswa, dan Tabel yang berisikan daftar prestasi yang telah dimasukkan oleh mahasiswa dan menunggu untuk verifikasi oleh bagian kemahasiswaan Fakultas Teknologi Pertanian. Pada tiap prestasi terdapat dua tombol yaitu tombol verifikasi untuk merubah status prestasi menjadi Verifikasi dan tombol tolak untuk menghapus prestasi. Pada perancangan halaman kemahasiswaan tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.

Gambar 4.7 merupakan perancangan form tambah kegiatan dimana terdapat komponen *field* untuk memasukkan informasi kegiatan dari beberapa mahasiswa secara langsung berupa nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kejuaraan, posisi, nama peserta, uraian kegiatan/prestasi, dan tombol untuk mengunggah Gambar sebagai tempat untuk mengunggah sertifikat maupun dokumentasi kegiatan, selain itu terdapat tombol simpan.



Pada perancangan form tambah kegiatan pada Gambar 4.7 terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian yaitu perlu ditambahkan masukkan untuk memilih jenis prestasi yang didapat agar prestasi yang telah dimasukkan secara langsung terbagi kedalam jenis – jenis yang telah disediakan, jenis prestasi dibagi menjadi karya ilmiah, prestasi akademik, pengabdian masyarakat, organisasi mahasiswa, religi, kesenian, olahraga, pelatihan/sertifikasi, kedisiplinan, dan lain – lain.

Gambar 4.7 Rancangan form tambah kegiatan

Nama	NIM	Program Studi	Jenis Prestasi	Nama Kegiatan	Lokasi Kegiatan	Tahun	Tingkat Kegiatan	Posisi
Wiku Galindra Wardhana	165150701111020	Teknologi Informasi	Prestasi Akademik	Gemastik VIII	Universitas Brawijaya	2019	Nasional	Juara 1
Brian Nur Islahudin	165150701111020	Sistem Informasi	Prestasi Akademik	Gemastik VIII	Universitas Brawijaya	2019	Nasional	Juara 1

Gambar 4.8 Rancangan halaman akademik



Pada perancangan halaman akademik yang ditunjukkan pada Gambar 4.8 dimana terdapat komponen seperti tombol unduh *excel* yang jika ditekan akan mengunduh data prestasi mahasiswa yang telah dimasukkan dan dengan status verifikasi ke dalam format *excel*, dan Tabel yang berisikan daftar prestasi yang telah dimasukkan oleh mahasiswa dan telah dilakukan verifikasi oleh bagian kemahasiswaan Fakultas Teknologi Pertanian. Tabel pada halaman akademik berisikan informasi seperti nama mahasiswa, NIM, program studi, jenis prestasi, nama kegiatan, lokasi kegiatan, tahun, tingkat kegiatan, dan posisi. Pada perancangan halaman kemahasiswaan tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.

Tingkat Kejuaraan

Internasional
Internasional
Nasional
Regional
Universitas
Fakultas
Jurusan/Program Studi
Komunitas Studi
-

Gambar 4.9 Rancangan pilihan tingkat kejuaraan

Posisi/Jabatan

Juara 1
- jika prestasi berupa perlombaan
Juara 1
Juara 2
Juara 3
Finalis
Terpilih
- jika prestasi berupa kepanitiaan/keorganisasian
Ketua
Wakil Ketua
Pengurus 1
Pengurus 2
Anggota
- lain - lain
-

Gambar 4.10 Rancangan pilihan posisi/jabatan

Pada form tambah prestasi pada Gambar 4.5 dan tambah kegiatan pada Gambar 4.7 dimana terdapat sebuah *field* yang ketika ditekan akan menampilkan beberapa pilihan sebagai masukan ketika menambah prestasi bagi mahasiswa maupun kegiatan bagi bagian kemahasiswaan. Terdapat dua pilihan yaitu tingkat kejuaraan yang ditunjukkan oleh Gambar 4.9 yang berisi pilihan tingkat kejuaraan yang diikuti berdasarkan prestasi yang akan dimasukkan dan posisi/jabatan yang ditunjukkan oleh Gambar 4.10 yang berisi pilihan dari posisi kejuaraan yang



ditambahkan atau jabatan dari kegiatan kepanitiaan atau keorganisasian yang ditambahkan. Pada perancangan pilihan tingkat kejuaraan dan pilihan posisi/jabatan tidak terdapat evaluasi dari pihak PSIK Fakultas Teknologi Pertanian.

4.4.2 Presentasi *Prototype* Kedua

Berdasarkan evaluasi yang diberikan oleh ketua PSIK Fakultas Teknologi Pertanian, dilakukan pembaharuan dari halaman *prototype* yang telah dibuat dan menghasilkan *prototype* baru. Pembaharuan dilakukan berdasarkan permintaan ketua PSIK. Pada penulisan hasil presentasi *prototype* kedua hanya ditampilkan rancangan halaman *prototype* yang sebelumnya terdapat evaluasi dan telah dilakukan perubahan.

The image shows a web form with a dropdown menu. The title of the dropdown is "Jenis Prestasi". The menu is open, showing a list of options. The first option, "Karya Ilmiah", is highlighted in blue, indicating it is the selected option. The other options listed are: Prestasi Akademik, Pengabdian Masyarakat, Organisasi Mahasiswa, Religi, Kesenian, Olahraga, Pelatihan/Sertifikasi, Kedisiplinan, and Lain-lain.

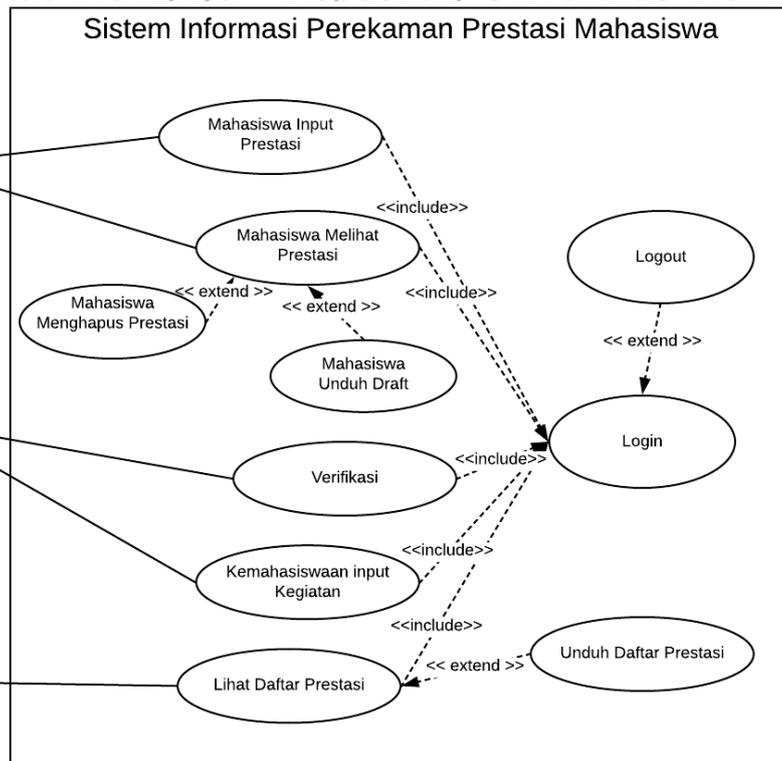
Gambar 4.11 Rancangan pilihan jenis prestasi

Pada Gambar 4.11 merupakan rancangan pilihan jenis prestasi yang menunjukkan tentang perubahan ketika melakukan tambah prestasi pada halaman mahasiswa maupun tambah kegiatan pada halaman kemahasiswaan. Pada form tambah prestasi dan kegiatan ditambahkan satu masukan baru yaitu jenis prestasi yang berbentuk *field* pilihan.

Pada presentasi kedua yang dilakukan kepada ketua PSIK Fakultas Teknologi Pertanian tidak ditemukan evaluasi lain. Hal ini menunjukkan bahwa kebutuhan *user* terhadap sistem sudah disepakati, dan hasil perancangan dari *prototype* akan digunakan sebagai acuan dalam tahap implementasi *user interface*.

4.5 Use Case Diagram

Setelah dilakukan presentasi *prototype* kepada ketua PSIK Fakultas Teknologi Pertanian, maka dilakukan pembuatan Diagram *Use Case* yang dilakukan dengan tujuan menggambarkan hubungan antara aktor dengan sistem. Pada *use case diagram* akan didapatkan fungsi pada sistem informasi perekaman prestasi mahasiswa. *Use case diagram* pada sistem informasi perekaman prestasi mahasiswa dapat dilihat pada Gambar 4.12.



Gambar 4.12 Use case diagram sistem informasi perekaman prestasi mahasiswa

4.6 Use Case Scenario

Berdasarkan *use case diagram* yang telah dibuat, maka dibuat *use case scenario* yang menerangkan urutan sederhana ketika aktor berinteraksi dengan sistem. Di tahap ini akan dijelaskan *use case scenario* dari masing – masing kebutuhan yang telah didapat berdasarkan objektif, aktor, prasyarat, alur utama, alur alternatif, dan kondisi sesudah dari sistem informasi perekaman prestasi mahasiswa.

Pada skenario *Login* menjelaskan alur dari aktor ketika akan masuk kedalam sistem sampai telah masuk kedalam halaman utama. Jika *login* gagal maka akan dikembalikan pada halaman *login*. *Use case scenario Login* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Use case scenario login

<i>Login</i> (SKPI-01-001)	
Objektif	Aktor masuk kedalam sistem
Aktor	Mahasiswa, Kemahasiswaan, dan Akademik
Prasyarat	Mahasiswa, Kemahasiswaan, dan Akademik membuka halaman <i>login</i> sistem



Alur Utama	<ol style="list-style-type: none"> 1. Mahasiswa menekan tombol login mahasiswa sedangkan Kemahasiswaan dan akademik menekan tombol login karyawan 2. Aktor memasukkan nomor induk serta <i>password</i>. 3. Sistem melakukan pengecekan terhadap nomor induk dan <i>password</i>
Alur Alternatif	Jika nomor induk dan <i>password</i> tidak sesuai maka tetap diarahkan kedalam halaman <i>login</i>
Kondisi Sesudah	Mahasiswa telah masuk kedalam sistem

Pada *use case scenario* mahasiswa tambah prestasi menjelaskan alur dari mahasiswa ketika akan memasukkan data prestasi yang didapat pada form masukan hingga prestasi telah dimasukkan dengan status menunggu. Jika form masukan belum terisi seluruhnya maka akan ada pesan untuk mengisi *field* yang masih kosong. *Use case scenario* mahasiswa Tambah prestasi dapat dilihat pada Tabel 4.5.

Tabel 4.5 Use case scenario mahasiswa tambah prestasi

Mahasiswa Tambah prestasi (SKPI-01-002)	
Objektif	Mahasiswa memasukkan data prestasi yang telah didapat
Aktor	Mahasiswa
Prasyarat	Mahasiswa telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Mahasiswa menekan tombol tambah prestasi 2. Mahasiswa menekan tombol tambah kegiatan 3. Mahasiswa memasukkan jenis prestasi 4. Mahasiswa memasukkan nama kegiatan 5. Mahasiswa memasukkan lokasi kegiatan 6. Mahasiswa memasukkan tahun kegiatan 7. Mahasiswa memasukkan tingkat kejuaraan 8. Mahasiswa memasukkan posisi. 9. Mahasiswa memasukkan peserta kegiatan. 10. Mahasiswa memasukkan uraian kegiatan. 11. Mahasiswa mengunggah bukti prestasi. 12. Sistem akan memasukkan prestasi dengan status 'menunggu'
Alur Alternatif	Jika ada salah satu <i>field</i> yang belum terisi, maka peringatan untuk mengisi <i>field</i> yang belum terisi
Kondisi Sesudah	Prestasi telah terdata dengan status 'menunggu'

Pada *use case scenario* mahasiswa lihat prestasi menjelaskan alur dari mahasiswa setelah masuk kedalam sistem dan melihat daftar prestasi yang telah



dimasukkan. *Use case scenario* mahasiswa lihat prestasi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Use case scenario mahasiswa lihat prestasi

Mahasiswa Lihat Prestasi (SKPI-01-003)	
Objektif	Mahasiswa melihat riwayat prestasi yang telah dimasukkan
Aktor	Mahasiswa
Prasyarat	Mahasiswa telah masuk kedalam sistem
Alur Utama	1. Setelah mahasiswa masuk ke dalam sistem maka akan ditampilkan Tabel riwayat prestasi yang dimunculkan
Alur Alternatif	
Kondisi Sesudah	Mahasiswa dapat melihat riwayat prestasi yang telah dimasukkan

Pada *use case scenario* mahasiswa hapus prestasi menjelaskan alur dari mahasiswa ketika akan menghapus prestasi dengan status menunggu yang telah dimasukkan hingga prestasi yang dihapus hilang dari daftar prestasi. *Use case scenario* mahasiswa hapus prestasi dapat dilihat pada Tabel 4.7.

Tabel 4.7 Use case scenario mahasiswa hapus prestasi

Mahasiswa Hapus Prestasi (SKPI-01-004)	
Objektif	Mahasiswa menghapus salah satu prestasi yang telah dimasukkan dengan status menunggu
Aktor	Mahasiswa
Prasyarat	Mahasiswa telah masuk kedalam sistem
Alur Utama	1. Mahasiswa menekan tombol hapus pada salah satu prestasi yang telah dimasukkan 2. Sistem menghapus prestasi yang telah ditekan tombol hapus
Alur Alternatif	
Kondisi Sesudah	Prestasi yang dimasukkan telah hilang dari <i>database</i>

Pada *use case scenario* mahasiswa unduh *draft* menjelaskan alur dari mahasiswa ketika akan mengunduh *draft* hingga *draft* telah terunduh. *Use case scenario* mahasiswa unduh *draft* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Use case scenario mahasiswa unduh draft

Mahasiswa Unduh Draft (SKPI-01-005)	
-------------------------------------	--



Objektif	Mahasiswa mengunduh <i>draft</i> prestasi yang telah dimasukkan dengan status verifikasi dalam format pdf
Aktor	Mahasiswa
Prasyarat	Mahasiswa telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Mahasiswa menekan tombol unduh <i>draft</i> 2. Sistem mengkonversi daftar prestasi ke dalam format pdf kemudian berkas telah terunduh
Alur Alternatif	
Kondisi Sesudah	Mahasiswa dapat mengunduh <i>draft</i> prestasi mahasiswa

Pada *use case scenario* kemahasiswaan tambah *kegiatan* menjelaskan alur dari kemahasiswaan ketika akan memasukkan data kegiatan yang didapat pada form masukan hingga kegiatan telah dimasukkan. Jika form masukan belum terisi seluruhnya maka akan ada pesan untuk mengisi *field* yang masih kosong. *Use case scenario* kemahasiswaan Tambah *kegiatan* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Use case scenario kemahasiswaan tambah kegiatan

Kemahasiswaan Tambah <i>kegiatan</i> (SKPI-01-006)	
Objektif	Kemahasiswaan memasukkan data prestasi yang telah didapat berdasarkan kegiatan.
Aktor	Kemahasiswaan
Prasyarat	Kemahasiswaan telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Kemahasiswaan menekan tombol tambah kegiatan 2. kemahasiswaan memasukkan jenis prestasi 3. Kemahasiswaan memasukkan nama kegiatan 4. Kemahasiswaan memasukkan lokasi kegiatan 5. Kemahasiswaan memasukkan tahun kegiatan. 6. Kemahasiswaan memasukkan tingkat kejuaraan 7. Kemahasiswaan memasukkan posisi. 8. Kemahasiswaan memasukkan nama peserta kegiatan. 9. Kemahasiswaan memasukkan uraian kegiatan. 10. Kemahasiswaan mengunggah bukti prestasi. 11. Sistem akan memasukkan prestasi dengan status 'verifikasi'
Alur Alternatif	Jika ada salah satu <i>field</i> yang belum terisi, maka peringatan untuk mengisi <i>field</i> yang belum terisi
Kondisi Sesudah	Prestasi yang dimasukkan berstatus 'verifikasi'



Pada *use case scenario* verifikasi menjelaskan alur dari kemahasiswaan ketika akan melakukan verifikasi terhadap salah satu prestasi dengan status menunggu yang telah dimasukkan hingga prestasi yang dilakukan verifikasi statusnya berubah menjadi verifikasi. *Use case scenario* mahasiswa verifikasi prestasi dapat dilihat pada Tabel 4.10.

Tabel 4.10 Use case scenario verifikasi

Verifikasi (SKPI-01-007)	
Objektif	Kemahasiswaan melakukan verifikasi prestasi yang telah dimasukkan.
Aktor	Kemahasiswaan
Prasyarat	Kemahasiswaan telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Kemahasiswaan menekan tombol verifikasi 2. Kemahasiswaan menekan tombol verifikasi pada Tabel prestasi 3. Sistem merubah status prestasi dari menunggu verifikasi menjadi terverifikasi
Alur Alternatif	<ol style="list-style-type: none"> 1. Kemahasiswaan menekan tombol tolak pada Tabel prestasi 2. Sistem menghapus prestasi yang ditekan tombol tolak
Kondisi Sesudah	Status Prestasi menjadi terverifikasi atau data prestasi terhapus

Pada *use case scenario* akademik lihat prestasi menjelaskan alur dari akademik setelah masuk kedalam sistem dan melihat daftar prestasi. *Use case scenario* lihat daftar prestasi dapat dilihat pada Tabel 4.11.

Tabel 4.11 Use case scenario lihat daftar prestasi

Lihat Daftar Prestasi (SKPI-01-008)	
Objektif	Akademik melihat daftar prestasi yang telah terverifikasi.
Aktor	Akademik
Prasyarat	Akademik telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Akademik menekan tombol tampilkan 2. Sistem menampilkan daftar prestasi yang ada pada <i>database</i>
Alur Alternatif	
Kondisi Sesudah	Akademik dapat melihat prestasi mahasiswa



Pada *use case scenario* unduh daftar prestasi menjelaskan alur dari akademik ketika akan mengunduh daftar prestasi hingga daftar prestasi telah terunduh. *Use case scenario* unduh daftar prestasi dapat dilihat pada Tabel 4.12.

Tabel 4.12 Use case scenario unduh daftar prestasi

Unduh Daftar Prestasi (SKPI-01-009)	
Objektif	Akademik mengunduh daftar prestasi yang telah terverifikasi.
Aktor	Akademik
Prasyarat	Akademik telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Akademik menekan tombol unduh prestasi 2. Sistem mengkonversi daftar prestasi ke dalam format excel kemudian berkas telah terunduh
Alur Alternatif	-
Kondisi Sesudah	Akademik dapat mengunduh prestasi mahasiswa

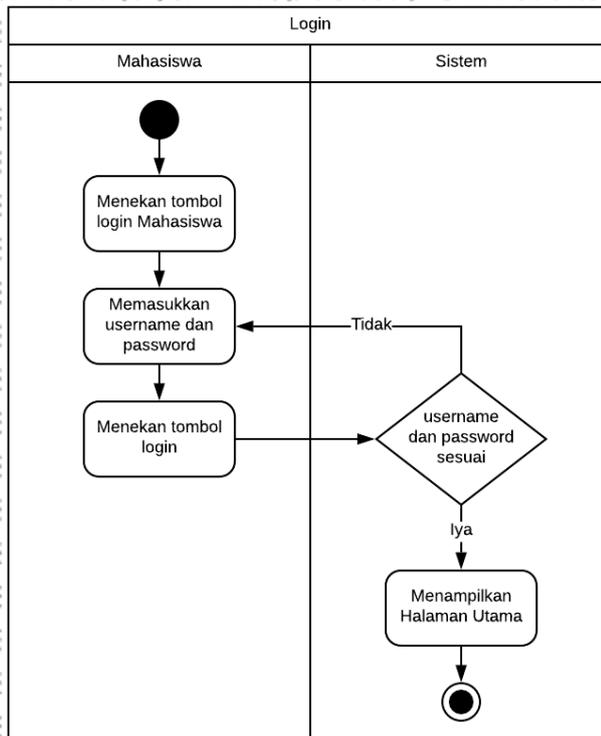
Pada *use case scenario logout* menjelaskan alur dari aktor ketika akan keluar dari sistem. *Use case scenario logout* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Use case scenario logout

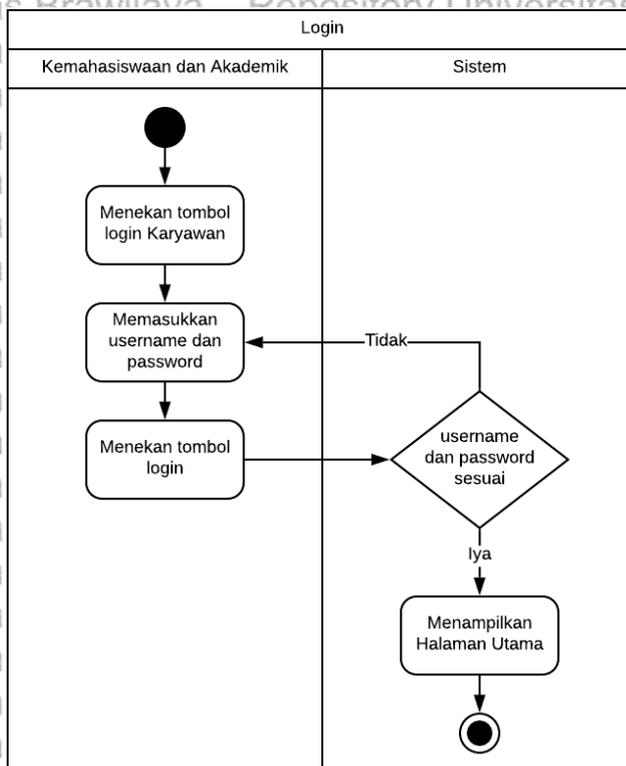
Logout (SKPI-01-010)	
Objektif	Aktor keluar dari sistem.
Aktor	Mahasiswa, Kemahasiswaan, dan Akademik
Prasyarat	Aktor telah masuk kedalam sistem
Alur Utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>user</i> 2. Aktor menekan tombol <i>logout</i> 3. Sistem mengeluarkan aktor dari sistem dan menampilkan halaman <i>login</i>
Alur Alternatif	-
Kondisi Sesudah	Sistem mengeluarkan aktor dari sistem dan menampilkan halaman <i>login</i>

4.7 Activity diagram

Activity diagram digunakan untuk menjelaskan langkah dari masing – masing *use case*. Pada diagram ini dapat dilihat bagaimana sebuah fungsi berjalan pada sistem dari mulai hingga sistem berakhir pada sistem informasi perekaman prestasi mahasiswa.



Gambar 4.13 Activity diagram login mahasiswa



Gambar 4.14 Activity diagram login karyawan

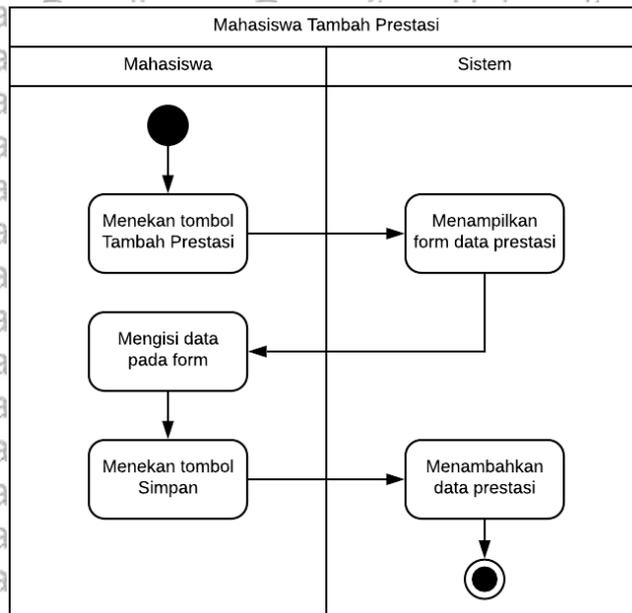
Activity diagram login mahasiswa pada Gambar 4.13 menjelaskan tentang alur dari aktor mahasiswa ketika melakukan login. Dimulai dengan menekan tombol login mahasiswa kemudian memasukkan nomor induk dan password dan



menekan tombol *login*, setelah itu dilakukan pengecekan oleh sistem apakah nomor induk dan *password* sesuai, jika tidak sesuai maka akan aktor akan diminta untuk memasukkan nomor induk dan *password* ulang, jika sesuai maka akan ditampilkan halaman mahasiswa.

Activity diagram login karyawan pada Gambar 4.14 menjelaskan tentang alur dari aktor kemahasiswaan dan akademik ketika melakukan *login*. Dimulai dengan menekan tombol login karyawan kemudian memasukkan nomor induk dan *password* dan menekan tombol *login*, setelah itu dilakukan pengecekan oleh sistem apakah nomor induk dan *password* sesuai, jika tidak sesuai maka akan aktor akan diminta untuk memasukkan nomor induk dan *password* ulang, jika sesuai maka akan ditampilkan halaman kemahasiswaan untuk aktor kemahasiswaan dan halaman akademik untuk aktor akademik.

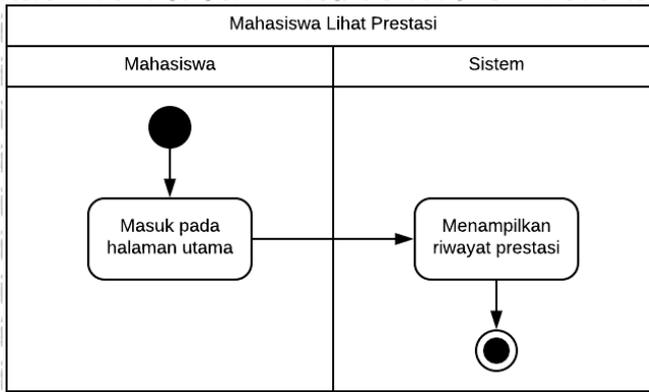
Activity diagram mahasiswa tambah prestasi pada Gambar 4.15 menjelaskan tentang alur dari mahasiswa ketika menambah prestasi. Dimulai dengan menekan tombol tambah prestasi setelah itu sistem akan menampilkan form data prestasi, setelah mahasiswa memasukkan data pada form maka mahasiswa menekan tombol simpan dan prestasi akan ditambahkan oleh sistem.



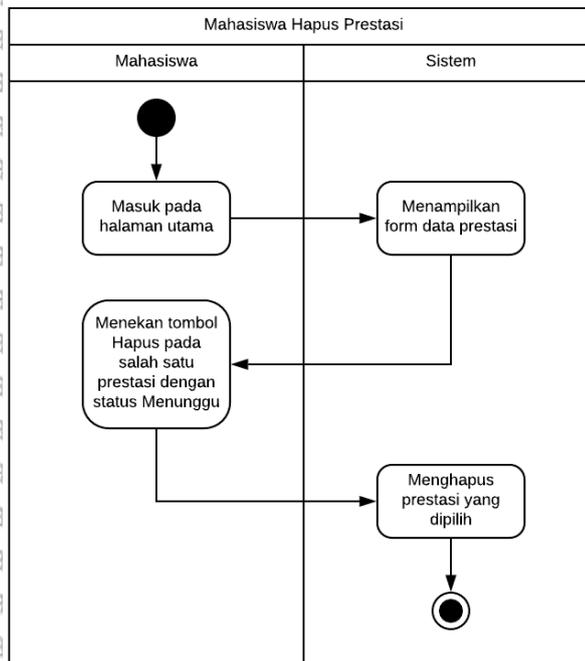
Gambar 4.15 Activity diagram mahasiswa tambah prestasi

Activity diagram mahasiswa lihat prestasi pada Gambar 4.16 menjelaskan tentang alur dari mahasiswa ketika melihat prestasi. Ketika mahasiswa telah melakukan *login* dan masuk ke dalam halaman utama mahasiswa, maka akan ditampilkan data prestasi berdasarkan nomor induk mahasiswa.

Activity diagram mahasiswa hapus prestasi pada Gambar 4.17 menjelaskan tentang alur dari mahasiswa ketika menghapus prestasi. Ketika mahasiswa telah masuk ke dalam halaman mahasiswa, mahasiswa menekan tombol hapus pada salah satu prestasi yang ditampilkan. Setelah itu sistem akan menghapus prestasi tersebut dari *database*.



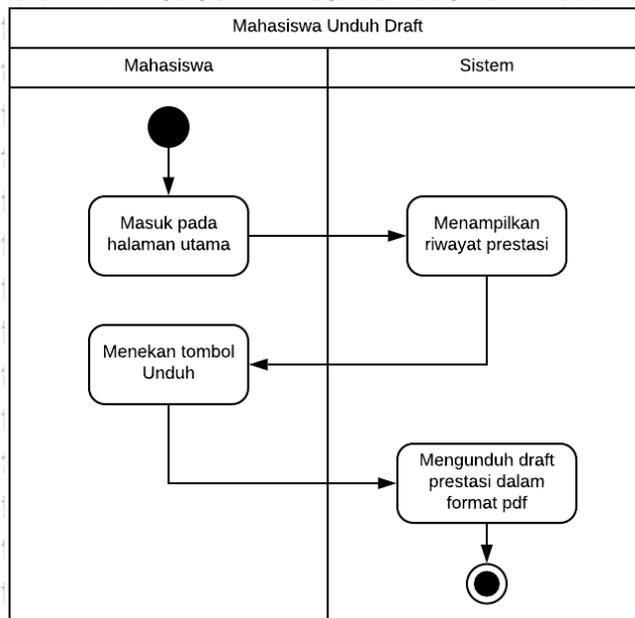
Gambar 4.16 Activity diagram mahasiswa lihat prestasi



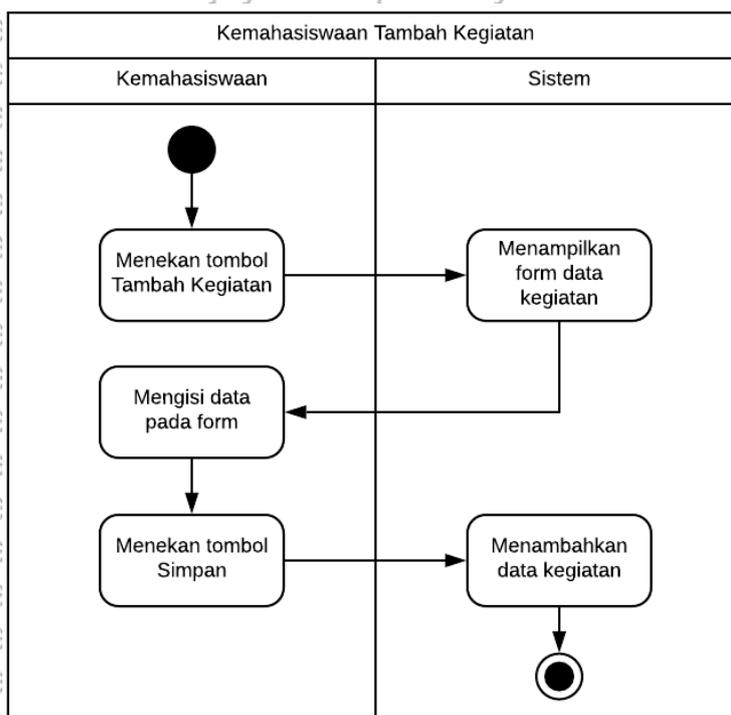
Gambar 4.17 Activity diagram mahasiswa hapus prestasi

Activity diagram mahasiswa tambah prestasi pada Gambar 4.18 menjelaskan tentang alur dari mahasiswa ketika mengunduh prestasi. Ketika mahasiswa telah masuk ke dalam halaman utama, mahasiswa menekan tombol unduh, setelah itu sistem akan mengunduh *draft* berisi riwayat prestasi mahasiswa dalam format *file pdf*.

Activity diagram mahasiswa tambah kegiatan pada Gambar 4.19 menjelaskan tentang alur dari kemahasiswaan ketika menambah kegiatan. Dimulai dengan menekan tombol tambah kegiatan, setelah itu sistem akan menampilkan form data kegiatan, setelah itu kemahasiswaan mengisi data pada form dan setelah itu kemahasiswaan menekan tombol simpan maka sistem akan mengirim data ke *database*.

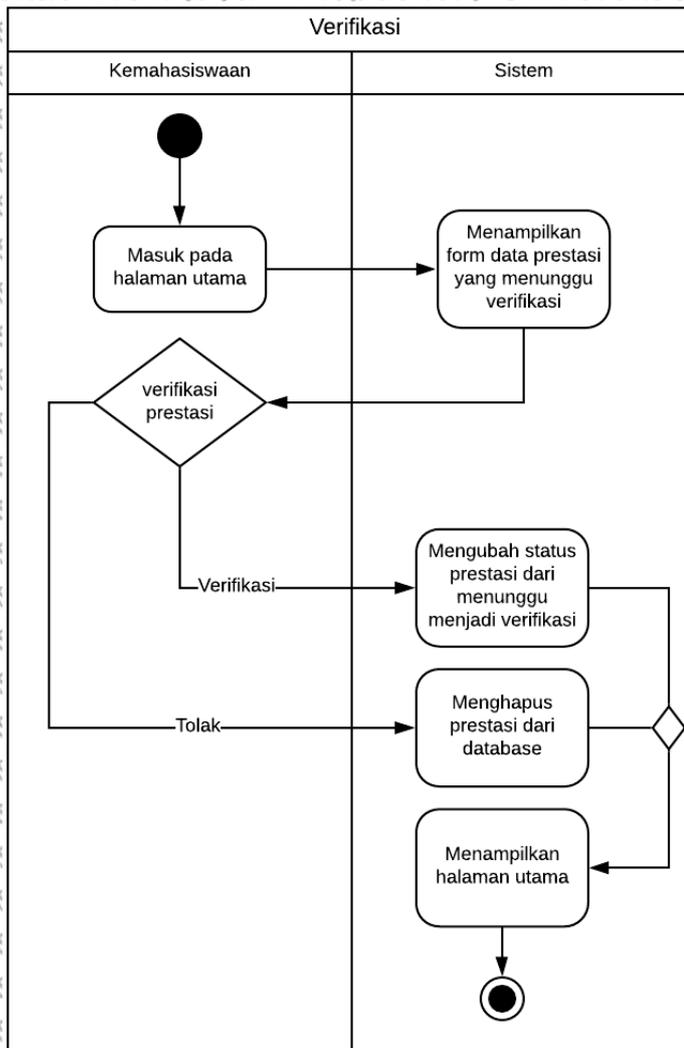


Gambar 4.18 Activity diagram mahasiswa unduh draft

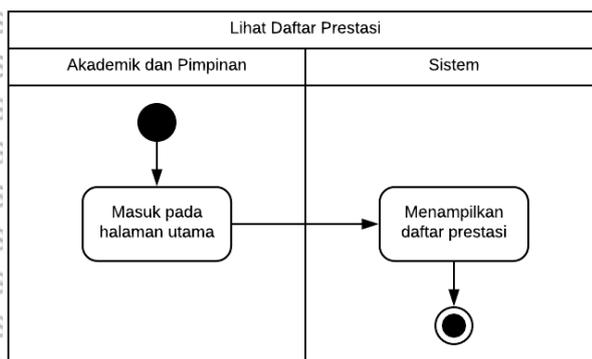


Gambar 4.19 Activity diagram kemahasiswaan tambah kegiatan

Activity diagram verifikasi pada Gambar 4.20 menjelaskan tentang alur dari kemahasiswaan ketika melakukan verifikasi prestasi. Ketika kemahasiswaan telah masuk ke dalam halaman utama, maka sistem akan memberikan dua pilihan yaitu verifikasi dan tolak, jika kemahasiswaan menekan tombol verifikasi maka sistem akan merubah status prestasi menjadi verifikasi dan jika kemahasiswaan menekan tolak maka sistem akan menghapus prestasi dari *database*.

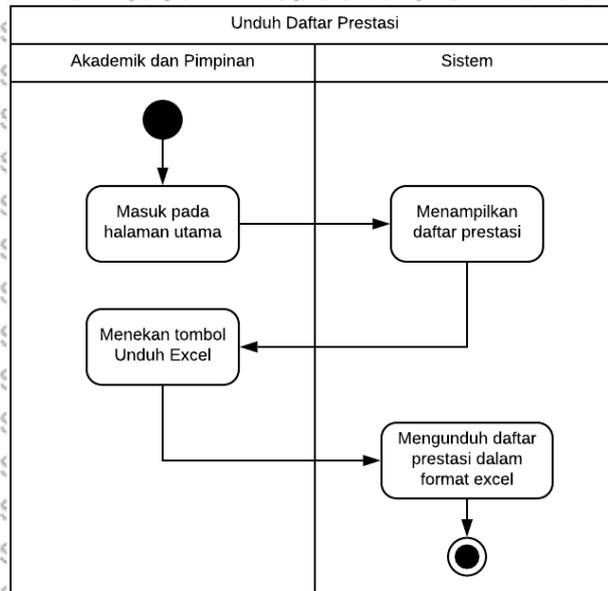


Gambar 4.20 Activity diagram verifikasi



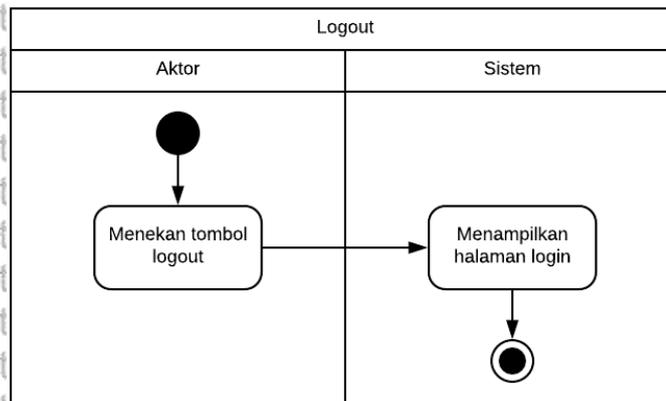
Gambar 4.21 Activity diagram lihat daftar prestasi

Activity diagram mahasiswa lihat prestasi pada Gambar 4.21 menjelaskan tentang alur dari akademik ketika melihat prestasi. Ketika akademik telah melakukan login dan masuk ke dalam halaman utama akademik, maka akan ditampilkan data prestasi mahasiswa.



Gambar 4.22 Activity diagram unduh daftar prestasi

Activity diagram unduh daftar prestasi pada Gambar 4.22 menjelaskan tentang alur dari akademik ketika melakukan unduh prestasi. Ketika kemahasiswaan telah masuk ke dalam halaman utama, akademik menekan tombol unduh *excel* kemudian sistem mengunduh *file* dengan format *excel* berisi prestasi dengan status verifikasi.



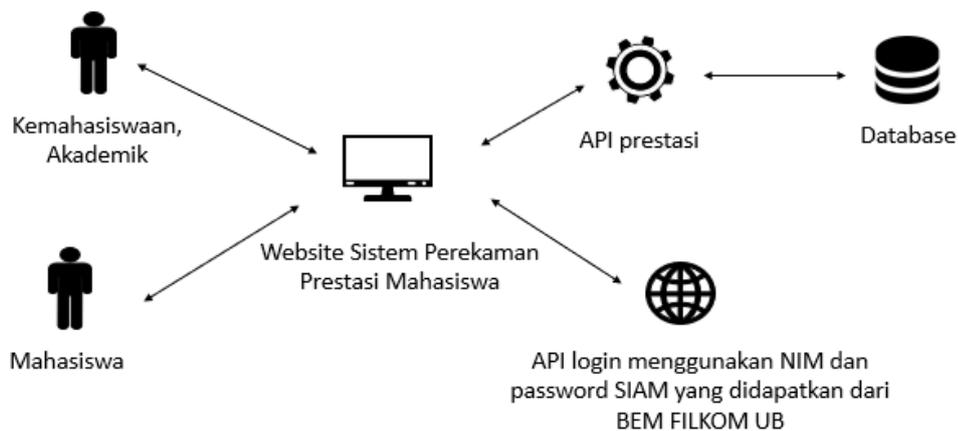
Gambar 4.23 Activity diagram logout

Activity diagram logout pada Gambar 4.23 menjelaskan tentang alur dari aktor yang terdiri dari mahasiswa, kemahasiswaan dan akademik ketika melakukan *logout* dari sistem informasi perekaman prestasi mahasiswa dimulai dengan menekan tombol *logout* kemudian sistem akan menampilkan halaman *login*.

BAB 5 PERANCANGAN SISTEM

5.1 Arsitektur sistem

Sistem informasi perekaman prestasi mahasiswa dikembangkan dalam platform *website* dengan memanfaatkan teknologi *Restful web service*, yang mana dalam perpindahan data dilakukan melalui perantara berupa REST API bernama API prestasi. Sistem melakukan *request* kepada API prestasi yang kemudian API prestasi melakukan *request* kepada database berdasarkan *request* dari *web client*. Selain itu sistem juga memanfaatkan REST API *login* menggunakan Nomor Induk Mahasiswa dan password SIAM yang didapatkan dari BEM Filkom UB. Untuk arsitektur sistem informasi perekaman prestasi mahasiswa dapat dilihat pada gambar 5.1.



Gambar 5.1 Arsitektur sistem informasi perekaman prestasi mahasiswa

Pada gambar 5.1 terdapat 2 aktor yaitu aktor karyawan yang terdiri dari kemahasiswaan dan akademik dan juga aktor mahasiswa, yang mana aktor mahasiswa memanfaatkan API login dengan menggunakan NIM dan *password* SIAM untuk autentikasi dan mendapatkan data mahasiswa. Untuk aktor kemahasiswaan dan akademik autentikasi dilakukan berdasarkan data yang ada pada *database*. Untuk setiap *request* yang dilakukan kepada *website* sistem informasi maka akan dilakukan *request* dari *web client* kepada API prestasi dan API prestasi akan melakukan *request* pada *database*. *Response* dari *database* akan dikirimkan kepada API prestasi dan diteruskan kepada *web client* sehingga bisa ditampilkan kepada aktor.

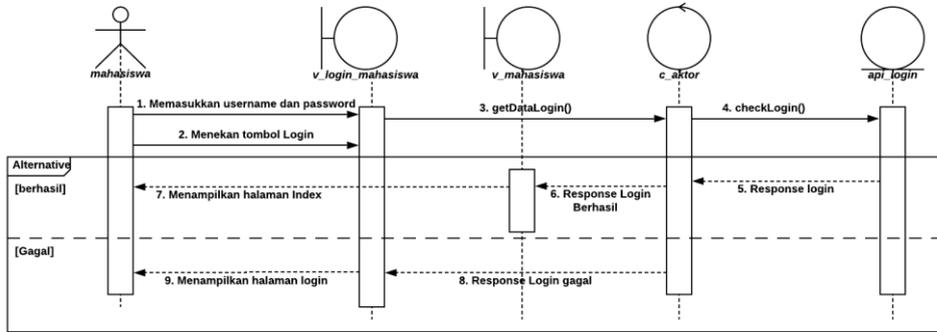
5.2 Sequence diagram

Sequence diagram menjelaskan urutan dari langkah yang dilakukan sistem sebagai respon dari sebuah fungsi yang menghasilkan sebuah *output*. Dimulai dengan pemicu aktivitas terkait, proses yang digambarkan dengan perubahan

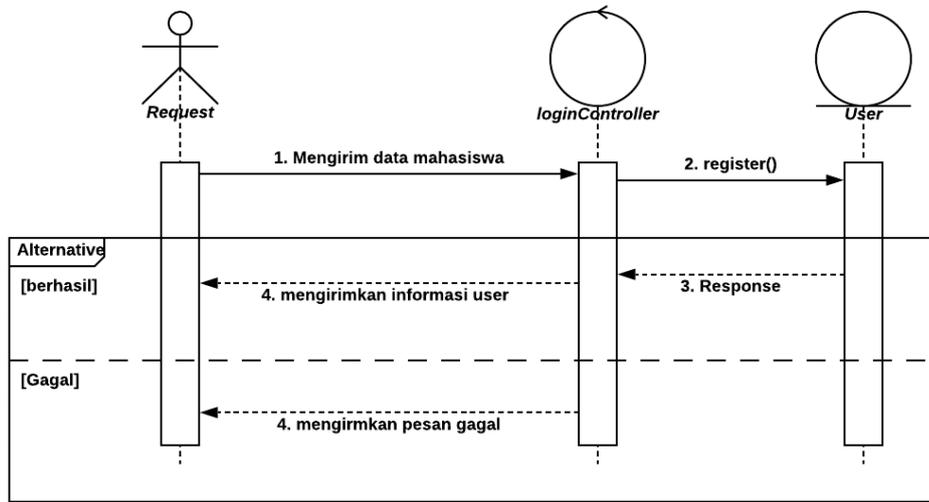


yang terjadi hingga *output* yang dihasilkan. *Sequence diagram* dirancang berdasarkan hasil dari *use case scenario* dan *activity diagram*.

5.2.1 Sequence Diagram Login



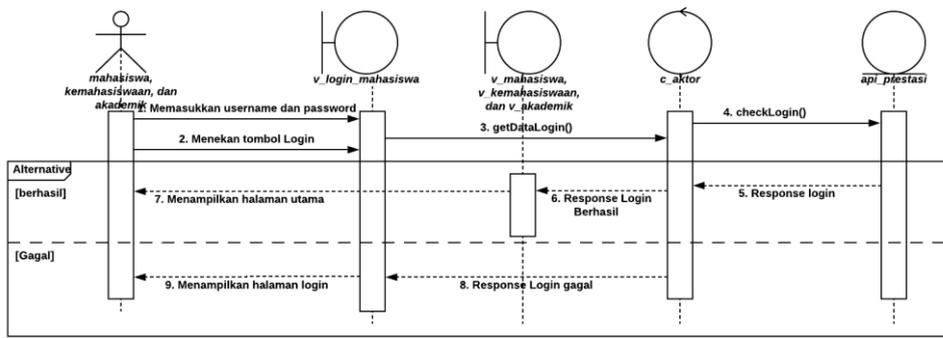
Gambar 5.2 Sequence diagram login mahasiswa



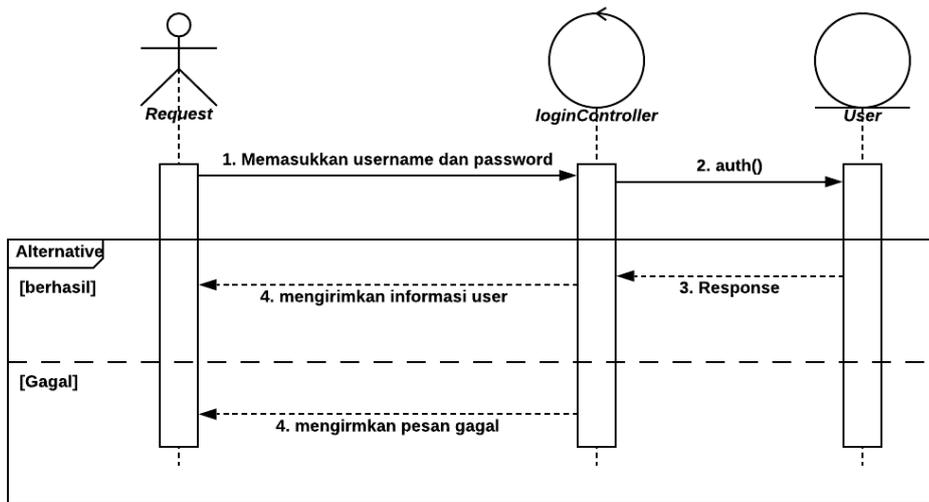
Gambar 5.3 Sequence diagram login mahasiswa pada API prestasi

Pada Gambar 5.2 terdapat *sequence diagram login* mahasiswa yang menjelaskan alur dari aktor mahasiswa ketika masuk kedalam sistem informasi perekaman prestasi mahasiswa. Aktor memasukkan nomor induk dan *password* kemudian dilakukan pengecekan pada *API login*. Jika *login* gagal maka akan tetap dikembalikan pada halaman *login* sedangkan jika berhasil diteruskan pada halaman *v_mahasiswa*.

Pada Gambar 5.3 menunjukkan *sequence diagram login* mahasiswa pada API prestasi. Setelah masukan dikirim ke API, maka akan dilakukan *create user* pada *database* yang direpresentasikan pada *model User* dengan informasi mahasiswa yang didapat dari *API login* ketika memasukkan nomor induk dan *password*. Jika berhasil maka data *user* akan dikirimkan dalam format JSON, jika gagal maka pesan "gagal" akan dikirimkan dalam format JSON.



Gambar 5.4 Sequence diagram login karyawan



Gambar 5.5 Sequence diagram login karyawan pada API prestasi

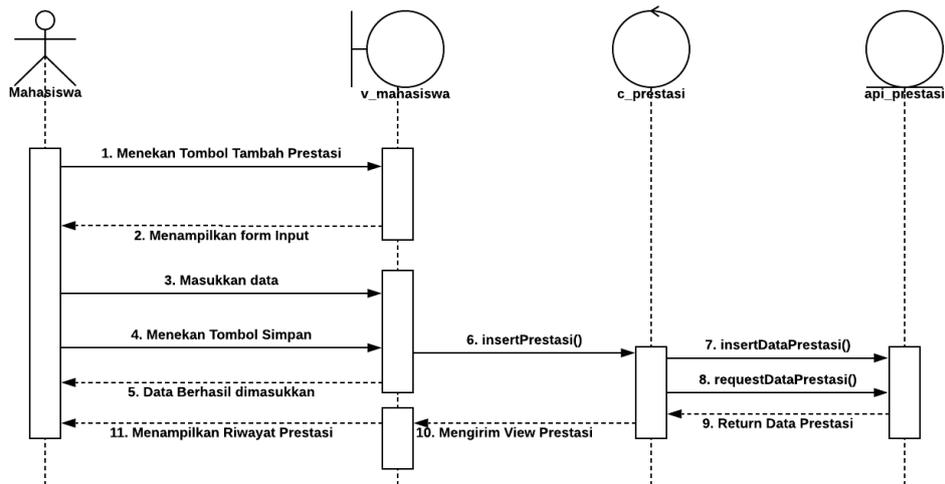
Pada Gambar 5.4 terdapat *Sequence diagram login* karyawan yang menjelaskan alur dari aktor baik kemahasiswaan dan akademik ketika masuk kedalam sistem informasi perekaman prestasi mahasiswa. Aktor memasukkan nomor induk dan *password* kemudian dilakukan pengecekan pada *API prestasi*. Jika *login* gagal maka akan tetap dikembalikan pada halaman *login* sedangkan jika berhasil diteruskan pada halaman *v_kemahasiswaan* untuk kemahasiswaan dan *v_akademik* untuk akademik.

Sequence diagram pada Gambar 5.5 merupakan *sequence diagram login* pada *API prestasi*. Setelah masukan dikirim melalui *API*, maka akan dilakukan pengecekan kesesuaian antara masukan dengan user yang ada pada *database* yang direpresentasikan pada *model User*. Jika sesuai maka akan dikirimkan informasi *user* dalam format *JSON* dan jika tidak, maka akan dikirimkan pesan bahwa nomor induk atau *password* tidak sesuai.

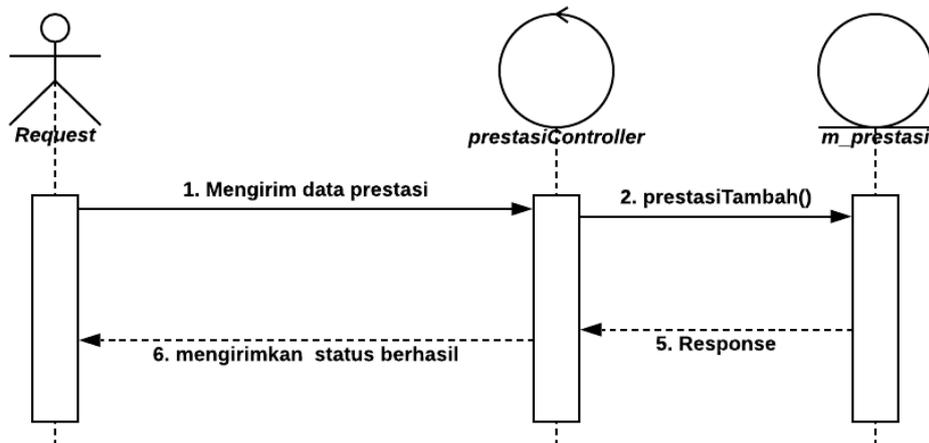


5.2.2 Sequence Diagram Mahasiswa Tambah prestasi

Sequence diagram mahasiswa Tambah prestasi menjelaskan alur dari mahasiswa ketika memasukkan prestasi kedalam sistem informasi perekaman prestasi mahasiswa. Mahasiswa menekan tombol tambah prestasi dan muncul form Input, kemudian mahasiswa memasukkan data prestasi dan menekan tombol simpan. Data akan dikirimkan pada database melalui c_prestasi dan kemudian ditampilkan halaman v_mahasiswa dan otomatis prestasi bertambah. Sequence diagram mahasiswa Tambah prestasi bisa dilihat pada Gambar 5.6.



Gambar 5.6 Sequence diagram mahasiswa tambah prestasi



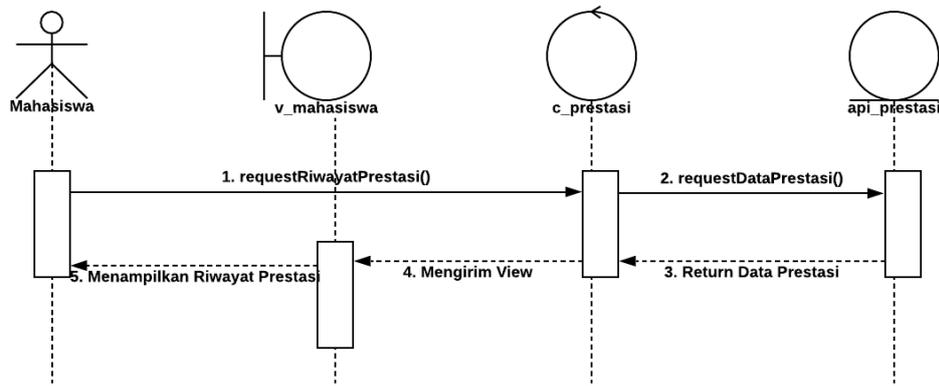
Gambar 5.7 Sequence diagram mahasiswa tambah prestasi pada API prestasi

Pada Gambar 5.7 menunjukkan sequence diagram tambah prestasi pada API prestasi. Data prestasi yang telah dimasukkan akan dikirim melewati prestasiController kemudian di kirim kepada database yang direpresentasikan pada model m_prestasi. Setelah data dimasukkan ke dalam database maka akan diberikan sebuah response dalam bentuk JSON yang berisi pesan status berhasil.

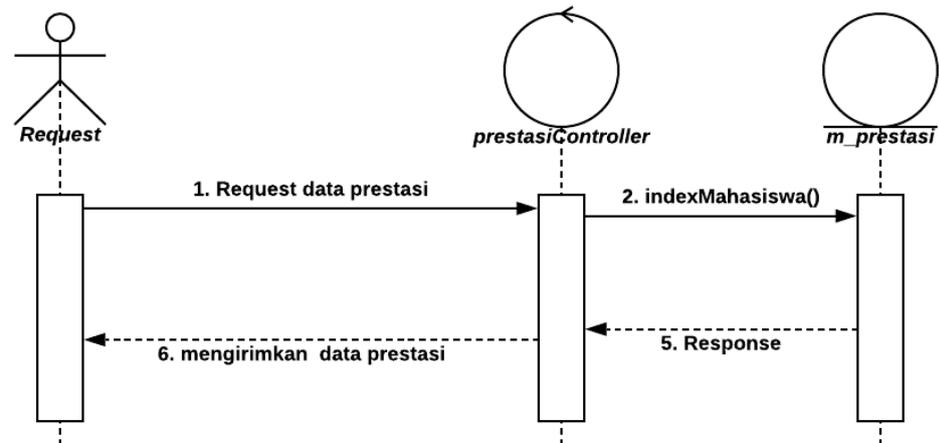


5.2.3 Sequence Diagram Mahasiswa Lihat Prestasi

Sequence diagram mahasiswa lihat prestasi menjelaskan alur menampilkan prestasi kedalam sistem informasi perekaman prestasi mahasiswa. Setelah masuk ke dalam sistem dengan login sebagai mahasiswa, sistem akan otomatis mengarahkan kepada halaman v_mahasiswa. Sequence diagram mahasiswa lihat prestasi bisa dilihat pada Gambar 5.8.



Gambar 5.8 Sequence diagram mahasiswa lihat prestasi



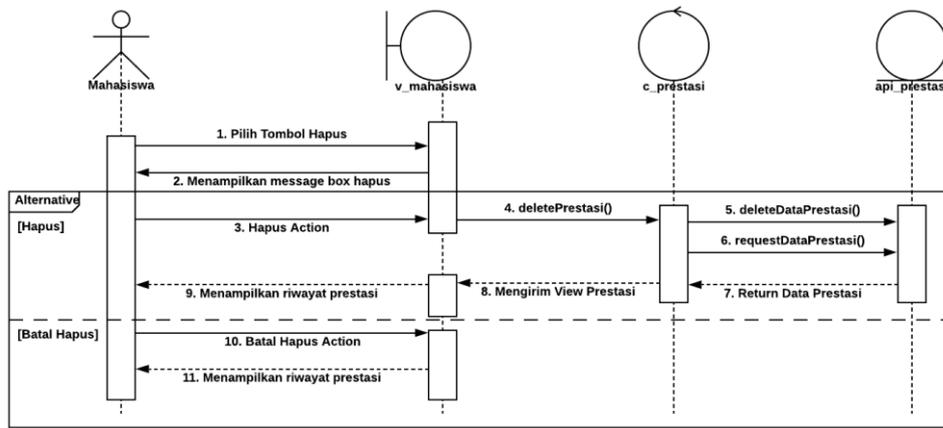
Gambar 5.9 Sequence diagram mahasiswa lihat prestasi pada API prestasi

Pada Gambar 5.9 menunjukkan sequence diagram lihat prestasi pada API prestasi. Dilakukan request data prestasi berdasarkan no_induk mahasiswa yang telah masuk ke dalam sistem. Request dilakukan melewati prestasiController kemudian meminta data pada database yang direpresentasikan pada model m_prestasi. Setelah itu akan diberikan response data prestasi dalam bentuk JSON.

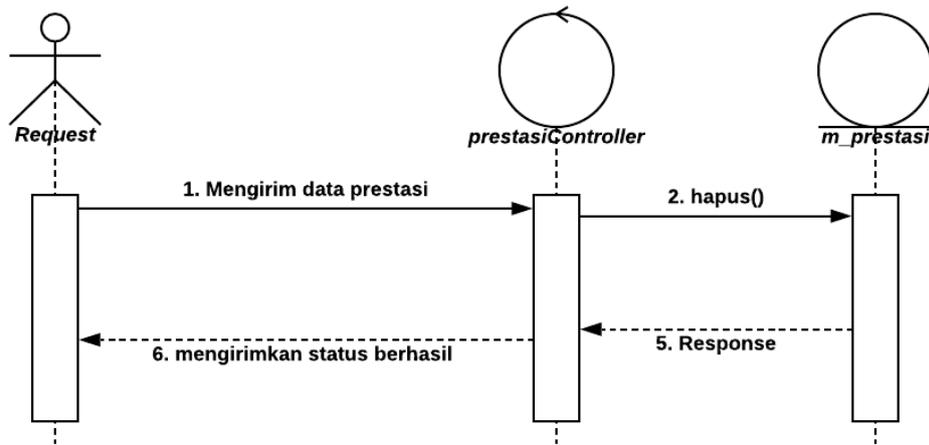
5.2.4 Sequence Diagram Mahasiswa Hapus Prestasi

Pada Gambar 5.10 terdapat sequence diagram mahasiswa hapus prestasi yang menjelaskan alur penghapusan prestasi dengan status menunggu pada sistem informasi perekaman prestasi mahasiswa. Mahasiswa menekan tombol hapus

pada salah satu prestasi dengan status menunggu, setelah itu muncul *message box* hapus dan jika pilih hapus maka data akan dikirim ke *c_prestasi* dan dihapus dari *database* setelah itu menampilkan halaman *v_mahasiswa*, jika pilih *cancel* maka akan dikembalikan pada halaman *v_mahasiswa*.



Gambar 5.10 Sequence diagram mahasiswa hapus prestasi



Gambar 5.11 Sequence diagram mahasiswa hapus prestasi pada API prestasi

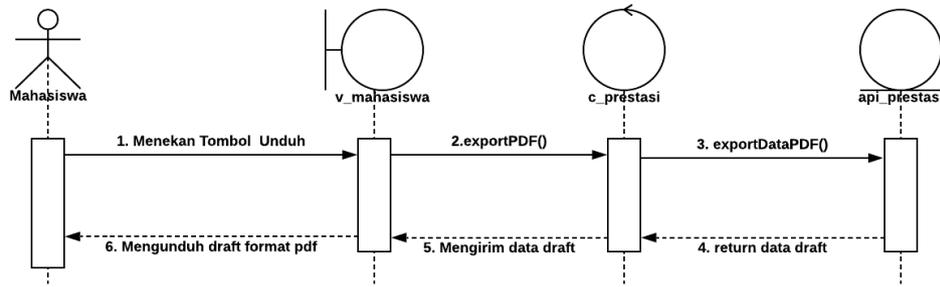
Pada Gambar 5.11 menunjukkan *sequence diagram* hapus prestasi pada API prestasi. Prestasi yang dipilih akan dikirim melalui *prestasiController* setelah dihapus pada *database* yang direpresentasikan pada *model m_prestasi*, setelah penghapusan berhasil maka akan diberikan *response* berupa JSON dengan pesan status berhasil.

5.2.5 Sequence Diagram Mahasiswa Unduh Draft

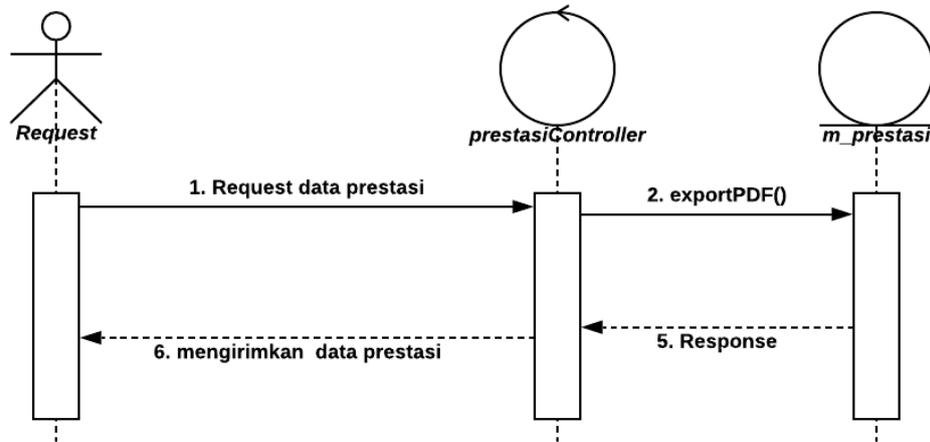
Sequence diagram mahasiswa unduh *draft* menjelaskan alur unduh prestasi mahasiswa yang telah dimasukkan pada sistem informasi perekaman prestasi mahasiswa. Mahasiswa menekan tombol unduh setelah itu setelah itu dilakukan *request* untuk meminta data prestasi mahasiswa dan dikembalikan dalam bentuk



file berformat pdf. *Sequence diagram* mahasiswa unduh draft bisa dilihat pada Gambar 5.12.



Gambar 5.12 *Sequence diagram* mahasiswa unduh draft



Gambar 5.13 *Sequence diagram* mahasiswa unduh draft pada API prestasi

Pada Gambar 5.13 menunjukkan *sequence diagram* unduh draft pada API prestasi. Dilakukan request data prestasi dengan status verifikasi dari mahasiswa yang sedang masuk ke dalam sistem pada database yang direpresentasikan oleh model m_prestasi melalui controller prestasiController. Setelah itu response data prestasi akan diberikan dalam format JSON.

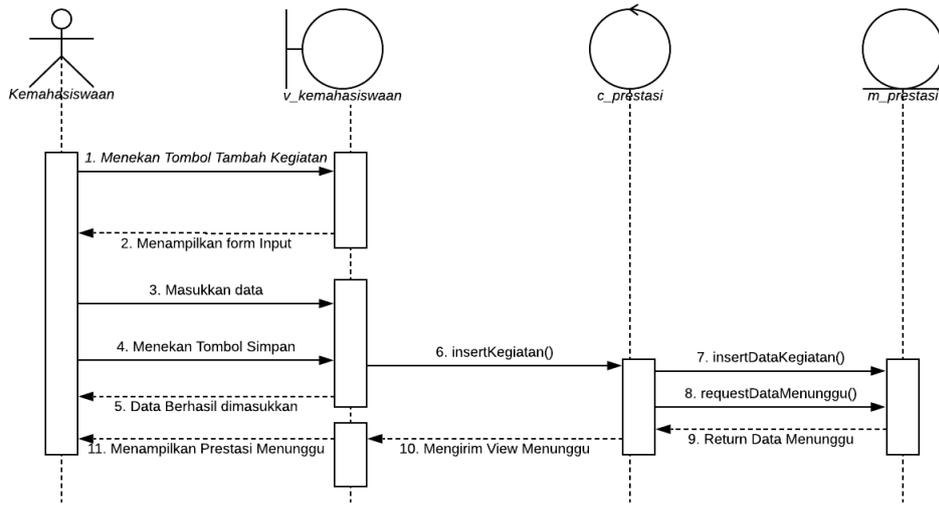
5.2.6 *Sequence Diagram* Kemahasiswaan Tambah Kegiatan

pada Gambar 5.14 terdapat *sequence diagram* kemahasiswaan tambah kegiatan yang menjelaskan alur dari aktor ketika memasukkan kegiatan kedalam sistem informasi perekaman prestasi mahasiswa. Kemahasiswaan menekan tombol tambah kegiatan dan muncul form Input, kemudian mahasiswa memasukkan data kegiatan dan menekan tombol simpan. Data akan dikirimkan pada database melalui c_prestasi dan kemudian menampilkan pesan data berhasil dimasukkan dan halaman v_kemahasiswaan.

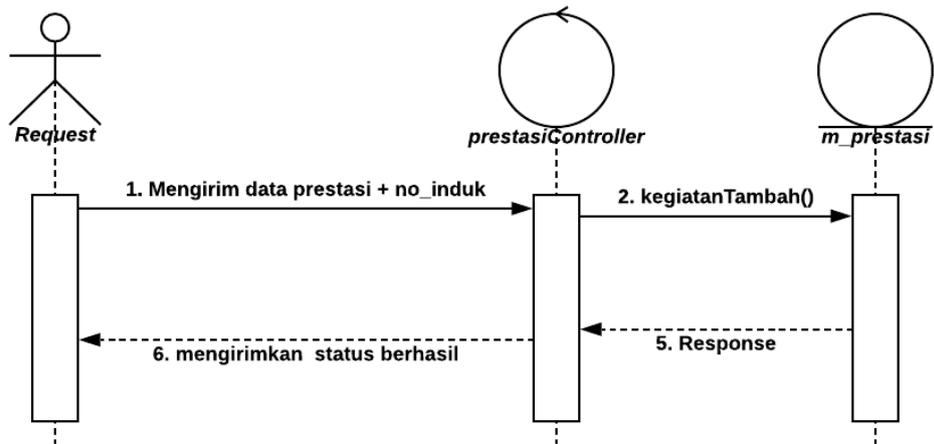
Pada Gambar 5.15 menunjukkan *sequence diagram* tambah prestasi pada API prestasi. Data prestasi yang telah dimasukkan akan dikirim melewati prestasiController kemudian di kirim kepada database yang direpresentasikan



pada *model m_prestasi*. Setelah data prestasi dikirim ke *database*, maka akan diberikan sebuah *response* dalam bentuk JSON yang berisi pesan status berhasil.



Gambar 5.14 Sequence diagram kemahasiswaan tambah kegiatan



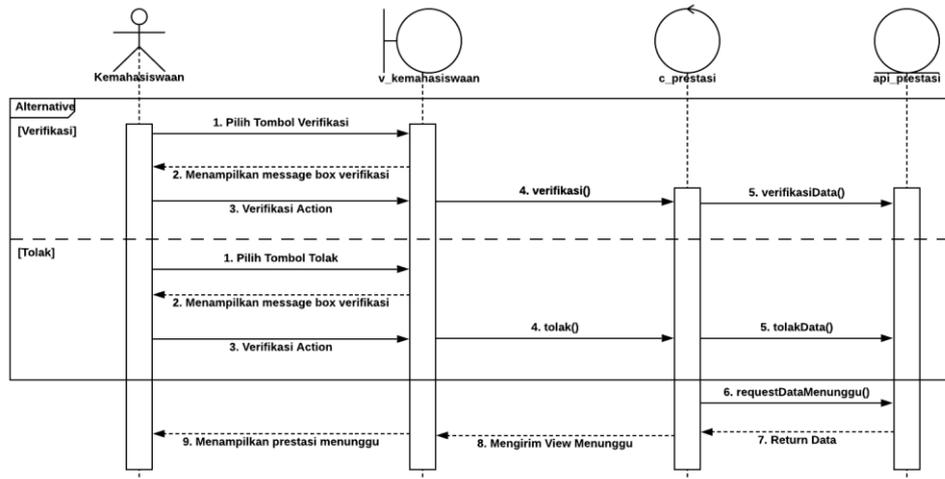
Gambar 5.15 Sequence diagram kemahasiswaan tambah kegiatan pada API prestasi

5.2.7 Sequence Diagram Verifikasi

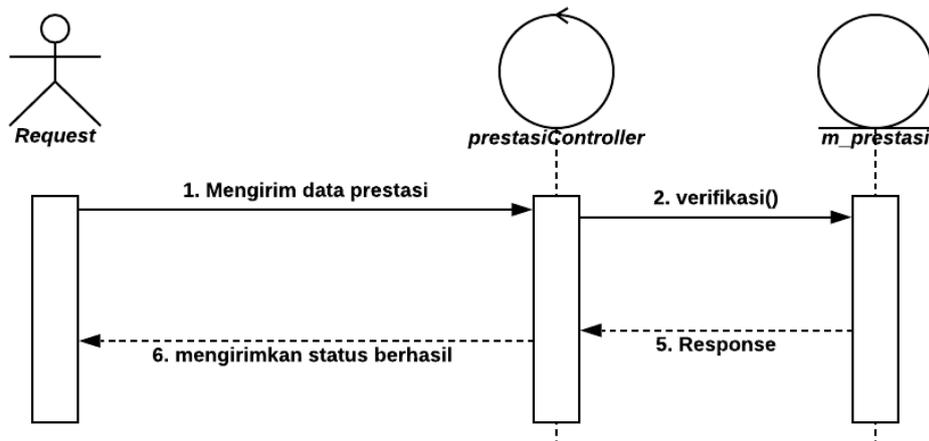
Sequence diagram verifikasi menjelaskan alur verifikasi prestasi dengan status menunggu. Kemahasiswaan menekan tombol verifikasi pada salah satu prestasi, setelah itu muncul *message box* verifikasi dan jika pilih verifikasi maka data akan dikirim ke *c_prestasi* dan diubah status menjadi verifikasi pada *database* setelah itu menampilkan halaman *v_kemahasiswaan*, jika pilih *cancel* maka akan diarahkan pada halaman *v_kemahasiswaan*. Selain itu kemahasiswaan dapat menekan tombol tolak pada salah satu prestasi, setelah itu muncul *message box* tolak dan jika pilih tolak maka data akan dihapus dari *database* setelah itu



menampilkan halaman v_kemahasiswaan. *Sequence diagram* verifikasi bisa dilihat pada Gambar 5.16.



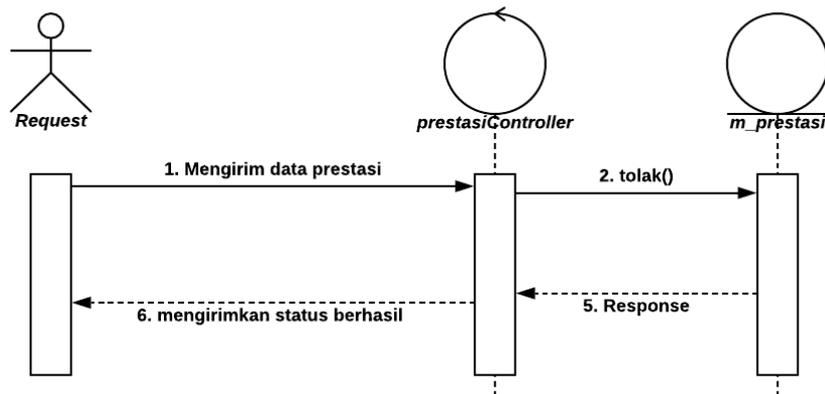
Gambar 5.16 *Sequence diagram* verifikasi



Gambar 5.17 *Sequence diagram* verifikasi bagian verifikasi pada API prestasi

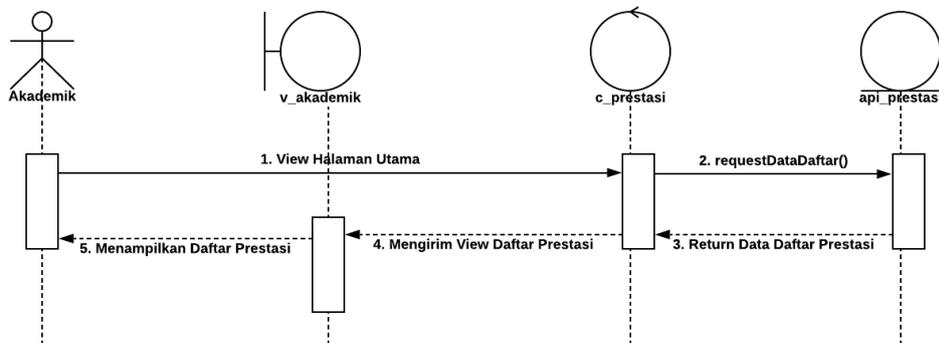
Pada Gambar 5.17 menunjukkan *sequence diagram* verifikasi pada API prestasi. Prestasi yang dipilih akan dikirim melalui prestasiController setelah diubah status nya menjadi verifikasi pada database yang direpresentasikan pada model m_prestasi, setelah perubahan berhasil maka akan diberikan response berupa JSON dengan pesan status berhasil.

Pada Gambar 5.18 menunjukkan *sequence diagram* tolak pada API prestasi. Prestasi yang dipilih akan dikirim melalui prestasiController setelah dihapus pada database yang direpresentasikan pada model m_prestasi, setelah perubahan berhasil maka akan diberikan response berupa JSON dengan pesan status berhasil.

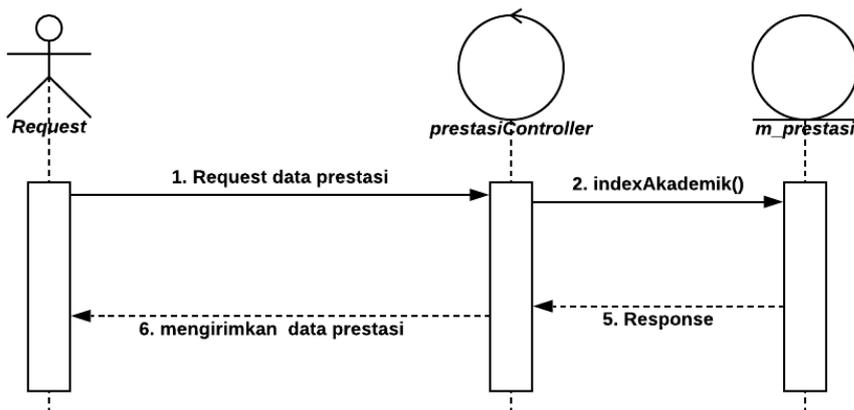


Gambar 5.18 Sequence diagram verifikasi bagian tolak pada API prestasi

5.2.8 Sequence Diagram Lihat Daftar Prestasi



Gambar 5.19 Sequence diagram lihat daftar prestasi



Gambar 5.20 Sequence diagram lihat daftar prestasi pada API prestasi

Sequence diagram lihat daftar prestasi menjelaskan alur menampilkan daftar prestasi pada sistem informasi perekaman prestasi mahasiswa pada karyawan bagian kemahasiswaan. Setelah masuk kedalam sistem dengan login sebagai

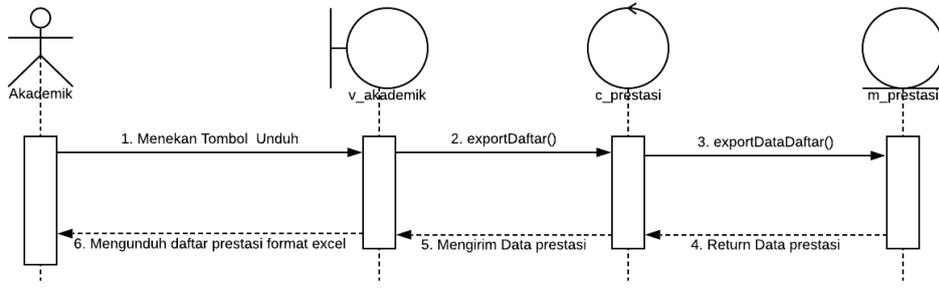


akademik, sistem akan mengarahkan kepada halaman `v_akademik` secara otomatis. *Sequence diagram* lihat daftar prestasi bisa dilihat pada Gambar 5.19.

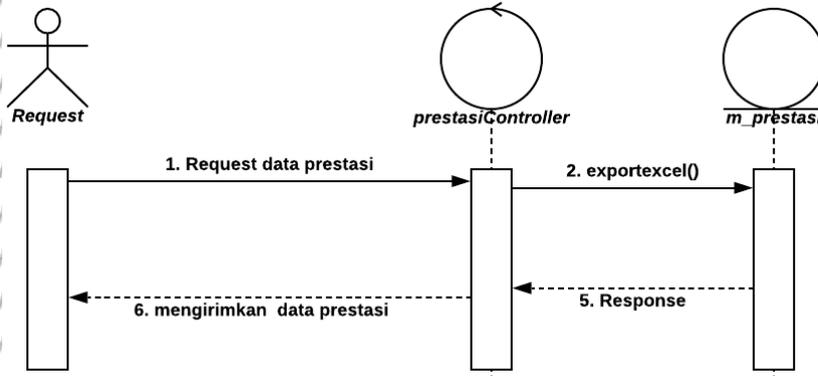
Pada Gambar 5.20 menunjukkan *sequence diagram* lihat daftar prestasi pada API prestasi. Dilakukan *request* data prestasi dengan status verifikasi. *Request* dilakukan melewati *prestasiController* kemudian meminta data pada *database* yang direpresentasikan pada *model m_prestasi*. Setelah itu akan diberikan *response* data prestasi dalam bentuk JSON.

5.2.9 Sequence Diagram Unduh Daftar Prestasi

Sequence diagram unduh daftar prestasi menjelaskan alur unduh prestasi mahasiswa yang ada pada sistem informasi perekaman prestasi mahasiswa. Akademik menekan tombol unduh setelah itu dilakukan *request* untuk meminta data prestasi mahasiswa dan dikembalikan dalam bentuk *file* berformat *excel*. *Sequence diagram* unduh daftar prestasi bisa dilihat pada Gambar 5.21.



Gambar 5.21 Sequence diagram unduh daftar prestasi



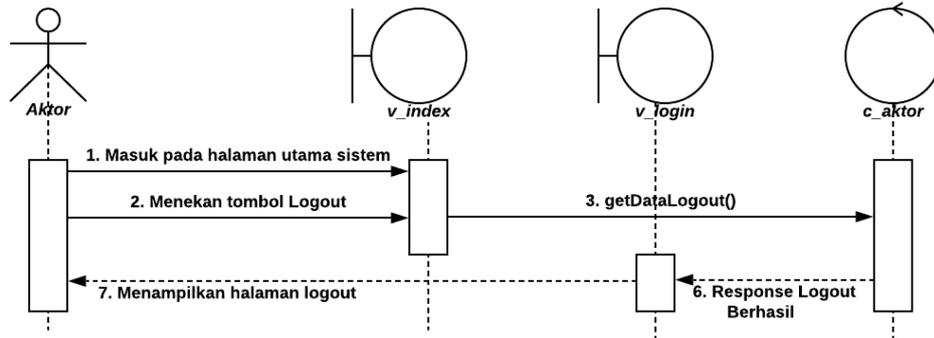
Gambar 5.22 Sequence diagram unduh daftar prestasi pada API prestasi

Pada Gambar 5.22 menunjukkan *sequence diagram* unduh daftar prestasi pada API prestasi. Dilakukan *request* data prestasi dengan status verifikasi pada *database* yang direpresentasikan oleh *model m_prestasi* melalui *controller* prestasiController. Setelah itu *response* data prestasi akan diberikan dalam format JSON.



5.2.10 Sequence Diagram Logout

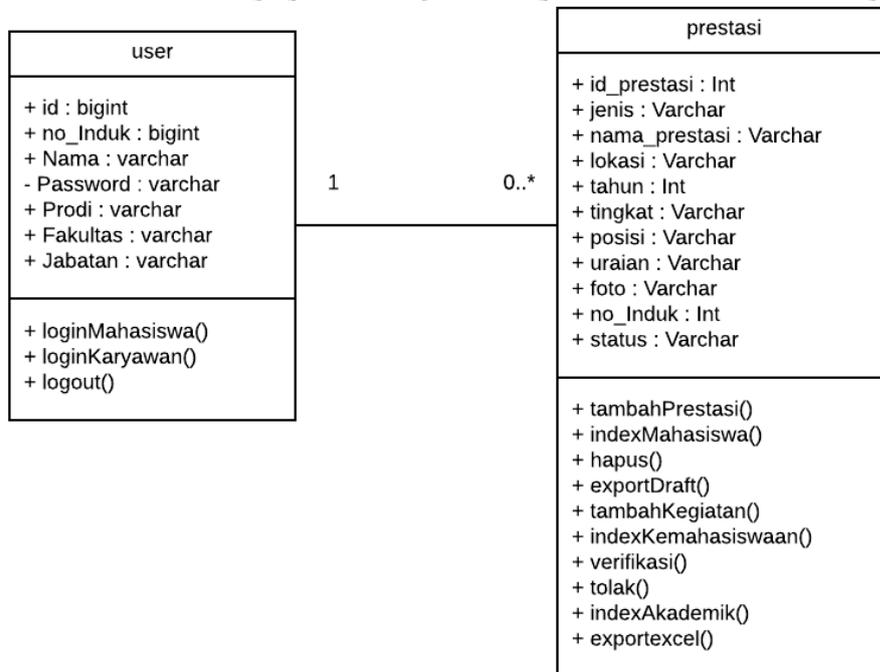
Sequence diagram logout menjelaskan alur dari aktor ketika telah masuk kedalam sistem informasi perekaman prestasi mahasiswa dan akan keluar dari sistem. Aktor menekan tombol *logout* pada *c_aktor* akan dilakukan *logout* dengan dan kemudian aktor akan diarahkan kepada halaman *login*. Sequence diagram *Logout* bisa dilihat pada Gambar 5.23



Gambar 5.23 Sequence diagram logout

5.3 Class Diagram

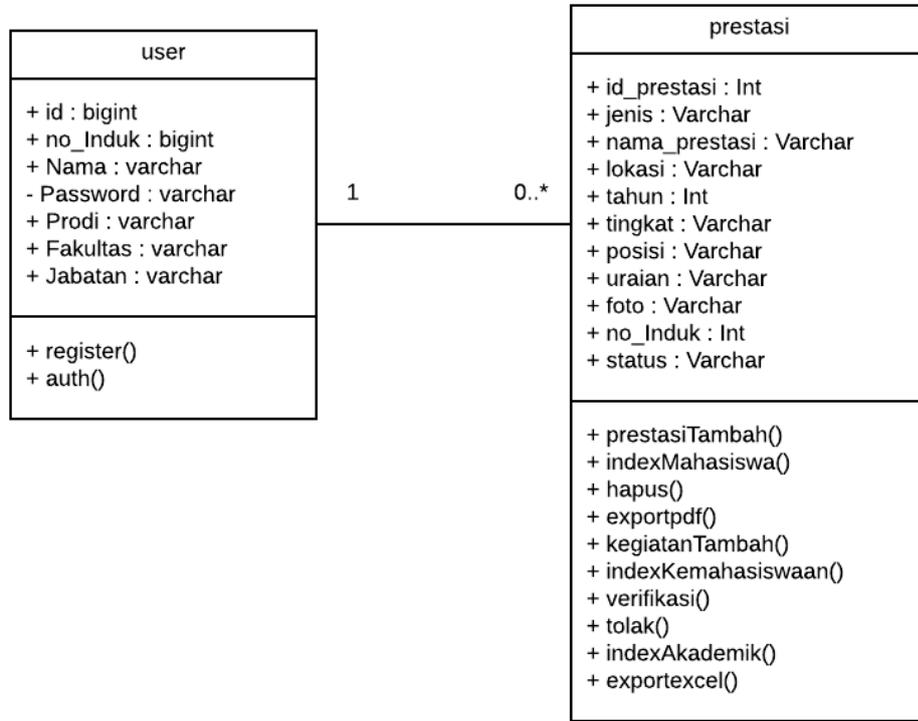
Class diagram menggambarkan objek berupa atribut serta *method* yang diperlukan oleh sistem informasi perekaman prestasi mahasiswa. Pada tahap ini, *class diagram* dibentuk berdasarkan *sequence diagram*.



Gambar 5.24 Class diagram website Sistem Informasi Perekaman Prestasi Mahasiswa



Pada Gambar 5.24, terdapat sebuah *class diagram* dari sistem informasi perekaman prestasi mahasiswa yang mana memiliki dua *class*, yaitu *user* dan *prestasi*. Relasi antar kedua *class* tersebut adalah *one to many* yang mana objek pada *class user* dapat memiliki banyak objek pada *prestasi*. Selain itu pada masing – masing *class* juga memiliki atribut dan *method*.



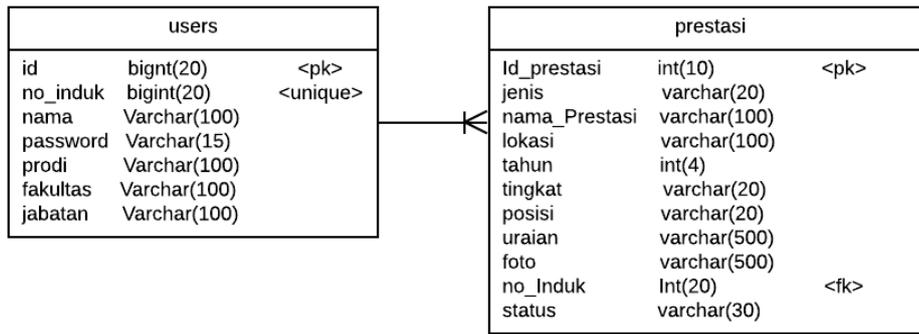
Gambar 5.25 Class Diagram API prestasi

Pada Gambar 5.25, terdapat sebuah *class diagram* dari API prestasi yang memiliki dua *class*, yaitu *user* dan *prestasi*. Hubungan antar kedua *class* tersebut adalah *one to many* dimana yang mana objek pada *class user* dapat memiliki banyak objek pada *prestasi*. Selain itu pada masing – masing *class* juga memiliki atribut dan *method*.

5.4 Physical Data model

Physical data model merupakan pemodelan yang digambarkan dengan sejumlah tabel yang saling terhubung (memiliki relasi). Tiap Tabel memiliki elemen yang disebut atribut. *Physical data model* sistem informasi perekaman prestasi mahasiswa dibuat berdasarkan *class diagram model* sistem informasi perekaman prestasi mahasiswa.

Gambar 5.26 menjelaskan tentang *physical data model* sistem informasi perekaman prestasi mahasiswa. Pada *physical data model* sistem informasi perekaman prestasi mahasiswa terdapat dua Tabel, yaitu Tabel *users* dan *prestasi*. Dimana hubungan anantara tabel tersebut adalah *one to many*, maksudnya setiap satu *user* dapat memiliki relasi satu hingga tak hingga *prestasi*.



Gambar 5.26 Physical data model sistem informasi perekaman prestasi mahasiswa

5.5 Perancangan REST API

Dilakukan perancangan REST API pada API prestasi dalam segi url yang nantinya akan diakses *web client*, *request* yang dibawa, dan *response* dalam bentuk *status code* dan konten dari *respon*. Pendefinisian parameter yang dibawa oleh *request* diambil berdasarkan atribut dari perancangan *class diagram* dan *physical data model*.

Tabel 5.1 Perancangan fitur *register* API Prestasi

URL	/register
Method	POST
Request body	no_induk = bigint(20) name = varchar(100) password = varchar(15) jabatan = varchar(100) prodi = varchar(100) fakultas = varchar(100)
Response	
Status Code	201 Created
Content	{ "sukses": true, "message": "Sukses", "data": { "no_induk": bigint(20), "name": varchar(100), "prodi": varchar(100), "jabatan": varchar(100), "fakultas": varchar(100), "id": int(20) } }



Pada tabel 5.1 merupakan perancangan dari fitur *register* pada API prestasi. Dengan menggunakan URL */register* dan *method* POST dan membawa *request* berisi *no_induk*, *name*, *password*, *jabatan*, *prodi*, dan *fakultas*. *Response* dari API akan berupa json dengan *key succes = true*, *message = success*, dan data berdasarkan *request* yang diberikan tanpa menampilkan dan menambahkan id *user* dan juga *status code = 201 Created*.

Tabel 5.2 Perancangan fitur *auth* API Prestasi

<i>URL</i>	<i>/auth</i>
<i>Method</i>	POST
<i>Request body</i>	<i>no_induk</i> = bigint(20) <i>password</i> = varchar(15)
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "message": "data user", "data": { "no_induk": bigint(20), "name": varchar(100), "prodi": varchar(100), "fakultas": varchar(100), "jabatan": varchar(100) } }

Pada tabel 5.2 merupakan perancangan dari fitur *auth* pada API prestasi. Dengan menggunakan URL */auth* dan *method* POST dan membawa *request* berisi *no_induk* dan *password*. *Response* dari API akan berupa json dengan *key message = data user*, dan data berisikan *no_induk* berdasarkan *request no_induk*, *name*, *prodi*, *fakultas*, dan *jabatan* dan juga *status code = 200 OK*.

Tabel 5.3 Perancangan fitur tambah prestasi API Prestasi

<i>URL</i>	<i>/mahasiswa/prestasitambah</i>
<i>Method</i>	POST
<i>Request body</i>	<i>jenis</i> = varchar(20) <i>nama_prestasi</i> = varchar(100) <i>lokasi</i> = varchar(100) <i>tahun</i> = int(4) <i>tingkat</i> = varchar(20) <i>posisi</i> = varchar(20) <i>uraian</i> = varchar(500) <i>foto</i> = varchar(500)



	no_induk = bigint(20) status = varchar(30)
<i>Response</i>	
<i>Status Code</i>	201 Created
<i>Content</i>	{ "status": "berhasil" }

Pada tabel 5.3 merupakan perancangan dari fitur tambah prestasi pada API prestasi. Dengan menggunakan URL /mahasiswa/prestasi_tambah dan *method* POST dan membawa *request* berisi jenis, nama_prestasi, lokasi, tahun, tingkat, posisi, uraian, foto, no_induk, status. *Response* dari API akan berupa json dengan key status = berhasil dan juga status code = 201 Created.

Tabel 5.4 Perancangan fitur lihat prestasi API Prestasi

<i>URL</i>	/mahasiswa/{no_induk}
<i>Method</i>	GET
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	[{ "id_prestasi": int(10), "jenis": varchar(20), "nama_prestasi": varchar(100), "lokasi": varchar(100), "tahun": int(4), "tingkat": varchar(20), "posisi": varchar(20), "uraian": varchar(500), "foto": varchar(500), "no_induk": bigint(20), "status": varchar(30) }]

Pada tabel 5.4 merupakan perancangan dari fitur lihat prestasi pada API prestasi. Dengan menggunakan URL /mahasiswa/{no_induk} dan *method* GET tanpa membawa *request*. *Response* dari API akan berupa json dengan key berisi id_prestasi, jenis, nama prestasi, lokasi, tahun, tingkat, posisi, uraian, foto, no_induk berdasarkan no_induk pada url dan status dan juga status code = 201 Created.

Tabel 5.5 Perancangan fitur hapus prestasi API Prestasi

<i>URL</i>	/mahasiswa/{id_prestasi}/hapus
------------	--------------------------------



<i>Method</i>	DELETE
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "status": "berhasil" }

Pada tabel 5.5 merupakan perancangan dari fitur hapus prestasi pada API prestasi berdasarkan id_prestasi pada URL. Dengan menggunakan URL /mahasiswa/{{id_prestasi}}/hapus dan *method* DELETE tanpa membawa *request*. *Response* dari API akan berupa json dengan key *status* = berhasil dan status dan juga *status code* = 200 OK.

Tabel 5.6 Perancangan fitur unduh *draft* API Prestasi

<i>URL</i>	/mahasiswa/exportpdf/{{no_induk}}
<i>Method</i>	GET
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "user": { "id": int(20), "no_induk": bigint(20), "name": varchar(100), "prodi": varchar(100), "fakultas": varchar(100), "jabatan": varchar(100) }, "data_prestasi": [{ "id_prestasi": int(10), "jenis": varchar(20), "nama_prestasi": varchar(100), "lokasi": varchar(100), "tahun": int(4), "tingkat": varchar(20), "posisi": varchar(20), "uraian": varchar(500), "foto": varchar(500), "no_induk": bigint(20), "status": varchar(30) }] }



Pada tabel 5.6 merupakan perancangan dari fitur unduh *draft* pada API. Dengan menggunakan URL `/mahasiswa/exportpdf/{no_induk}` dan *method* GET tanpa membawa *request*. *Response* dari API akan berupa json dengan *key* *user* berisi *id*, *no_induk*, *name*, *prodi*, *fakultas*, *jabatan* serta *key* *data_prestasi* berisi *id_prestasi*, *jenis*, *nama_prestasi*, *lokasi*, *tahun*, *tingkat*, *posisi*, *uraian*, *foto*, *no_induk* berdasarkan *no_induk* dan *status* dan juga *status code* = 200 OK.

Tabel 5.7 Perancangan fitur tambah kegiatan API Prestasi

<i>URL</i>	<code>/kemahasiswaan/kegiatanTambah</code>
<i>Method</i>	POST
<i>Request body</i>	<i>jenis</i> = varchar(20) <i>nama_prestasi</i> = varchar(100) <i>lokasi</i> = varchar(100) <i>tahun</i> = int(4) <i>tingkat</i> = varchar(20) <i>posisi</i> = varchar(20) <i>uraian</i> = varchar(500) <i>foto</i> = varchar(500) <i>no_induk</i> = array(bigint(20)) <i>status</i> = varchar(30)
<i>Response</i>	
<i>Status Code</i>	201 Created
<i>Content</i>	<pre>{ "status": "berhasil" }</pre>

Pada tabel 5.7 merupakan perancangan dari fitur tambah kegiatan pada API prestasi. Dengan menggunakan URL `/kemahasiswaan/kegiatanTambah` dan *method* POST dan membawa *request* berisi *jenis*, *nama_prestasi*, *lokasi*, *tahun*, *tingkat*, *posisi*, *uraian*, *foto*, *array* berisi *no_induk*, dan *status*. *Response* dari API akan berupa json dengan *key* *status* = berhasil dan juga *status code* = 201 Created.

Tabel 5.8 Perancangan fitur verifikasi prestasi API Prestasi

<i>URL</i>	<code>/kemahasiswaan/{id_prestasi}/verifikasi</code>
<i>Method</i>	PUT
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	<pre>{ "status": "berhasil" }</pre>



Pada tabel 5.8 merupakan perancangan dari fitur verifikasi prestasi pada API prestasi berdasarkan `id_prestasi` pada URL. Dengan menggunakan URL `/kemahasiswaan/{id_prestasi}/verifikasi` dan *method* PUT tanpa membawa *request*. *Response* dari API akan berupa json dengan key *status* = berhasil dan status dan juga *status code* = 200 OK.

Tabel 5.9 Perancangan fitur tolak prestasi API Prestasi

<i>URL</i>	<code>/kemahasiswaan/{id_prestasi}/tolak</code>
<i>Method</i>	DELETE
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "status": "berhasil" }

Pada tabel 5.9 merupakan perancangan dari fitur tolak prestasi pada API prestasi berdasarkan `id_prestasi` pada URL. Dengan menggunakan URL `//kemahasiswaan/{id_prestasi}/tolak` dan *method* DELETE tanpa membawa *request*. *Response* dari API akan berupa json dengan key *status* = berhasil dan status dan juga *status code* = 200 OK.

Tabel 5.10 Perancangan fitur lihat daftar prestasi API Prestasi

<i>URL</i>	<code>/akademik</code>
<i>Method</i>	GET
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	[{ "id_prestasi": int(10), "jenis": varchar(20), "nama_prestasi": varchar(100), "lokasi": varchar(100), "tahun": int(4), "tingkat": varchar(20), "posisi": varchar(20), "uraian": varchar(500), "foto": varchar(500), "no_induk": bigint(20), "status": varchar(30) "user": {"id": int(20), "no_induk": bigint(20), "name": varchar(100), "prodi": varchar(100),



```

    "fakultas": varchar(100),
    "jabatan": varchar(100)
  }
}

```

Pada tabel 5.10 merupakan perancangan dari fitur lihat daftar prestasi pada API prestasi. Dengan menggunakan URL /akademik dan *method* GET tanpa membawa *request*. *Response* dari API akan berupa json berisi id_prestasi, jenis, nama prestasi, lokasi, tahun, tingkat, posisi, uraian, foto, no_induk, status, dan user yang berisi id, no_induk, name, prodi, fakultas, jabatan dan juga *status code* = 200 OK.

Tabel 5.11 Perancangan fitur unduh daftar prestasi API Prestasi

<i>URL</i>	/akademik/exportexcel
<i>Method</i>	GET
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	<pre> { "id_prestasi": int(10), "jenis": varchar(20), "nama_prestasi": varchar(100), "lokasi": varchar(100), "tahun": int(4), "tingkat": varchar(20), "posisi": varchar(20), "uraian": varchar(500), "foto": varchar(500), "no_induk": bigint(20), "status": varchar(30) "user": { "id": int(20), "no_induk": bigint(20), "name": varchar(100), "prodi": varchar(100), "fakultas": varchar(100), "jabatan": varchar(100) } } </pre>

Pada tabel 5.11 merupakan perancangan dari fitur unduh daftar prestasi pada API prestasi. Dengan menggunakan URL /akademik/exportexcel dan *method* GET tanpa membawa *request*. *Response* dari API akan berupa json berisi id_prestasi, jenis, nama prestasi, lokasi, tahun, tingkat, posisi, uraian, foto, no_induk, status, dan user yang berisi id, no_induk, name, prodi, fakultas, jabatan dan juga *status code* = 200 OK.

BAB 6 IMPLEMENTASI SISTEM

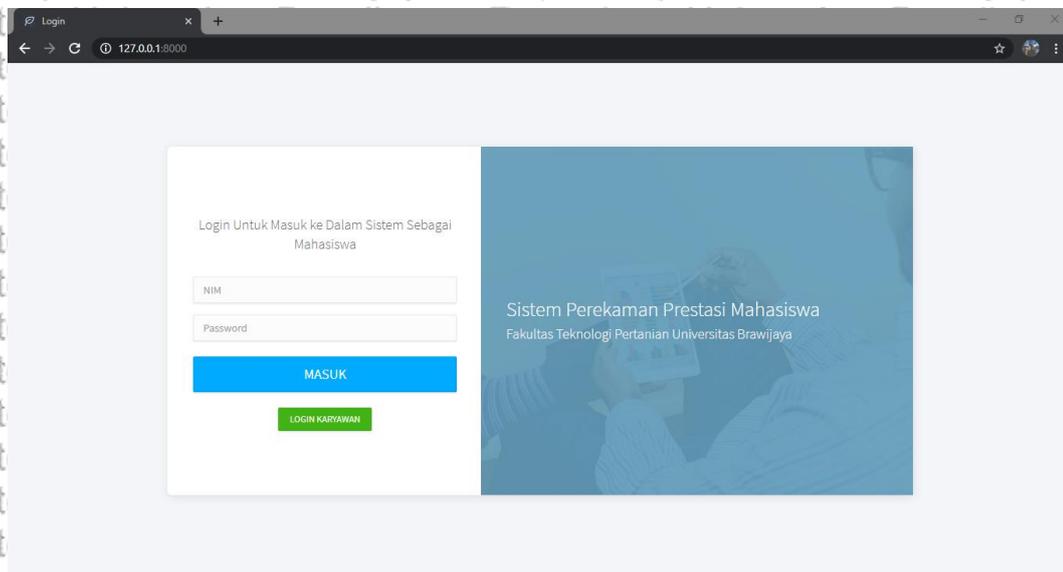
Setelah dilakukan analisa kebutuhan dan perancangan sistem, maka dilakukan implementasi kedalam bentuk *Source Code* berdasarkan perancangan yang telah dibuat, pada tahap ini sistem diimplementasikan pada *platform website* dengan menggunakan *framework* Laravel serta menggunakan *framework* Lumen untuk pengembangan REST API. Pada bagian implementasi sistem didokumentasikan berdasarkan implementasi UI (*User interface*) kemudian dilanjutkan potongan kode dari bagian *controller* dari masing – masing fitur.

Pada implementasi *website*. Dilakukan konfigurasi saat mengakses alamat API prestasi dengan menambahkan sintaks pada halaman *.env* seperti pada *Source Code* 6.1. Tujuan penambahan sintaks tersebut adalah ketika *controller* pada *website* akan mengakses API prestasi maka hanya akan perlu menuliskan sintaks *Base_url* nya saja.

```
.env
Base_url = http://localhost:8090/
```

Source Code 6.1 Implementasi *base url* pada *website*

6.1 Impelementasi *Login*



Gambar 6.1 Implementasi UI fitur *login* Mahasiswa

Pada fitur *login* mahasiswa seperti pada Gambar 6.1, terdapat sebuah halaman yang berisikan sebuah form masukkan untuk mengisi NIM serta *password* dan tombol masuk serta tombol *login* karyawan.

```
c_aktor.php
public function loginMahasiswa(Request $request) {
    $mahasiswa = '';
```




```

        'contents' =>
        $request->input('password'),
    ]
    ];
    $url = env('Base_url').'register';
    $req = $client->request("POST", $url, $data);
    $response = $req->getBody();
    session(['nama' => $json['nama']]);
    session(['no_induk' => $json['nim']]);
    session(['prodi' => $json['prodi']]);
    return redirect('/mahasiswa');
} else {
    return redirect('/');
}
}
}

```

Source Code 6.2 Implementasi kode fitur login Mahasiswa

Pada *Source Code 6.2* terdapat *function login* mahasiswa dengan algoritma dimulai dari melakukan *request* data pada API login menggunakan NIM UB dan *password* SIAM UB yang didapat dari BEM FILKOM UB, kemudian dilakukan pengecekan terhadap fakultas dari NIM yang telah dimasukkan, jika Teknologi Pertanian maka sistem akan melakukan *create user* yang akan disimpan pada *database* dengan mengakses API prestasi. Kemudian akan dibuat *session* pada nama, nim dan prodi *user* yang telah masuk dan diarahkan kepada halaman mahasiswa.

c_aktor.php

```

public function register(Request $request) {
    $ngecek = User::where
    (['no_induk' => $request->input('no_induk')])->first();
    if ($ngecek['no_induk'] == null) {
        $no_induk = $request->input('no_induk');
        $name = $request->input('name');
        $password = Hash::make($request->input('password'));
        $jabatan = $request->input('jabatan');
        $prodi = $request->input('prodi');
        $fakultas = $request->input('fakultas');

        $register = User::create([
            'no_induk' => $no_induk,
            'name' => $name,
            'password' => $password,
            'prodi' => $prodi,
            'jabatan' => $jabatan,
            'fakultas' => $fakultas
        ]);

        if ($register) {
            return response()->json([
                'succes' => true,
                'message' => 'Sukses',
                'data' => $register
            ], 201);
        } else {

```



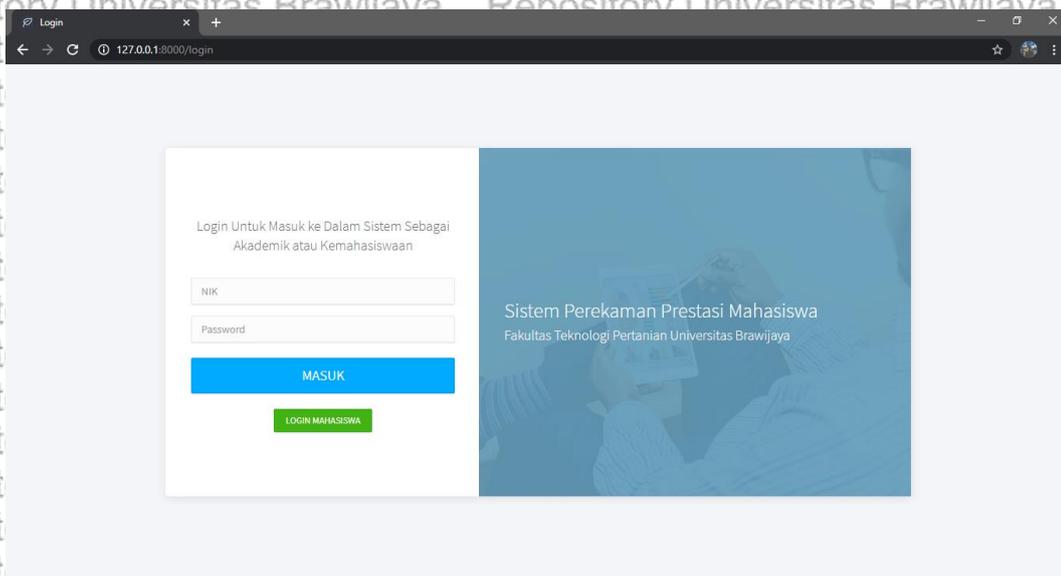
```

return response()->json([
    'succes' => false,
    'message' => 'Fail',
    'data' => [
        ], 400);
}
} else {
return response()->json([
    'succes' => true,
    'message' => 'Sukses',
    'data' => $ngecek
    ], 201);
}
}

```

Source Code 6.3 Implementasi kode fitur login mahasiswa pada API prestasi

Pada *Source Code 6.3* terdapat *function auth* dengan algoritma dimulai dengan dilakukan pengecekan apakah *no_induk* yang dimasukkan telah ada pada *database*. Jika tidak, maka akan dilakukan *create* berdasarkan informasi NIM, nama, *password* yang dienkripsi, prodi, serta fakultas. Setelah itu masuk kedalam seleksi kondisi jika berhasil maka akan mengirimkan informasi *user* dan jika gagal akan dikirimkan pesan *no_induk* atau pesan salah. Jika *nim* yang dimasukkan sudah ada sebelumnya maka data akan dikembalikan dalam bentuk JSON.



Gambar 6.2 Implementasi UI fitur Login Karyawan

Pada fitur *login* karyawan yang terdiri dari kemahasiswaan dan akademik seperti pada Gambar 6.2. Terdapat sebuah halaman yang berisikan sebuah form masukkan untuk mengisi NIK serta *password* dan tombol masuk serta tombol *login* mahasiswa.

```

c_aktor.php
public function loginKaryawan(Request $request) {
    $client = new \GuzzleHttp\Client();
    $data = [
        'multipart' => [

```



```

        'name' => 'no_induk',
        'contents' => $request->no_induk,
    ],
    [
        'name' => 'password',
        'contents' => $request->password,
    ]
];

$url = env('Base url').'auth';
$req = $client->request("POST", $url, $data);
$response = $req->getBody();
$akses = json_decode($response, true);
session(['message' => $akses['message']]);
if (session('message') == 'salah') {
    return redirect('/');
} elseif (session('message') == 'data user') {
    session(['no_induk' => $akses['data']['no_induk']]);
    session(['nama' => $akses['data']['name']]);
    session(['jabatan' => $akses['data']['jabatan']]);
    switch (session('jabatan')) {
        case 'Kemahasiswaan':
            return redirect('/kemahasiswaan');
            break;
        case 'Akademik':
            return redirect('/akademik');
            break;
    }
}
}
}

```

Source Code 6.4 Implementasi kode fitur login Karyawan

Pada Source Code 6.4 terdapat *function login* dengan algoritma dimulai dari melakukan *request* data pada API prestasi dengan masukkan *no_induk* dan *password*, kemudian akan masuk seleksi kondisi jika aktor kemahasiswaan akan diarahkan pada halaman kemahasiswaan, jika aktor akademik maka akan diarahkan pada halaman akademik.

loginController.php

```

public function auth(Request $request) {
    $no_induk = $request->no_induk;
    $password = $request->password;
    $encryptCode = sha1($password);
    if ($no_induk == null || $password == null) {
        return response()->json(
            [
                'message'=>'masukkan kosong',
                'data'=>(object) []
            ], 400);
    }
    $users = User::where('no_induk', $no_induk)->first();
    if ($users == null) {
        return response()->json(
            [
                'message'=>'akun tidak ditemukan',
                'data'=>(object) []
            ]
        );
    }
}

```



```

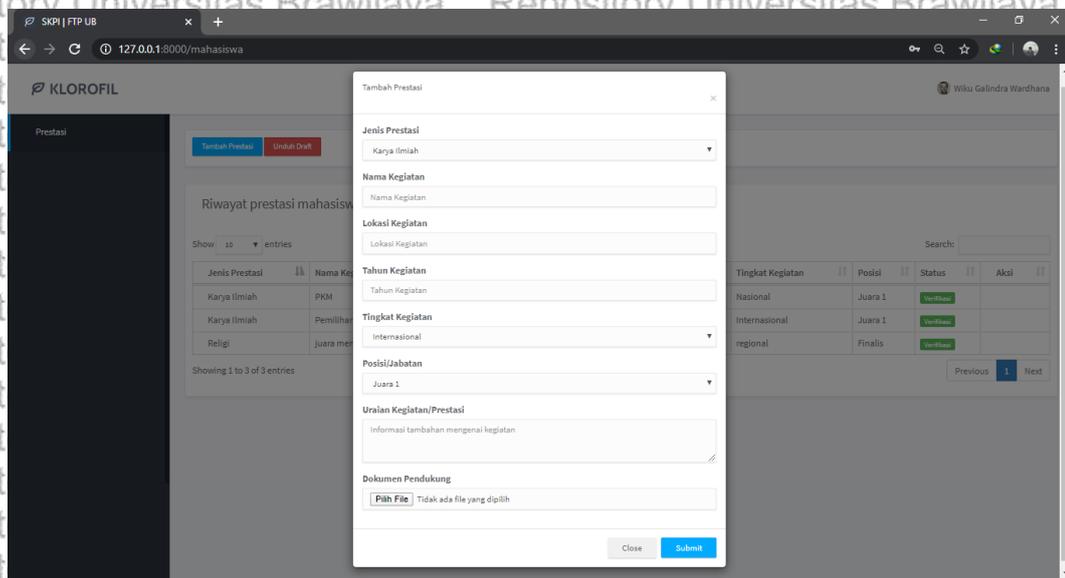
    ],404);
    }
    if (Hash::check($password, $users->password)) {
        return response()->json(
            [
                'message'=>'data user',
                'data'=>[
                    'no_induk'=>$users->no_induk,
                    'name'=>$users->name,
                    'prodi'=>$users->prodi,
                    'fakultas'=>$users->fakultas,
                    'jabatan'=>$users->jabatan
                ],200);
            } else {
                return response()->json(
                    [
                        'message'=>'no. induk atau password salah',
                        'data'=>(object) []
                    ],401);
            }
        }
    }
}

```

Source Code 6.5 Implementasi kode fitur login pada API prestasi

Pada *Source Code 6.5* terdapat *function auth* dengan algoritma dimulai dengan mengambil data masukkan kemudian dilakukan pengecekan apakah masukan kosong, jika tidak maka akan disamakan sesuai dengan *user* yang telah terdaftar pada *database*, jika sesuai maka akan dilakukan pengecekan *password* dan jika sesuai maka akan mengirimkan informasi *user* dan jika tidak sesuai akan dikirimkan pesan salah.

6.2 Implementasi Tambah prestasi



Gambar 6.3 Implementasi UI fitur mahasiswa tambah prestasi

Pada fitur mahasiswa tambah prestasi seperti pada Gambar 6.3 terdapat sebuah formulir masukkan data prestasi yang terdiri dari jenis prestasi berupa pilihan, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan berupa



pilihan, posisi/jabatan berupa pilihan, uraian kegiatan prestasi, dan unggah Gambar setelah itu terdapat tombol *close* dan *submit*.

c_prestasi.php

```

public function tambahPrestasi(Request $request){
    $client = new \GuzzleHttp\Client();
    $image_path = $request->file('foto')->getPathname();
    $image_mime = $request->file('foto')->getMimeType();
    $image_org = $request->file('foto')->getClientOriginalName();
    $data = [
        'multipart' => [
            [
                'name' => 'foto',
                'filename' => $image_org,
                'Mime-Type' => $image_mime,
                'contents' => fopen($image_path, 'r'),
            ],
            [
                'name' => 'jenis',
                'contents' => $request->jenis,
            ],
            [
                'name' => 'nama_prestasi',
                'contents' => $request->nama_prestasi,
            ],
            [
                'name' => 'lokasi',
                'contents' => $request->lokasi,
            ],
            [
                'name' => 'tahun',
                'contents' => $request->tahun,
            ],
            [
                'name' => 'tingkat',
                'contents' => $request->tingkat,
            ],
            [
                'name' => 'posisi',
                'contents' => $request->posisi,
            ],
            [
                'name' => 'uraian',
                'contents' => $request->uraian,
            ],
            [
                'name' => 'no_induk',
                'contents' => session('no_induk'),
            ],
        ],
    ];
    $url = env('Base_url').'/mahasiswa/prestasi/tambah';
    $req = $client->request("POST", $url, $data);
    $response = $req->getBody();
    return redirect('/mahasiswa');
}

```

Source Code 6.6 Implementasi kode fitur mahasiswa tambah prestasi



Pada *Source Code* 6.6 terdapat *function* tambahPrestasi dengan algoritma dimulai dengan mengambil data masukkan yang dijadikan ke dalam satu buah *variable array*. Kemudian dikirim kepada API prestasi dengan metode *post*. Setelah itu akan diarahkan kepada halaman mahasiswa.

prestasiController.php

```
public function prestasiTambah(Request $request) {
    $data_prestasi = \App\m_prestasi::create
    ($request->all());
    if ($request->hasFile('foto')) {
        $request->file('foto')->move('images/',
        $request->file('foto')->getClientOriginalName());
        $data_prestasi->foto =
        $request->file('foto')->getClientOriginalName();
        $data_prestasi->save();
    }
    return response()->json(['status'=>'berhasil'], 201);
}
```

Source Code 6.7 Implementasi kode fitur mahasiswa tambah prestasi pada API prestasi

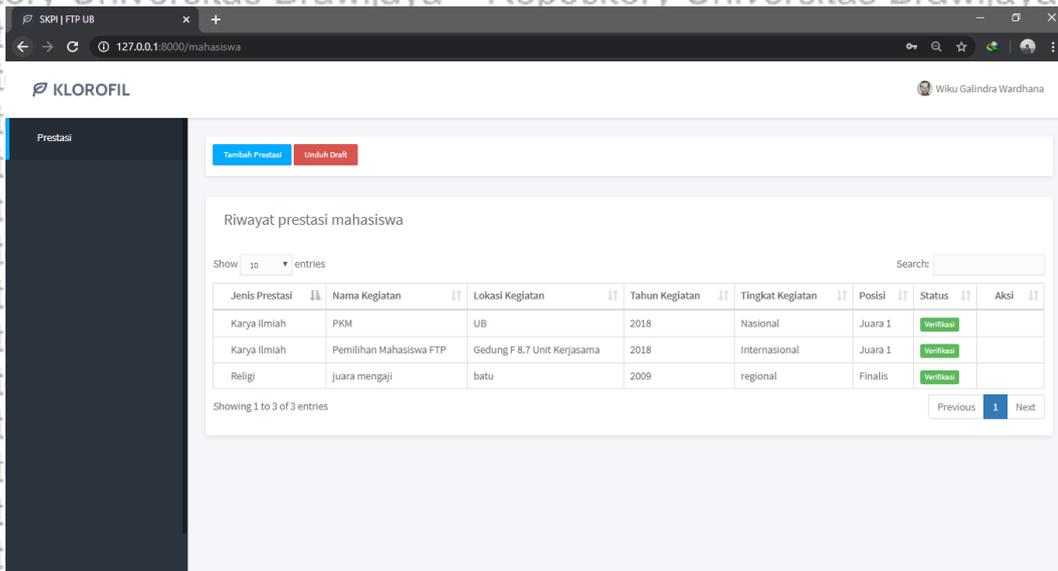
Pada *Source Code* 6.7 terdapat *function* mahasiswa tambah dengan algoritma dimulai dengan mengambil *request* berupa data masukkan yang telah dikirimkan kemudian dimasukkan ke dalam *database* yang direpresentasikan oleh *model* *m_prestasi* setelah itu akan mengembalikan sebuah *response* dengan status berhasil.

6.3 Implementasi Mahasiswa Lihat Prestasi

Pada fitur mahasiswa lihat prestasi seperti pada Gambar 6.4. Terdapat sebuah halaman mahasiswa yang terdiri dari tombol tambah prestasi dan unduh *draft*. Selain itu terdapat Tabel yang berisikan informasi prestasi yang telah dimasukkan dengan status verifikasi dan menunggu.

Pada *Source Code* 6.8. Terdapat *function* indexMahasiswa dengan algoritma dimulai dengan mengambil data prestasi berdasarkan *no_induk* dan dikirim kepada API prestasi, kemudian merubah *response* berupa format JSON menjadi *array* kemudian ditampilkan pada view *v_mahasiswa*.

Pada *Source Code* 6.9. Terdapat *function* indexMahasiswa dengan algoritma dimulai dengan mengambil data prestasi berdasarkan *no_induk* kemudian data prestasi akan dikembalikan dalam bentuk *response* dengan format JSON.



Gambar 6.4 Implementasi UI fitur mahasiswa lihat prestasi

```
c_prestasi.php
public function indexKemahasiswaan() {
    $client = new \GuzzleHttp\Client();
    $url1 = env('Base_url').'kemahasiswaan';
    $request = $client->get($url1);
    $response = json_decode($request->getBody());
    $url2 = env('Base_url').'kemahasiswaan/listuser';
    $request2 = $client->get($url2);
    $response2 = json_decode($request2->getBody());
    return view('v_kemahasiswaan',
        ['data_prestasi' => $response, 'list_user' => $response2]);
}
```

Source Code 6.8 Implementasi kode fitur mahasiswa lihat prestasi

```
prestasiController.php
public function indexMahasiswa($nim) {
    $data_prestasi = \App\m_prestasi::where
        ('no_induk', $nim)->get();
    return response()->json($data_prestasi, 200);
}
```

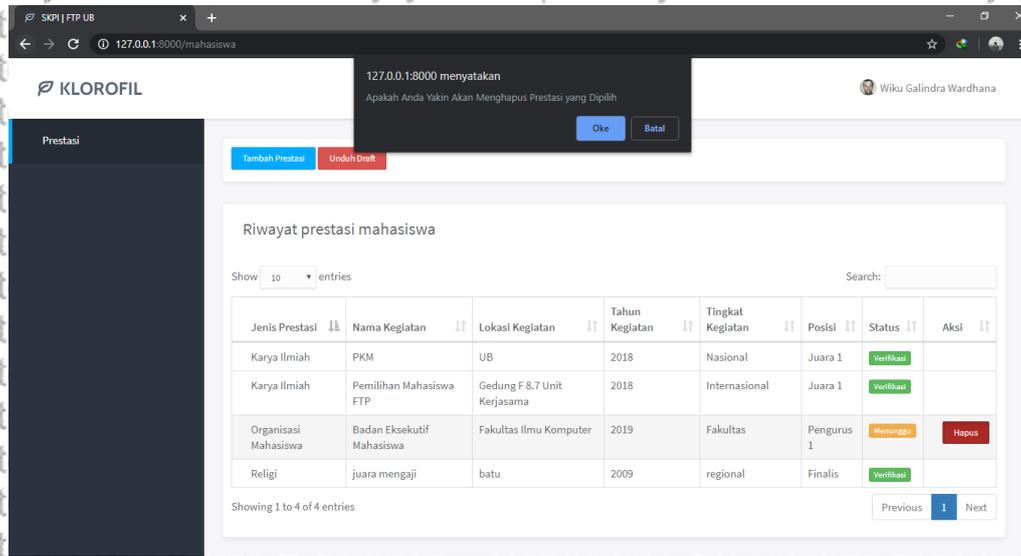
Source Code 6.9 Implementasi kode fitur mahasiswa lihat prestasi pada API prestasi

6.4 Implementasi Mahasiswa Hapus Prestasi

Pada fitur mahasiswa hapus prestasi seperti pada Gambar 6.5 terdapat sebuah halaman mahasiswa dan terdapat tombol hapus prestasi pada prestasi dengan status menunggu, dan jika ditekan akan muncul sebuah konfirmasi untuk menghapus prestasi yang dipilih.

Pada Source Code 6.10 terdapat function hapus dengan algoritma dimulai dengan mengambil id data prestasi dan dikirim kepada API prestasi, kemudian

merubah *response* berupa format JSON dan mengembalikan dengan memanggil *route web /mahasiswa* yang mengarah pada *controller indexMahasiswa*.



Gambar 6.5 Implementasi UI fitur mahasiswa hapus prestasi

```
c_prestasi.php
public function hapus($id_prestasi) {
    $client = new \GuzzleHttp\Client();
    $url = env('Base url') . 'mahasiswa/' . $id_prestasi . '/hapus';
    $request = $client->get($url);
    $response = json_decode($request->getBody());
    return redirect('/mahasiswa');
}
```

Source Code 6.10 Implementasi kode fitur mahasiswa hapus prestasi

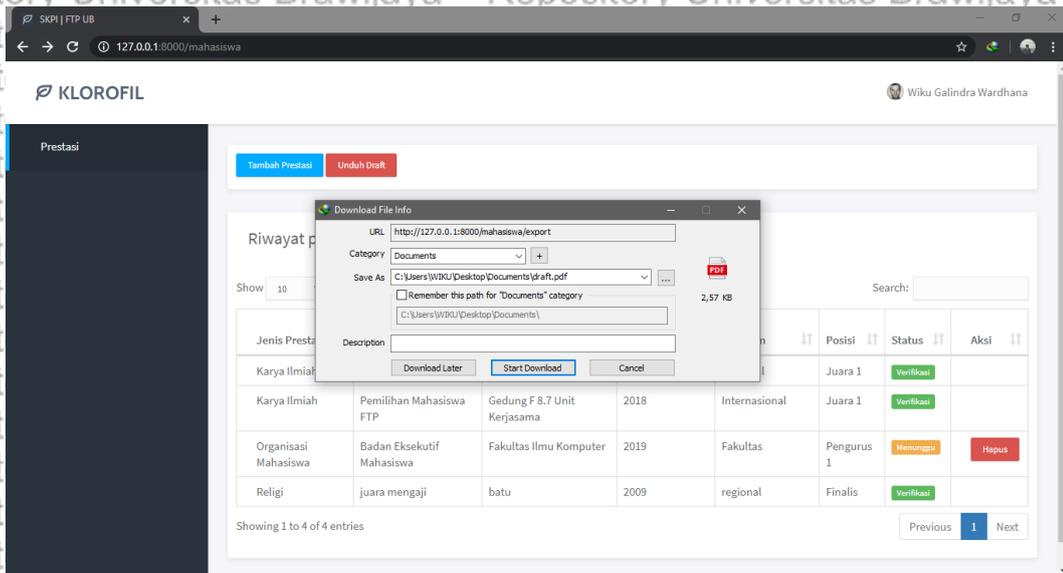
```
prestasiController.php
public function hapus($id_prestasi) {
    $prestasi = \App\m_prestasi::find($id_prestasi);
    $prestasi->delete();
    return response()->json(['status'=>'berhasil']);
}
```

Source Code 6.11 Implementasi kode fitur mahasiswa hapus prestasi pada API prestasi

Pada *Source Code 6.11* terdapat *function* hapus dengan algoritma dimulai dengan mengambil id prestasi yang dikirim setelah itu dilakukan penghapusan dan mengembalikan *response* dengan status berhasil.

6.5 Implementasi Mahasiswa Unduh Draft

Pada fitur mahasiswa unduh *draft* seperti pada Gambar 6.6. Terdapat sebuah halaman mahasiswa dan terdapat tombol unduh *draft* yang jika ditekan akan mengunduh data prestasi mahasiswa yang masuk ke dalam sistem dengan status verifikasi dengan format *file pdf*.



Gambar 6.6 Implementasi UI fitur mahasiswa unduh draft

Pada *Source Code 6.12* terdapat *function* `exportDraft` dengan algoritma dimulai dengan memanggil API prestasi, kemudian merubah *response* berupa format JSON dan merubah file `prestasi_pdf.html` menjad format pdf setelah itu mengembalikan hasil dengan mengunduh file bernama `draft.pdf`.

`c_prestasi.php`

```
public function exportDraft() {
    $client = new \GuzzleHttp\Client();
    $url = env('Base_url') .
    'mahasiswa/exportpdf/' . session('no_induk');
    $request = $client->get($url);
    $response = json_decode($request->getBody());
    $data_prestasi = $response->data_prestasi;
    $data_user = $response->user;
    $pdf = PDF::loadView('prestasi_pdf',
    ['prestasi' => $data_prestasi, 'user' => $data_user]);
    return $pdf->download('draft.pdf');
}
```

Source Code 6.12 Implementasi kode fitur mahasiswa unduh draft

`prestasiController.php`

```
public function exportpdf() {
    $data_prestasi = \App\m_prestasi::with('User')
    ->where('no_induk', session('no_induk'))
    ->where('status', 'Verifikasi')->get();
    return response()->json($data_prestasi, 200);
}
```

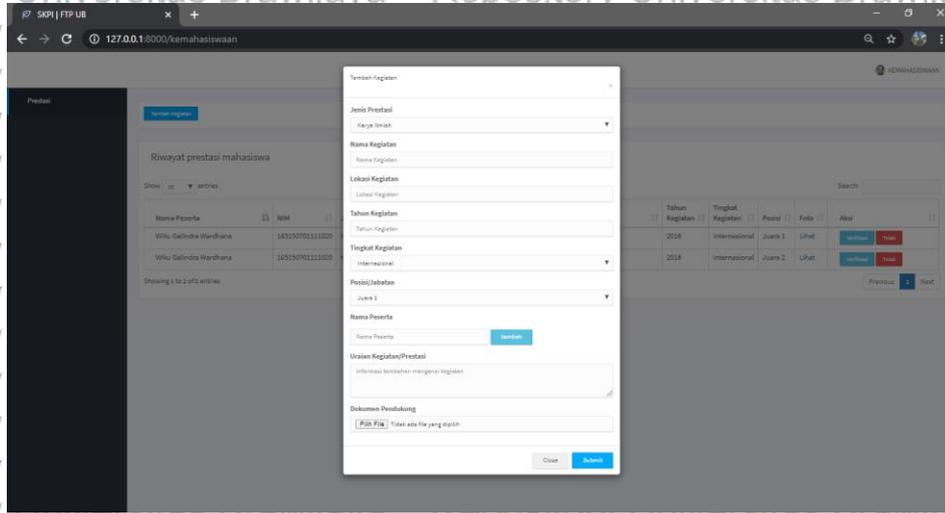
Source Code 6.13 Implementasi kode fitur mahasiswa unduh draft pada API prestasi

Pada *Source Code 6.13* terdapat *function* `exportpdf` dengan algoritma dimulai dengan mengambil data prestasi pada *database* yang direpresentasikan pada *model* `m_prestasi` berdasarkan `no_induk` mahasiswa yang sedang masuk ke dalam



sistem dan dengan status verifikasi, setelah itu mengembalikan hasil data prestasi berupa JSON.

6.6 Implementasi Kemahasiswaan Tambah kegiatan



Gambar 6.7 Implementasi UI fitur kemahasiswaan tambah kegiatan

Pada fitur kemahasiswaan tambah kegiatan seperti pada Gambar 6.7. Terdapat sebuah form masukkan data prestasi yang terdiri dari jenis prestasi berupa pilihan, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan berupa pilihan, posisi/jabatan berupa pilihan, nama peserta yang bisa dimasukkan lebih dari satu mahasiswa dengan kondisi dia telah terdaftar di pada *database*, uraian kegiatan prestasi, dan unggah Gambar setelah itu terdapat tombol *close* dan *submit*.

c_prestasi.php

```
public function tambahKegiatan(Request $request) {
    $client = new \GuzzleHttp\Client();
    $image_path = $request->file('foto')->getPathname();
    $image_mime = $request->file('foto')->getMimeType();
    $image_org = $request->file('foto')
    ->getClientOriginalName();
    $data = [
        'multipart' => [
            [
                'name' => 'foto',
                'filename' => $image_org,
                'Mime-Type' => $image_mime,
                'contents' => fopen($image_path, 'r'),
            ],
            [
                'name' => 'jenis',
                'contents' => $request->jenis,
            ],
            [
                'name' => 'nama_prestasi',
                'contents' => $request->nama_prestasi,
            ],
            [
                'name' => 'lokasi',
```



```

        'contents' => $request->lokasi,
    ],
    'name' => 'tahun',
    'contents' => $request->tahun,
],
    'name' => 'tingkat',
    'contents' => $request->tingkat,
],
    'name' => 'posisi',
    'contents' => $request->posisi,
],
    'name' => 'uraian',
    'contents' => $request->uraian,
],
    'name' => 'no_induk',
    'contents' => session('no_induk'),
];
}
}
$url = env('Base url').'kemahasiswaan/kegiatanTambah';
$req = $client->request("POST", $url, $data);
$response = $req->getBody();
return 'berhasil';
}
}

```

Source Code 6.14 Implementasi kode fitur kemahasiswaan tambah kegiatan

Pada *Source Code 6.14* terdapat *function* tambahKegiatan dengan algoritma dimulai dengan mengambil data masukkan kemudian dikirim kepada API prestasi dan mengambil *response* dari API prestasi. Setelah itu mengembalikan 'berhasil'.

prestasiController.php

```

public function kegiatanTambah(Request $request) {
    $arrayUser = json_decode($request->list_user, true);
    $jenisPrestasi = $request->jenisPrestasi;
    $namaPrestasi = $request->namaPrestasi;
    $lokasiKegiatan = $request->lokasiKegiatan;
    $tahunKegiatan = $request->tahunKegiatan;
    $tingkatKejuaraan = $request->tingkatKejuaraan;
    $posisi = $request->posisi;
    $uraian = $request->uraian;
    if ($request->hasFile('foto')) {
        $request->file('foto')->move('images/', $request
        ->file('foto')->getClientOriginalName());
    }
    for ($i=0; $i < count($arrayUser); $i++) {
        m_prestasi::insert([
            'jenis' => $jenisPrestasi,
            'no_induk' => $arrayUser[$i],
            'foto' => $request->file('foto')
            ->getClientOriginalName(),
            'nama_prestasi' => $namaPrestasi,
            'lokasi' => $lokasiKegiatan,

```



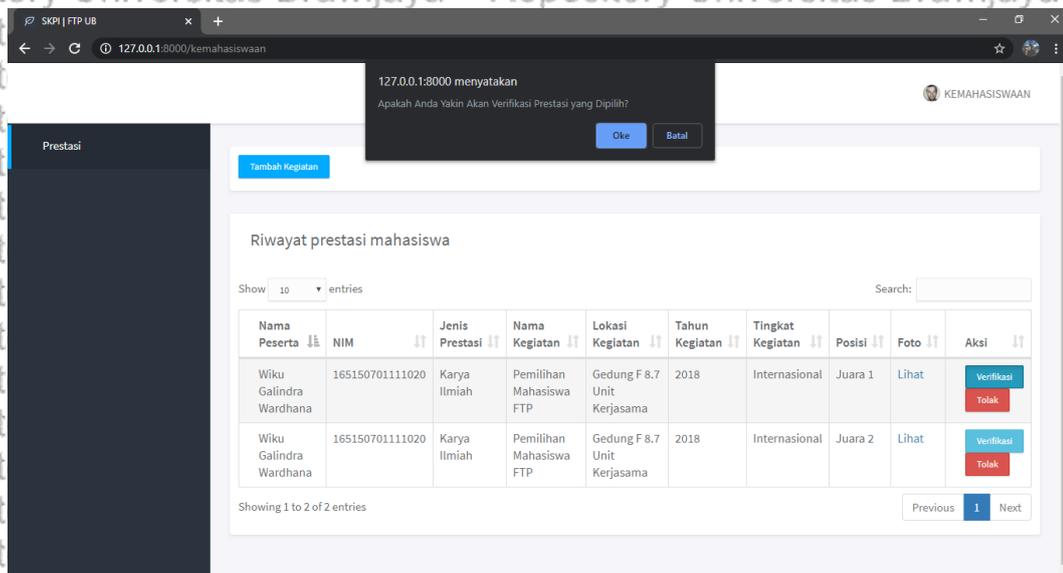
```
'tahun' => $tahunKegiatan,
'tingkat' => $tingkatKejuaraan,
'posisi' => $posisi,
'uraian' => $uraian,
'status' => 'Verifikasi'
);
}
return response()->json(['status'=>'berhasil'], 201);
}
```

Source Code 6.15 Implementasi kode fitur kemahasiswaan tambah kegiatan pada API prestasi

Pada *Source Code 6.15* terdapat *function* kegiatanTambah dengan algoritma dimulai dengan mengambil *request* berupa data masukkan yang telah dikirimkan kemudian dimasukkan ke dalam *database* yang direpresentasikan oleh *model* m_prestasi dengan metode perulangan berdasarkan no_induk mahasiswa yang ditambahkan pada form masukkan tambah kegiatan. Setelah itu akan mengembalikan sebuah *response* dengan status “berhasil”

6.7 Implementasi Verifikasi

Pada fitur verifikasi (bagian verifikasi) seperti pada Gambar 6.8. Terdapat sebuah halaman kemahasiswaan dan terdapat tombol verifikasi pada prestasi, dan jika ditekan akan muncul sebuah konfirmasi untuk verifikasi prestasi yang dipilih.



Gambar 6.8 Implementasi UI fitur verifikasi (bagian verifikasi)

Pada *Source Code 6.16*. Terdapat *function* verifikasi dengan algoritma dimulai dengan mengambil id data prestasi dan dikirim kepada API prestasi, kemudian merubah *response* berupa format JSON dan mengembalikan dengan memanggil *route* web /mahasiswa yang mengarah pada *controller* indexKemahasiswaan.

```
c_prestasi.php
```

```
public function verifikasi($id_prestasi){
    $client = new \GuzzleHttp\Client();
```

```

$url = env('Base url') .
'kemahasiswaan/'.$_GET['id_prestasi'].'/verifikasi';
$request = $client->put($url);
$response = json_decode($request->getBody());
return redirect('/kemahasiswaan');
}

```

Source Code 6.16 Implementasi kode fitur verifikasi (bagian verifikasi)

```

prestasiController.php

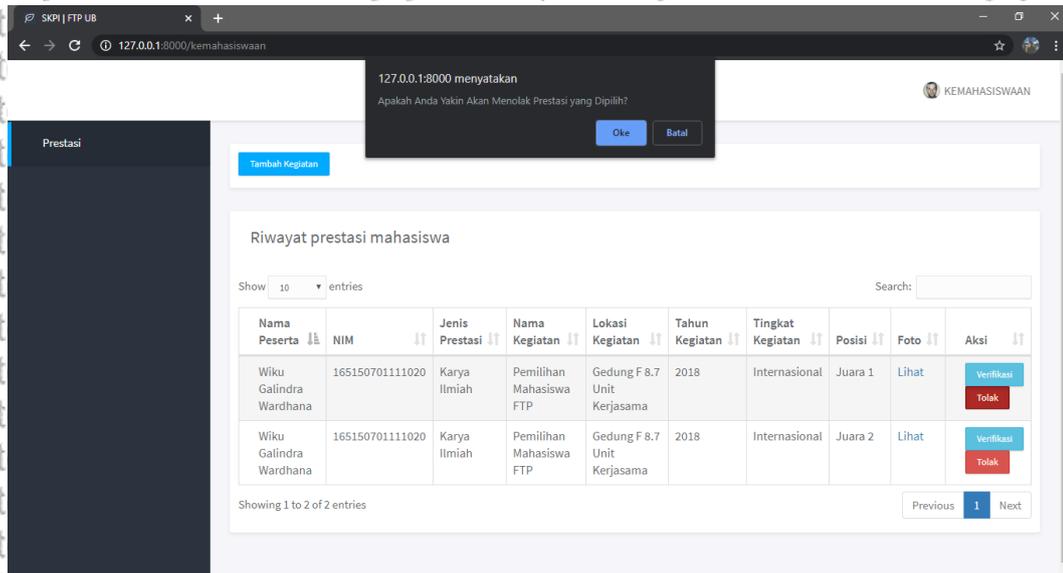
public function verifikasi($id_prestasi) {
    $prestasi = App\Models\Prestasi::find($id_prestasi);
    $prestasi->status = 'Verifikasi';
    $prestasi->save();
    return response()->json(['status'=>'berhasil']);
}

```

Source Code 6.17 Implementasi kode fitur verifikasi (bagian verifikasi) pada API prestasi

Pada *Source Code 6.17* terdapat *function* verifikasi dengan algoritma dimulai dengan mengambil id prestasi yang dikirim setelah itu dilakukan *update* status prestasi menjadi verifikasi dan mengembalikan *response* dengan status berhasil.

Pada fitur verifikasi (bagian tolak) seperti pada Gambar 6.9. Terdapat sebuah halaman kemahasiswaan dan terdapat tombol tolak pada prestasi, dan jika ditekan akan muncul sebuah konfirmasi untuk verifikasi prestasi yang dipilih.



Gambar 6.9 Implementasi UI fitur verifikasi (bagian tolak)

Pada *Source Code 6.18*. Terdapat *function* verifikasi dengan algoritma dimulai dengan mengambil id data prestasi dan dikirim kepada API prestasi, kemudian merubah *response* berupa format JSON dan mengembalikan dengan memanggil *route* web /mahasiswa yang mengarah pada *controller* indexKemahasiswaan.

```

c_prestasi.php

public function tolak($id_prestasi) {
    $client = new \GuzzleHttp\Client();
}

```

```

$url = env('Base url') .
'kemahasiswaan/' . $sid_prestasi . '/tolak';
$request = $client->get($url);
$response = json_decode($request->getBody());
return redirect('/kemahasiswaan');
}

```

Source Code 6.18 Implementasi kode fitur verifikasi (bagian tolak)

prestasiController.php

```

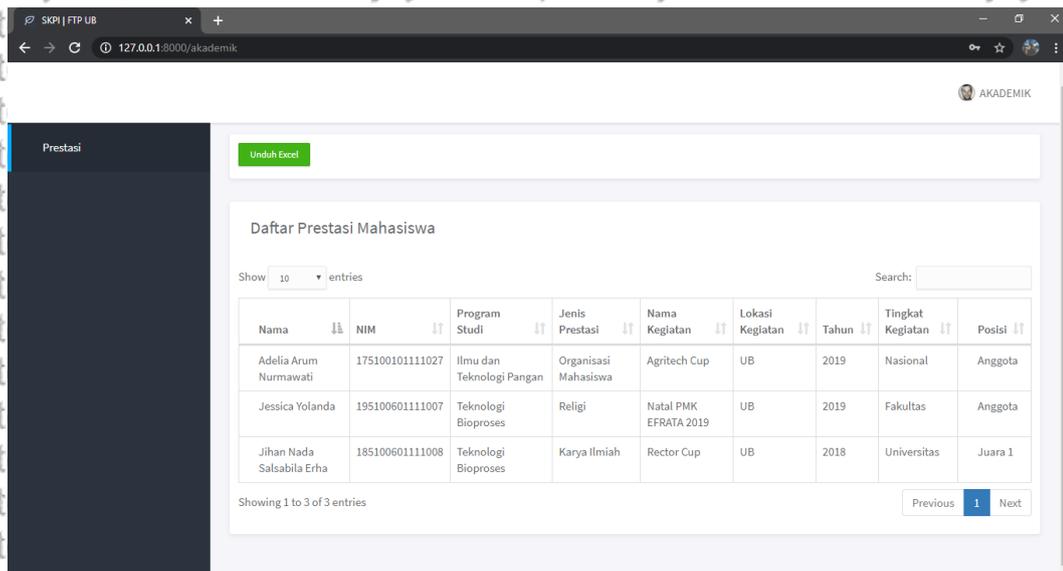
public function tolak($sid_prestasi) {
    $prestasi = \App\m_prestasi::find($sid_prestasi);
    $prestasi->delete();
    return response()->json(['status'=>'berhasil']);
}

```

Source Code 6.19 Implementasi kode fitur verifikasi (bagian tolak) pada API prestasi

Pada Source Code 6.19. Terdapat *function* verifikasi dengan algoritma dimulai dengan mengambil id prestasi yang dikirim setelah itu dilakukan *delete* pada *database* dan mengembalikan *response* dengan status berhasil.

6.8 Implementasi Lihat Daftar Prestasi



Gambar 6.10 Implementasi UI fitur lihat daftar prestasi

Pada fitur lihat daftar prestasi seperti pada Gambar 6.10. Terdapat sebuah halaman akademik yang terdiri dari tombol unduh *excel*. Selain itu terdapat Tabel yang berisikan informasi prestasi yang dengan status verifikasi.

c_prestasi.php

```

public function indexAkademik() {
    $client = new \GuzzleHttp\Client();
    $url = env('Base url') . 'akademik';
    $request = $client->get($url);
    $response = json_decode($request->getBody());
    return view('v_akademik', ['data_prestasi' => $response]);
}

```



Source Code 6.20 Implementasi kode fitur lihat daftar prestasi

Pada *Source Code 6.20*, Terdapat *function* `indexAkademik` dengan algoritma dimulai dengan mengambil data pada API prestasi, kemudian merubah *response* berupa format JSON menjadi *array* kemudian ditampilkan pada *view* `v_akademik`.

`prestasiController.php`

```
public function indexAkademik() {
    $data_prestasi = \App\m_prestasi::with('User')
        ->where('status', 'Verifikasi')->get();
    return response()->json($data_prestasi,200);
}
```

Source Code 6.21 Implementasi kode fitur lihat daftar prestasi pada API prestasi

Pada *Source Code 6.21*, Terdapat *function* `indexAkademik` dengan algoritma dimulai dengan mengambil data prestasi dengan status verifikasi dan data prestasi akan dikembalikan dalam bentuk *response* dengan format JSON.

6.9 Implementasi unduh Daftar Prestasi

The screenshot shows a web browser window displaying a page titled 'Prestasi' with a sub-header 'Daftar Prestasi Mahasiswa'. A green button labeled 'Unduh Excel' is visible at the top. Below the button is a table with columns: Nama, NIM, Program Studi, Jenis Prestasi, Nama Kegiatan, Lokasi Kegiatan, Tahun, Tingkat, and Posisi. The table contains three rows of data. At the bottom of the table, there is a 'Showing 1 to 3 of 3 entries' indicator and navigation buttons for 'Previous', '1', and 'Next'. A download icon and file name 'prestasi (7).xlsx' are visible at the bottom of the browser window.

Nama	NIM	Program Studi	Jenis Prestasi	Nama Kegiatan	Lokasi Kegiatan	Tahun	Tingkat	Posisi
Adelia Arum Nurmawati	175100101111027	Ilmu dan Teknologi Pangan	Organisasi Mahasiswa	Agritech Cup	UB	2019	Nasional	Anggota
Jessica Yolanda	195100601111007	Teknologi Bioproses	Religi	Natal PMK EFRATA 2019	UB	2019	Fakultas	Anggota
Jihan Nada Salsabila Erha	185100601111008	Teknologi Bioproses	Karya Ilmiah	Rector Cup	UB	2018	Universitas	Juara 1

Gambar 6.11 Implementasi UI fitur unduh daftar prestasi

Pada fitur mahasiswa unduh daftar prestasi seperti pada Gambar 6.11. Terdapat sebuah halaman akademik dan terdapat tombol unduh excel yang jika ditekan akan mengunduh data prestasi mahasiswa dengan status verifikasi dengan format file excel.

Pada *Source Code 6.22*, Terdapat *function* `exportExcel` dengan algoritma dimulai dengan memanggil API prestasi, kemudian merubah *response* berupa format JSON dan membuat *file excel* setelah itu mengembalikan hasil dengan mengunduh *file* bernama `prestasi.xlsx`.

`c_prestasi.php`

```

public function exportExcel() {
    $client = new \GuzzleHttp\Client();
    $url = env('Base_url').'/akademik/exportexcel';
    $request = $client->get($url);
    $response = json_decode($request->getBody());
    $excel = new prestasiExport($response);
    return Excel::download($excel, 'prestasi.xlsx');
}

```

Source Code 6.22 Implementasi kode fitur unduh daftar prestasi

prestasiController.php

```

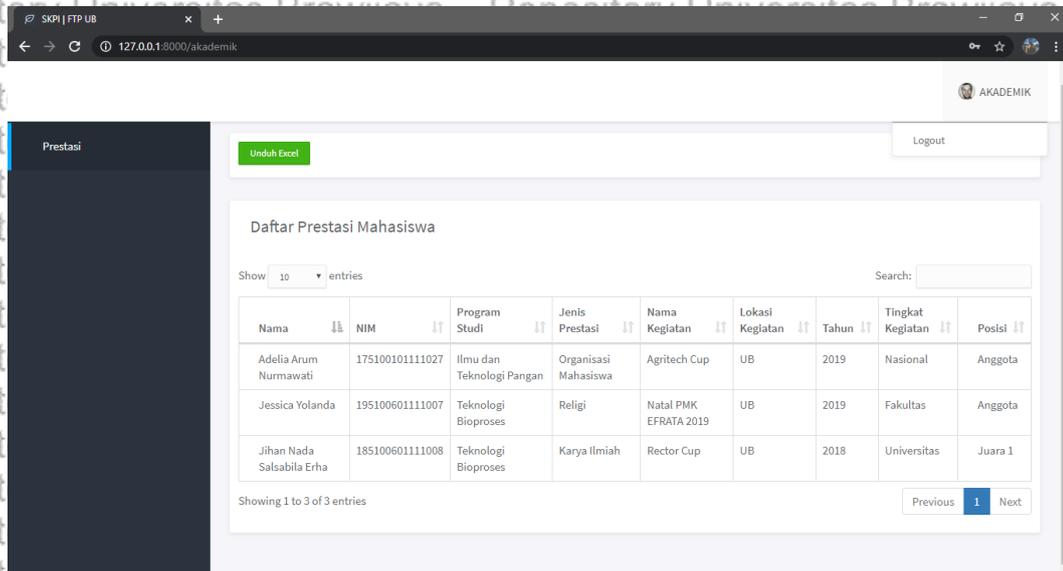
public function exportexcel() {
    $data_prestasi = \App\Models\prestasi::with('User')
        ->where('status', 'Verifikasi')->get();
    return response()->json($data_prestasi, 200);
}

```

Source Code 6.23 Implementasi kode fitur unduh daftar prestasi pada API prestasi

Pada *Source Code 6.23*. Terdapat *function exportexcel* dengan algoritma dimulai dengan mengambil data prestasi pada *database* yang direpresentasikan pada *model* *m_prestasi* dengan status verifikasi, setelah itu mengembalikan hasil data prestasi berupa JSON.

6.10 Implementasi Logout



Gambar 6.12 Implementasi UI fitur logout

Pada fitur *logout* seperti pada Gambar 6.12. Pada tiap halaman baik mahasiswa, kemahasiswaan, dan akademik terdapat tombol *logout* di bagian pojok kiri atas.

Pada *Source Code 6.24*. Terdapat *function logout* dengan algoritma dimulai dengan menghapus *session no_induk*, nama, prodi, fakultas, dan jabatan, setelah itu diarahkan pada halaman index yang mengarah pada halaman *login*.



c_aktor.php

```

public function logout() {
    session()->forget('no_induk');
    session()->forget('nama');
    session()->forget('prodi');
    session()->forget('jabatan');
    return redirect('/');
}

```

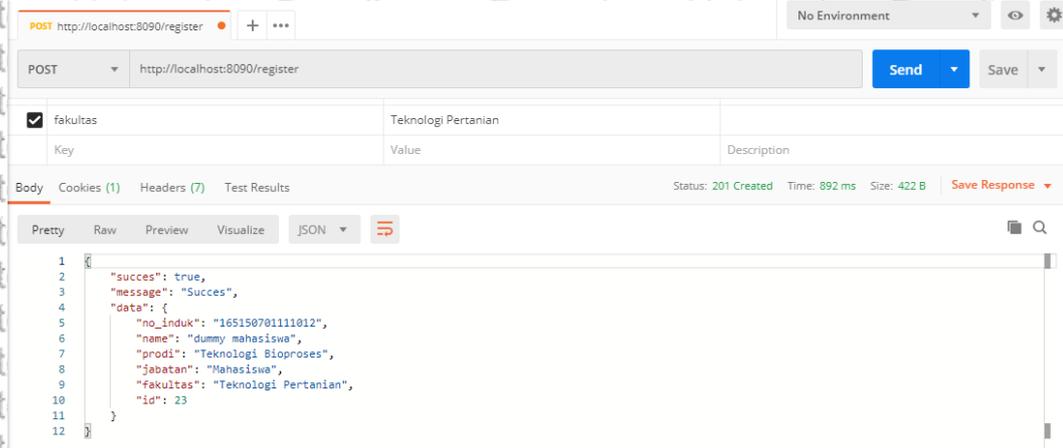
Source Code 6.24 Implementasi kode fitur logout

BAB 7 PENGUJIAN SISTEM

7.1 Pengujian REST API

Dilakukan pengujian API prestasi yang telah dibuat menggunakan bantuan aplikasi Postman dengan tujuan untuk mengetahui *response* dari API prestasi berdasarkan request yang dilakukan dari masing – masing fitur yang telah dibuat.

7.1.1 Pengujian Fungsi Register



Gambar 7.1 Pengujian fungsi register API prestasi

Gambar 7.1 menunjukkan hasil *request* kepada fungsi *register* pada API prestasi menggunakan aplikasi Postman. Tabel 7.1 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.1.

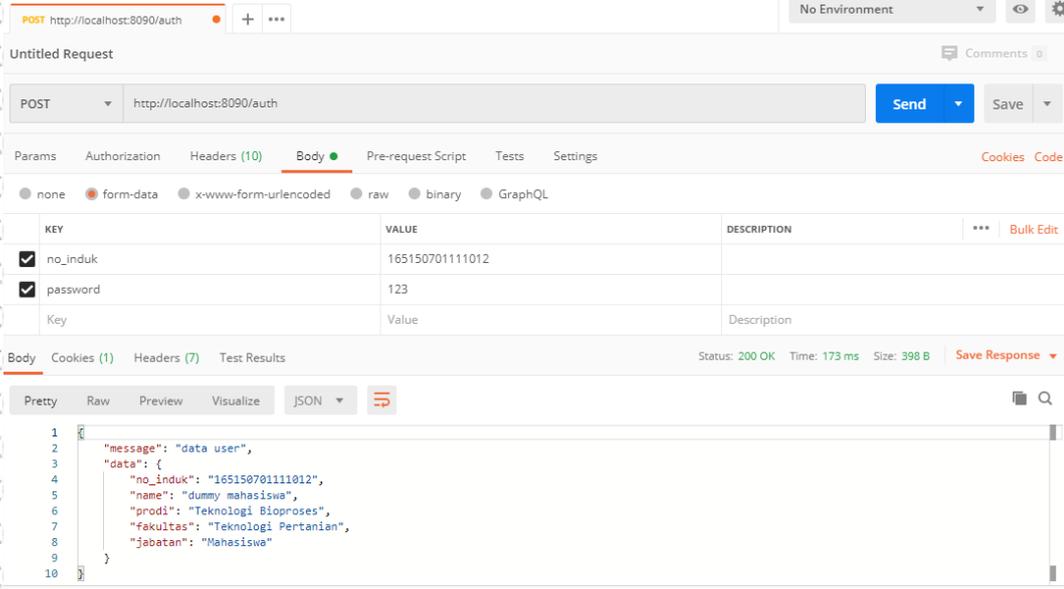
Tabel 7.1 Penjelasan pengujian fungsi register API prestasi

URL	http://localhost:8090/register
Method	POST
Request body	no_induk = 165150701111012 name = dummy mahasiswa password = 123 jabatan = Mahasiswa prodi = Teknologi Bioproses fakultas = Teknologi Pertanian
Response	
Status Code	201 Created
Content	{ "succes": true, "message": "Sukses", "data": { "no_induk": "165150701111012", "name": "dummy mahasiswa",

```

        "prodi": "Teknologi Bioproses",
        "jabatan": "Mahasiswa",
        "fakultas": "Teknologi Pertanian",
        "id": 23
    }
}
    
```

7.1.2 Pengujian Fungsi Auth



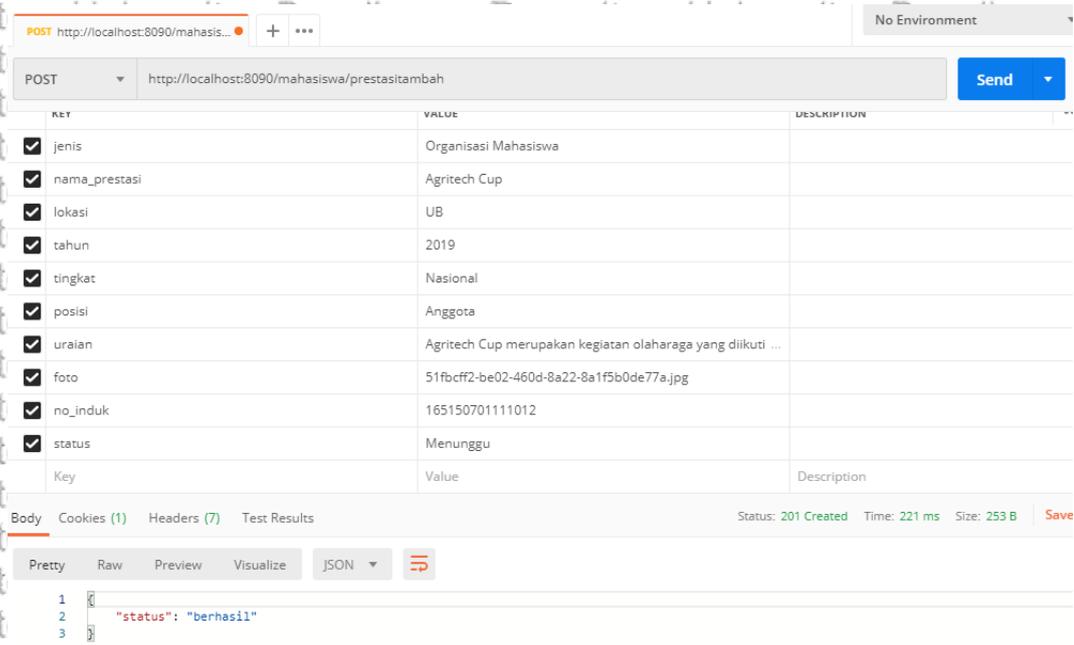
Gambar 7.2 pengujian fungsi *auth* API prestasi

Gambar 7.2 menunjukkan hasil *request* kepada fungsi *auth* pada API prestasi menggunakan aplikasi Postman. Tabel 7.2 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.2.

Tabel 7.2 Penjelasan pengujian fungsi *auth* API prestasi

URL	http://localhost:8090/auth
Method	POST
Request body	no_induk = 165150701111012 password = 123
Response	
Status Code	200 OK
Content	<pre> { "message": "data user", "data": { "no_induk": "165150701111012", "name": "dummy mahasiswa", "prodi": "Teknologi Bioproses", "fakultas": "Teknologi Pertanian", "jabatan": "Mahasiswa" } } </pre>

7.1.3 Pengujian Fungsi Tambah Prestasi



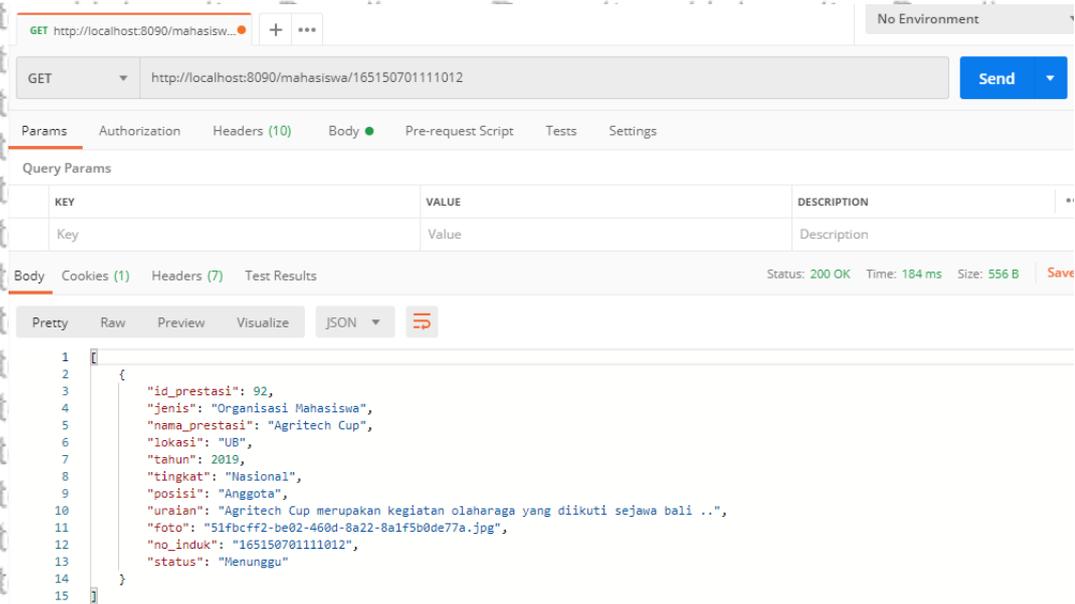
Gambar 7.3 pengujian fungsi tambah prestasi API prestasi

Gambar 7.3 menunjukkan hasil *request* kepada fungsi tambah prestasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.3 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.3.

Tabel 7.3 Penjelasan pengujian fungsi tambah prestasi API prestasi

URL	<code>http://localhost:8090/mahasiswa/prestasitambah</code>
Method	POST
Request body	<pre> jenis = Organisasi Mahasiswa nama_prestasi = Agritech Cup lokasi = UB tahun = 2019 tingkat = Nasional posisi = Anggota uraian = Agritech Cup merupakan kegiatan olahraga yang diikuti sejawa bali ... foto = 51fbcff2-be02-460d-8a22-8a1f5b0de77a.jpg no_induk = 165150701111012 status = Menunggu </pre>
Response	
Status Code	201 Created
Content	<pre> { "status": "berhasil" } </pre>

7.1.4 Pengujian Fungsi Lihat Prestasi Mahasiswa



Gambar 7.4 pengujian fungsi lihat prestasi mahasiswa API prestasi

Gambar 7.4 menunjukkan hasil *request* kepada fungsi lihat prestasi mahasiswa pada API prestasi menggunakan aplikasi Postman. Tabel 7.4 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.4. Pada pengujian ini melakukan lihat prestasi dari user dengan *no_induk* = 165150701111012 pada *database*.

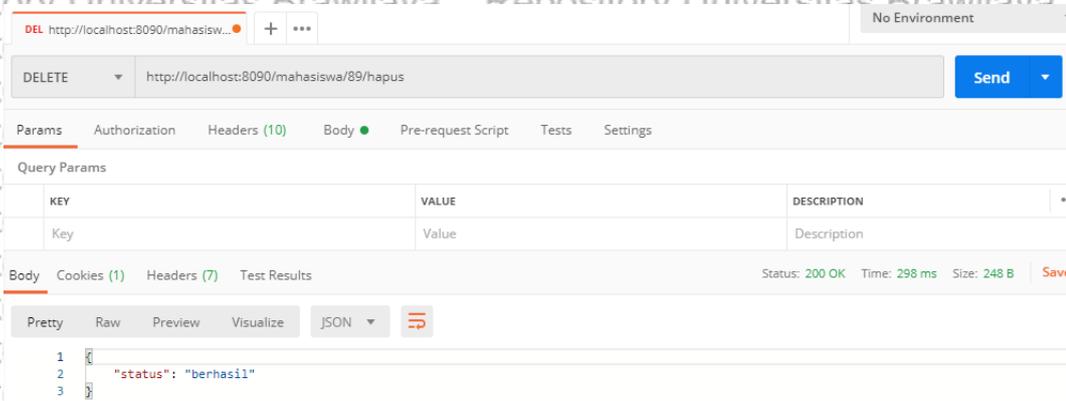
Tabel 7.4 Penjelasan pengujian fungsi lihat prestasi mahasiswa API prestasi

URL	http://localhost:8090/mahasiswa/165150701111012
Method	GET
Request body	
Response	
Status Code	200 OK
Content	[{ "id_prestasi": 92, "jenis": "Organisasi Mahasiswa", "nama_prestasi": "Agritech Cup", "lokasi": "UB", "tahun": 2019, "tingkat": "Nasional", "posisi": "Anggota", "uraian": "Agritech Cup merupakan kegiatan olahraga yang diikuti sejawat bali ..", "foto": "51fbcff2-be02-460d-8a22- 8a1f5b0de77a.jpg", "no_induk": "165150701111012", "status": "Menunggu"



```
} ]
```

7.1.5 Pengujian Fungsi Hapus Prestasi



Gambar 7.5 Pengujian fungsi hapus prestasi API prestasi

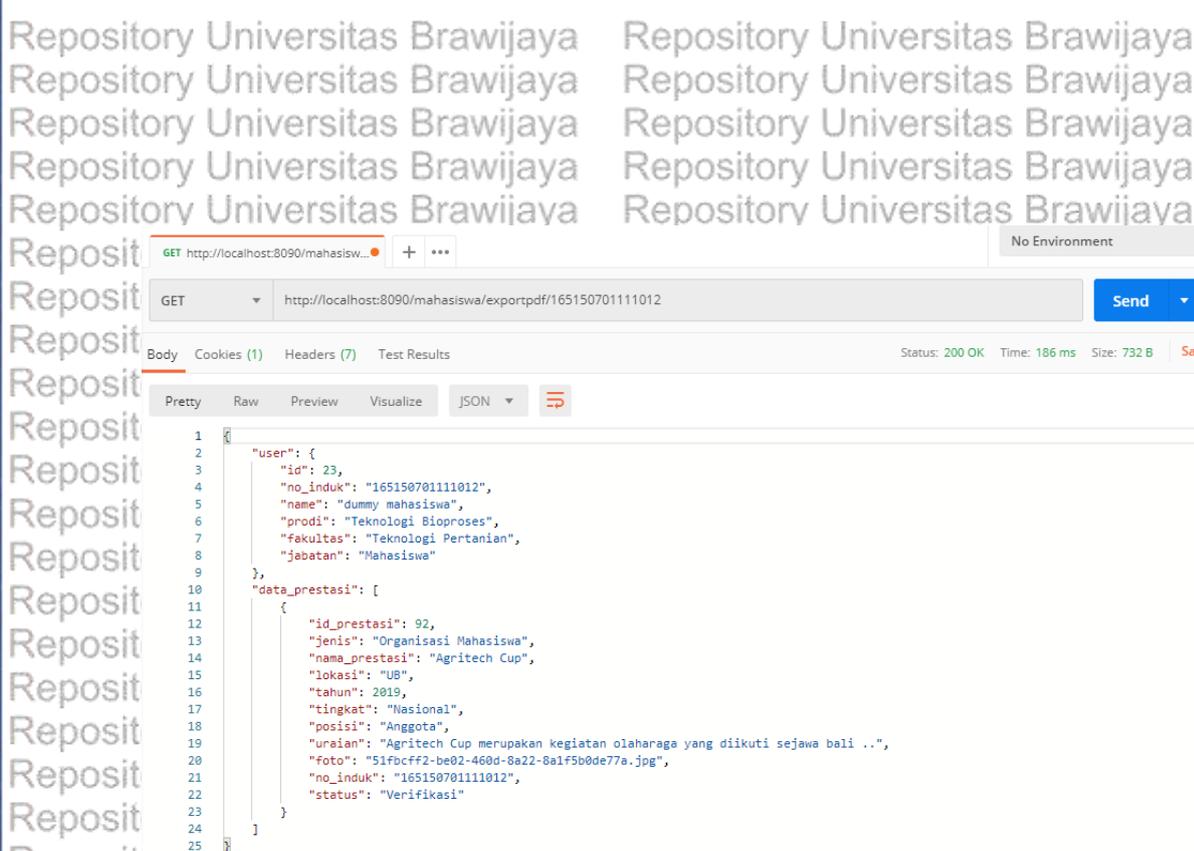
Gambar 7.5 menunjukkan hasil *request* kepada fungsi hapus prestasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.5 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.5. Pada pengujian ini dilakukan hapus prestasi dengan *id_prestasi* = 89 pada *database*.

Tabel 7.5 Penjelasan pengujian fungsi hapus prestasi API prestasi

<i>URL</i>	http://localhost:8090/mahasiswa/89/hapus
<i>Method</i>	DELETE
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "status": "berhasil" }

7.1.6 Pengujian Fungsi Unduh Draft

Gambar 7.5 menunjukkan hasil *request* kepada fungsi unduh draft pada API prestasi menggunakan aplikasi Postman. Tabel 7.5 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.5. Pada pengujian ini dilakukan untuk mendapatkan data user dengan *no_induk* = 165150701111012 dan prestasi dengan status prestasi verifikasi pada *database*.



Gambar 7.6 Pengujian fungsi unduh *draft* API prestasi

Tabel 7.6 Penjelasan pengujian fungsi unduh *draft* API prestasi

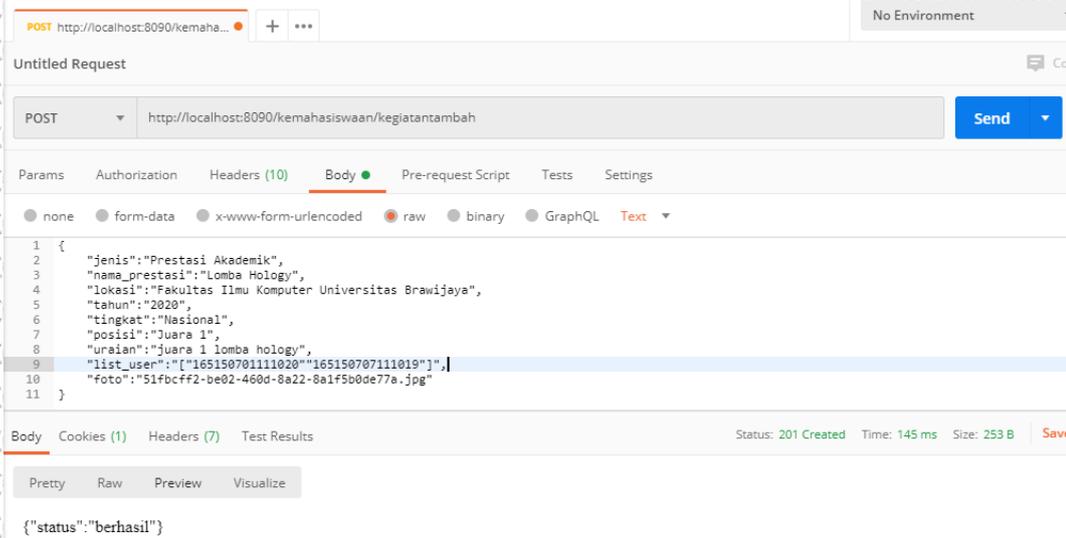
URL	http://localhost:8090/mahasiswa/exportpdf/165150701111012
Method	GET
Request body	
Response	
Status Code	200 OK
Content	<pre> { "user": { "id": 23, "no_induk": "165150701111012", "name": "dummy mahasiswa", "prodi": "Teknologi Bioproses", "fakultas": "Teknologi Pertanian", "jabatan": "Mahasiswa" }, "data_prestasi": [{ "id_prestasi": 92, "jenis": "Organisasi Mahasiswa", "nama_prestasi": "Agritech Cup", "lokasi": "UB", "tahun": 2019, "tingkat": "Nasional", "posisi": "Anggota", "uraian": "Agritech Cup merupakan kegiatan olahraga yang diikuti sejawa bali ..", </pre>

```

"foto": "51fbcff2-be02-460d-8a22-8a1f5b0de77a.jpg",
"no_induk": "165150701111012",
"status": "Verifikasi"
}}

```

7.1.7 Pengujian Fungsi Tambah Kegiatan



Gambar 7.7 pengujian fungsi tambah kegiatan API prestasi

Gambar 7.3 menunjukkan hasil *request* kepada fungsi tambah kegiatan pada API prestasi menggunakan aplikasi Postman. Tabel 7.3 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.3. Dilakukan *request* dengan *body raw* karena dilakukan *request* dalam bentuk *array* pada atribut *no_induk* yang direpresentasikan pada atribut *list_user*.

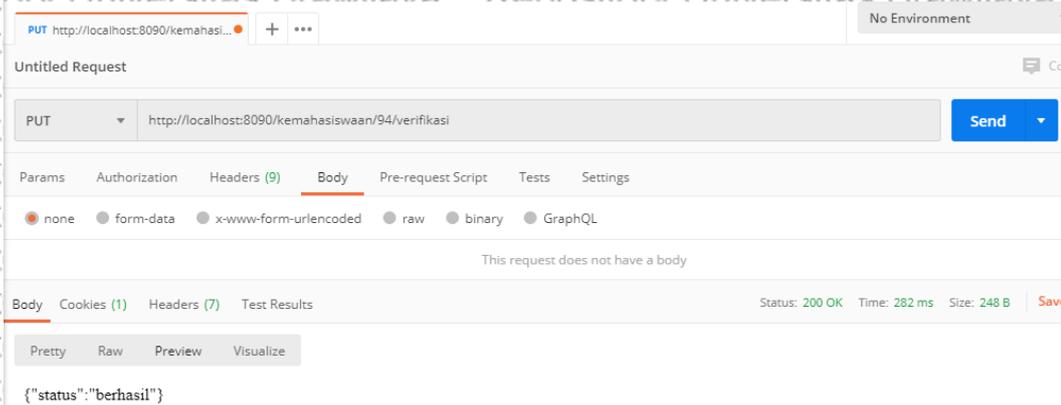
Tabel 7.7 Penjelasan pengujian fungsi tambah kegiatan API prestasi

URL	http://localhost:8090/kemahasiswaan/kegiatanTambah
Method	POST
Request body	<pre> { "jenis": "Prestasi Akademik", "nama_prestasi": "Lomba Hology", "lokasi": "Fakultas Ilmu Komputer Universitas Brawijaya", "tahun": "2020", "tingkat": "Nasional", "posisi": "Juara 1", "uraian": "juara 1 lomba hology", "list_user": "[\"165150701111020\", \"165150701111019\"]", "foto": "51fbcff2-be02-460d-8a22-8a1f5b0de77a.jpg" } </pre>



<i>Response</i>	
<i>Status Code</i>	201 Created
<i>Content</i>	{ "status": "berhasil" }

7.1.8 Pengujian Fungsi Verifikasi Prestasi



Gambar 7.8 Pengujian fungsi verifikasi prestasi API prestasi

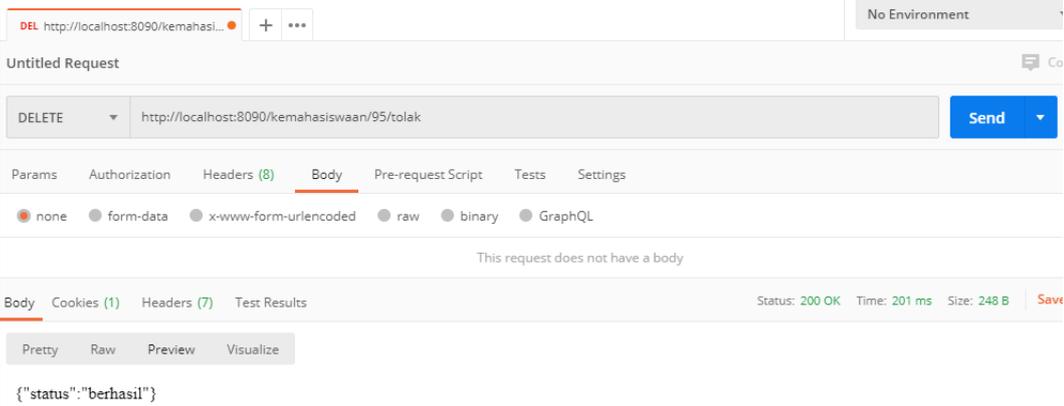
Gambar 7.8 menunjukkan hasil *request* kepada fungsi verifikasi prestasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.8 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.8. Pada pengujian ini dilakukan update status prestasi menjadi verifikasi dengan id_prestasi = 94 pada *database*.

Tabel 7.8 Penjelasan pengujian fungsi verifikasi prestasi API prestasi

<i>URL</i>	http://localhost:8090/kemahasiswaan/94/verifikasi
<i>Method</i>	PUT
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	{ "status": "berhasil" }

7.1.9 Pengujian Fungsi Tolak Prestasi

Gambar 7.8 menunjukkan hasil *request* kepada fungsi verifikasi prestasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.8 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.8. Pada pengujian ini dilakukan hapus prestasi dengan id_prestasi = 95 pada *database*.

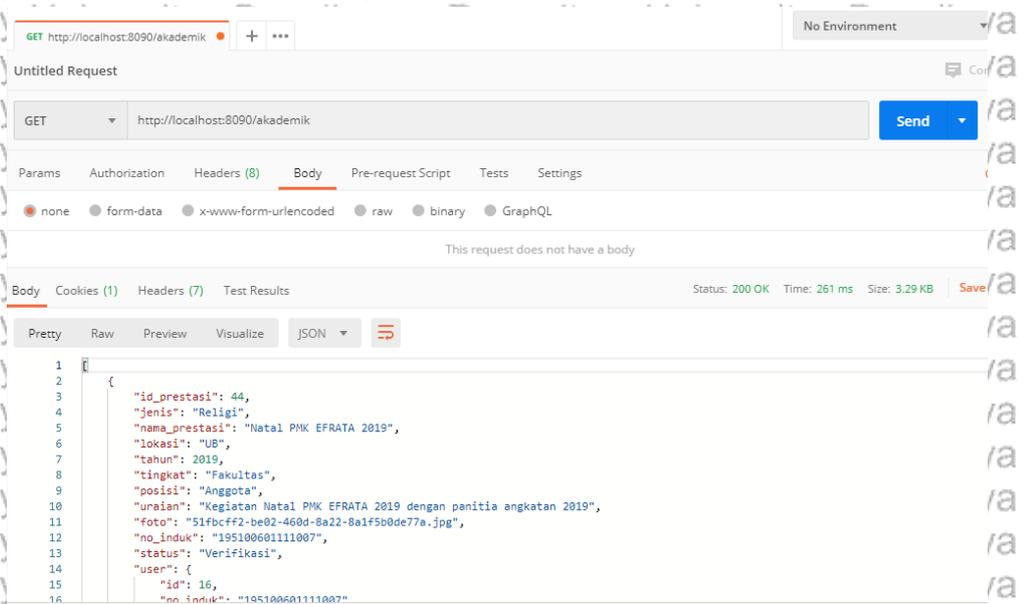


Gambar 7.9 Pengujian fungsi tolak prestasi API prestasi

Tabel 7.9 Penjelasan pengujian fungsi tolak prestasi API prestasi

URL	http://localhost:8090/kemahasiswaan/95/tolak
Method	DELETE
Request body	
Response	
Status Code	200 OK
Content	{ "status": "berhasil" }

7.1.10 Pengujian Fungsi Lihat Prestasi Verifikasi



Gambar 7.10 pengujian fungsi lihat prestasi verifikasi API prestasi

Gambar 7.10 menunjukkan hasil *request* kepada fungsi lihat prestasi verifikasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.10 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.10.

Tabel 7.10 Penjelasan pengujian fungsi lihat prestasi verifikasi API prestasi

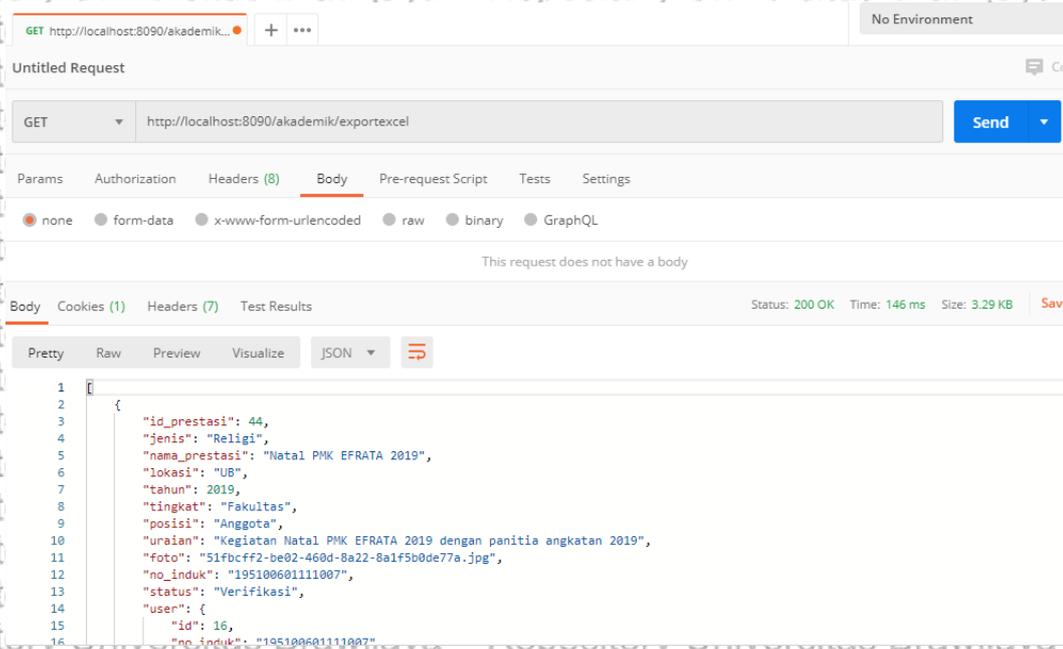
<i>URL</i>	http://localhost:8090/akademik
<i>Method</i>	GET
<i>Request body</i>	
<i>Response</i>	
<i>Status Code</i>	200 OK
<i>Content</i>	<pre>[{ "id_prestasi": 44, "jenis": "Religi", "nama_prestasi": "Natal PMK EFRATA 2019", "lokasi": "UB", "tahun": 2019, "tingkat": "Fakultas", "posisi": "Anggota", "uraian": "Kegiatan Natal PMK EFRATA 2019 dengan panitia angkatan 2019", "foto": "51fbcff2-be02-460d-8a22- 8alf5b0de77a.jpg", "no_induk": "195100601111007", "status": "Verifikasi", "user": { "id": 16, "no_induk": "195100601111007", "name": "Jessica Yolanda", "prodi": "Teknologi Bioproses", "fakultas": "Teknologi Pertanian", "jabatan": "Mahasiswa" } }, { "id_prestasi": 94, "jenis": "Karya Ilmiah", "nama_prestasi": "Sertifikasi IT", "lokasi": "UB", "tahun": 2018, "tingkat": "Internasional", "posisi": "Juara 1", "uraian": "a", "foto": "sertifikat 1.jpg", "no_induk": "165150707111019", "status": "Verifikasi", "user": { "id": 14, "no_induk": "165150707111019",</pre>

```

    "name": "ABI FIRMANDHANI",
    "prodi": "Teknologi Informasi",
    "fakultas": "Fakultas Ilmu Komputer",
    "jabatan": "Mahasiswa"
  }
}
]

```

7.1.11 Pengujian Fungsi unduh Prestasi Verifikasi



Gambar 7.11 pengujian fungsi unduh daftar prestasi API prestasi

Gambar 7.11 menunjukkan hasil *request* kepada fungsi unduh daftar prestasi verifikasi pada API prestasi menggunakan aplikasi Postman. Tabel 7.11 menjelaskan hasil pengujian berupa *response* API prestasi berdasarkan *request* yang dilakukan pada gambar 7.11. dari hasil daftar prestasi yang menjadi *response* dari API akan dilakukan *export* kedalam bentuk *excel* pada *web client*.

Tabel 7.11 Penjelasan pengujian unduh daftar prestasi verifikasi API prestasi

URL	http://localhost:8090/akademik/exportexcel
Method	GET
Request body	
Response	
Status Code	200 OK
Content	<pre> { "id_prestasi": 44, "jenis": "Religi", "nama_prestasi": "Natal PMK EFRATA 2019", </pre>



```

"lokasi": "UB",
"tahun": 2019,
"tingkat": "Fakultas",
"posisi": "Anggota",
"uraian": "Kegiatan Natal PMK EFRATA 2019
          dengan panitia angkatan 2019",
"foto": "51fbcff2-be02-460d-8a22-
          8a1f5b0de77a.jpg",
"no_induk": "195100601111007",
"status": "Verifikasi",
"user": {
  "id": 16,
  "no_induk": "195100601111007",
  "name": "Jessica Yolanda",
  "prodi": "Teknologi Bioproses",
  "fakultas": "Teknologi Pertanian",
  "jabatan": "Mahasiswa"
}
},
{
  "id_prestasi": 94,
  "jenis": "Karya Ilmiah",
  "nama_prestasi": "Sertifikasi IT",
  "lokasi": "UB",
  "tahun": 2018,
  "tingkat": "Internasional",
  "posisi": "Juara 1",
  "uraian": "a",
  "foto": "sertifikat 1.jpg",
  "no_induk": "165150707111019",
  "status": "Verifikasi",
  "user": {
    "id": 14,
    "no_induk": "165150707111019",
    "name": "ABI FIRMANDHANI",
    "prodi": "Teknologi Informasi",
    "fakultas": "Fakultas Ilmu Komputer",
    "jabatan": "Mahasiswa"
  }
}
}

```

7.2 Pengujian Blackbox

Setelah sistem selesai dilakukan implementasi maka dilakukan pengujian *blackbox* dengan tujuan untuk mengetahui apakah dari masing – masing fitur telah berjalan sesuai dengan kebutuhan, selain itu pada pengujian *blackbox* juga dilakukan untuk melakukan pengecekan kesalahan teknis dari fitur ketika dijalankan. Pada pengujian ini dilakukan oleh bapak Mas'ud Effendi, STP., MP. Selaku ketua PSIK Fakultas Teknologi Pertanian.



Tabel 7.12 Pengujian login

Pengujian login	
Aktor	Mahasiswa, Kemahasiswaan, Akademik
Kasus pengujian	Memasukkan nomor induk dan <i>password</i>
Hasil yang diharapkan	Aktor dapat masuk ke dalam halaman utama (jika mahasiswa masuk halaman mahasiswa, kemahasiswaan masuk halaman kemahasiswaan, akademik masuk halaman akademik)
Hasil pengujian	Aktor akan diarahkan pada halaman utama (mahasiswa diarahkan pada halaman mahasiswa, kemahasiswaan diarahkan pada halaman kemahasiswaan, akademik diarahkan pada halaman akademik)
Status	Valid

Pada Tabel 7.12 dijelaskan mengenai pengujian *login* yang dilakukan oleh aktor mahasiswa, kemahasiswaan, dan akademik. Pengujian diawali dengan aktor memasukkan nomor induk serta *password*. Setelah itu aktor akan diarahkan kepada halaman utama (halaman mahasiswa untuk mahasiswa, halaman kemahasiswaan untuk kemahasiswaan, dan halaman akademik untuk akademik). Pada pengujian ini mendapatkan status valid.

Tabel 7.13 Pengujian mahasiswa tambah prestasi

Pengujian mahasiswa tambah prestasi	
Aktor	Mahasiswa
Kasus pengujian	Menekan tombol Tambah Prestasi, kemudian mengisi data prestasi pada form berupa jenis prestasi, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan, posisi/jabatan, uraian kegiatan/prestasi, serta unggah dokumen pendukung berupa Gambar kemudian menekan tombol simpan
Hasil yang diharapkan	Prestasi berhasil ditambahkan dan muncul pada tabel prestasi dengan status menunggu
Hasil pengujian	Prestasi berhasil ditambahkan dan muncul pada tabel prestasi dengan status menunggu
Status	Valid

Pada Tabel 7.13 dijelaskan mengenai pengujian mahasiswa tambah prestasi yang dilakukan oleh aktor mahasiswa. Pengujian dilakukan dengan menekan tombol Tambah Prestasi, kemudian mengisi data prestasi pada form berupa jenis prestasi, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan, posisi/jabatan, uraian kegiatan/prestasi, serta unggah dokumen pendukung berupa Gambar kemudian menekan tombol simpan. Setelah itu prestasi berhasil ditambahkan dan muncul pada Tabel prestasi dengan status menunggu. Pada pengujian ini mendapatkan status valid.

**Tabel 7.14 Pengujian mahasiswa lihat prestasi**

Pengujian mahasiswa lihat prestasi	
Aktor	Mahasiswa
Kasus pengujian	Telah melakukan <i>login</i> sebagai mahasiswa
Hasil yang diharapkan	Menampilkan daftar prestasi mahasiswa setelah masuk sistem
Hasil pengujian	Menampilkan daftar prestasi mahasiswa setelah masuk sistem
Status	Valid

Pada Tabel 7.14 dijelaskan mengenai pengujian mahasiswa lihat prestasi yang dilakukan oleh aktor mahasiswa. Pengujian dilakukan setelah melakukan *login* sebagai mahasiswa, maka sistem akan menampilkan daftar prestasi mahasiswa yang telah masuk sistem sebelumnya. Pada pengujian ini mendapatkan status valid.

Tabel 7.15 Pengujian mahasiswa hapus prestasi

Pengujian mahasiswa hapus prestasi	
Aktor	Mahasiswa
Kasus pengujian	Menekan tombol Hapus pada salah satu prestasi, setelah itu akan ada peringatan untuk menghapus prestasi kemudian tekan ok
Hasil yang diharapkan	Prestasi terpilih terhapus dari tabel prestasi dan <i>database</i>
Hasil pengujian	Prestasi terpilih terhapus dari tabel prestasi dan <i>database</i>
Status	Valid

Pada Tabel 7.15 dijelaskan mengenai pengujian mahasiswa hapus prestasi yang dilakukan oleh aktor mahasiswa. Pengujian dilakukan dengan menekan tombol Hapus pada salah satu prestasi, setelah itu akan ada peringatan untuk menghapus prestasi kemudian tekan ok. Setelah itu prestasi terpilih terhapus dari Tabel prestasi dan *database*. Pada pengujian ini mendapatkan status valid.

Tabel 7.16 Pengujian mahasiswa unduh draft

Pengujian mahasiswa unduh draft	
Aktor	Mahasiswa
Kasus pengujian	Menekan tombol unduh <i>draft</i>
Hasil yang diharapkan	Mengunduh berkas <i>draft</i> prestasi mahasiswa dalam format pdf
Hasil pengujian	Mengunduh berkas <i>draft</i> prestasi mahasiswa dalam format pdf
Status	Valid



Pada Tabel 7.16 dijelaskan mengenai pengujian mahasiswa unduh *draft* yang dilakukan oleh aktor mahasiswa. Pengujian dilakukan dengan Menekan tombol unduh *draft*. Setelah itu sistem akan mengunduh berkas *draft* prestasi mahasiswa dalam format pdf. Pada pengujian ini mendapatkan status valid.

Tabel 7.17 Pengujian kemahasiswaan tambah kegiatan

Pengujian kemahasiswaan tambah kegiatan	
Aktor	Kemahasiswaan
Kasus pengujian	Menekan tombol tambah Kegiatan, kemudian mengisi data prestasi pada form berupa jenis prestasi, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan, posisi/jabatan, nama peserta, uraian kegiatan/prestasi, serta unggah dokumen pendukung berupa Gambar kemudian menekan tombol submit
Hasil yang diharapkan	Prestasi berhasil ditambahkan dengan status verifikasi
Hasil pengujian	Prestasi berhasil ditambahkan dengan status verifikasi
Status	Valid

Pada Tabel 7.17 dijelaskan mengenai pengujian mahasiswa tambah kegiatan yang dilakukan oleh aktor kemahasiswaan. Pengujian dilakukan dengan menekan tombol Tambah Prestasi, kemudian mengisi data prestasi pada form berupa jenis prestasi, nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kegiatan, posisi/jabatan, nama peserta, uraian kegiatan/prestasi, serta unggah dokumen pendukung berupa Gambar kemudian menekan tombol simpan. Setelah itu prestasi berhasil ditambahkan dengan status verifikasi. Pada pengujian ini mendapatkan status valid.

Tabel 7.18 Pengujian verifikasi (verifikasi)

Pengujian verifikasi (verifikasi)	
Aktor	Kemahasiswaan
Kasus pengujian	Menekan tombol verifikasi pada salah satu prestasi, setelah itu akan ada peringatan untuk verifikasi prestasi kemudian tekan ok
Hasil yang diharapkan	Status prestasi berubah menjadi verifikasi
Hasil pengujian	Status prestasi berubah menjadi verifikasi
Status	Valid

Pada Tabel 7.18 dijelaskan mengenai pengujian mahasiswa verifikasi bagian verifikasi yang dilakukan oleh aktor kemahasiswaan. Dilakukan pengujian dengan cara menekan tombol verifikasi pada salah satu prestasi, setelah itu akan ada



peringatan untuk verifikasi prestasi kemudian tekan ok. Setelah itu status prestasi berubah menjadi verifikasi. Pada pengujian ini mendapatkan status valid.

Tabel 7.19 Pengujian verifikasi (tolak)

Pengujian verifikasi (tolak)	
Aktor	Kemahasiswaan
Kasus pengujian	Menekan tombol tolak pada salah satu prestasi, setelah itu akan ada peringatan untuk menolak prestasi kemudian tekan ok
Hasil yang diharapkan	Prestasi terpilih terhapus dari tabel prestasi dan <i>database</i>
Hasil pengujian	Prestasi terpilih terhapus dari tabel prestasi dan <i>database</i>
Status	Valid

Pada Tabel 7.19 dijelaskan mengenai pengujian mahasiswa verifikasi bagian tolak yang dilakukan oleh aktor kemahasiswaan. Dilakukan pengujian dengan cara menekan tombol tolak pada salah satu prestasi, setelah itu akan ada peringatan untuk menolak prestasi kemudian tekan ok. Setelah itu prestasi terpilih terhapus dari Tabel prestasi dan *database*. Pada pengujian ini mendapatkan status valid.

Tabel 7.20 Pengujian lihat daftar prestasi

Pengujian lihat daftar prestasi	
Aktor	Akademik
Kasus pengujian	Telah melakukan <i>login</i> sebagai akademik
Hasil yang diharapkan	Sistem menampilkan daftar prestasi mahasiswa dengan status verifikasi
Hasil pengujian	Menampilkan daftar prestasi mahasiswa dengan status verifikasi
Status	Valid

Pada Tabel 7.20 dijelaskan mengenai pengujian mahasiswa lihat daftar prestasi yang dilakukan oleh aktor akademik. Pengujian dilakukan setelah melakukan *login* sebagai akademik, maka sistem akan menampilkan daftar prestasi mahasiswa dengan status verifikasi. Pada pengujian ini mendapatkan status valid.

Tabel 7.21 Pengujian unduh daftar prestasi

Pengujian unduh daftar prestasi	
Aktor	Akademik
Kasus pengujian	Menekan tombol unduh <i>excel</i>
Hasil yang diharapkan	Mengunduh berkas prestasi mahasiswa dalam format <i>excel</i>



Hasil pengujian	Mengunduh berkas prestasi mahasiswa dalam format excel/
Status	Valid

Pada Tabel 7.21 dijelaskan mengenai pengujian mahasiswa unduh *excel* yang dilakukan oleh aktor akademik. Pengujian dilakukan dengan Menekan tombol unduh *excel*. Sistem akan secara otomatis mengunduh berkas prestasi mahasiswa dalam format *excel*. Pada pengujian ini mendapatkan status valid.

Tabel 7.22 Pengujian *logout*

Pengujian <i>logout</i>	
Aktor	Mahasiswa, Kemahasiswaan, Akademik
Kasus pengujian	Menekan nama aktor kemudian menekan tombol <i>logout</i>
Hasil yang diharapkan	Menampilkan halaman <i>login</i>
Hasil pengujian	Menampilkan halaman <i>login</i>
Status	Valid

Pada Tabel 7.22 menjelaskan pengujian *logout* yang dilakukan oleh aktor mahasiswa, kemahasiswaan, dan akademik. Pengujian dilakukan dengan Menekan nama aktor kemudian menekan tombol *logout*. Setelah itu sistem akan menampilkan halaman *login*. Pada pengujian ini mendapatkan status valid.

Pada pengujian terhadap sistem informasi perekaman prestasi mahasiswa yang telah dilakukan mendapatkan hasil sebagai berikut.

$$\text{Presentase} = (\text{Total Kasus berstatus Valid} / \text{Total Kasus yang diujikan}) \times 100\%$$

$$\text{Presentase} = (11/11) \times 100\%$$

$$\text{Presentase} = 100\% \text{ (Valid)}$$

Dapat disimpulkan bahwa sistem informasi perekaman prestasi mahasiswa dapat diterima karena telah berhasil dilakukan pengujian serta memenuhi karakteristik dan fungsi yang telah didefinisikan.

7.3 Pengujian *Usability*

Setelah dilakukan pengujian unit untuk menguji apakah ada kesalahan dari segi fitur yang dikembangkan, dilakukan pengujian *usability* dengan menggunakan SUS (*System Usability Scale*) kepada masing – masing aktor yang terdiri dari mahasiswa Fakultas Teknologi Pertanian, Kemahasiswaan Fakultas Teknologi Pertanian, dan Akademik Fakultas Teknologi Pertanian. Pengujian ini dilakukan dengan tujuan untuk mengetahui tingkat kemampuan sistem informasi perekaman prestasi mahasiswa ketika dijalankan. Tabel 7.23 menunjukkan tentang hasil dari pengujian *System Usability Scale (SUS)*. Pada Tabel 7.23 terdapat kolom P1 hingga P10 yang

berisikan jawaban dari pernyataan 1 hingga 10 yang ditunjukkan kepada penguji, daftar pernyataan dapat ditemukan pada Tabel 2.5. Tabel 7.24 menunjukkan tentang hasil dari perhitungan pengujian *System Usability Scale (SUS)*.

Tabel 7.23 Hasil Pengujian *System Usability Scale (SUS)*

Responden	Jabatan	Skor Asli									
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	Mahasiswa	5	1	5	1	5	1	5	1	5	2
2	Mahasiswa	5	2	4	2	4	1	4	1	5	4
3	Mahasiswa	4	1	5	3	4	2	5	1	4	4
4	Mahasiswa	5	2	4	1	4	1	5	2	4	2
5	Mahasiswa	5	2	4	2	4	1	5	2	4	1
6	Mahasiswa	4	2	5	2	4	2	4	2	4	2
7	Kemahasiswaan	4	2	4	3	4	3	3	2	3	4
8	Akademik	4	3	3	2	5	3	4	3	3	5

Setelah dilakukan pengujian kepada masing – masing aktor sistem, maka dilakukan perhitungan skor SUS dari data responden. Dengan aturan perhitungan sebagai berikut:

1. Pertanyaan nomor ganjil, skor yang didapat dikurangi 1.
2. Pertanyaan nomor genap, nilai 5 dikurangi skor yang didapat.
3. Skor dari setiap pertanyaan dijumlahkan kemudian dikali 2,5 untuk mendapatkan skor SUS.

Setelah didapat nilai skor SUS pada masing – masing aktor maka dicari skor rata – rata dengan menjumlahkan semua nilai skor SUS kemudian dibagi dengan jumlah responden. Untuk hasil perhitungan skor SUS dapat dilihat pada Tabel 7.24.

Tabel 7.24 Hasil perhitungan *System usability scale*

Skor Hasil Hitung										Jumlah	Nilai (Jumlah x 2,5)
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
4	4	4	4	4	4	4	4	4	3	39	98
4	3	3	3	3	4	3	4	4	1	32	80
3	4	4	2	3	3	4	4	3	1	31	78
4	3	3	4	3	4	4	3	3	3	34	85
4	3	3	3	3	4	4	3	3	4	34	85
3	3	4	3	3	3	3	3	3	3	31	78
3	3	3	2	3	2	2	3	2	1	24	60
3	2	2	3	4	2	3	2	2	0	23	58
Skor Rata-rata (Hasil Akhir)											78

Berdasarkan skor SUS yang tertera pada Gambar 2.4 dan skor rata – rata pengujian sistem informasi perekaman prestasi mahasiswa yang mendapatkan



nilai 78. dari skor tersebut sistem informasi perekaman prestasi mahasiswa masuk kedalam kategori *Good* dengan *grade scale C*. Dengan *acceptability ranges* berada pada *acceptable* atau dapat diterima.



BAB 8 KESIMPULAN DAN SARAN

8.1 Kesimpulan

Berdasarkan hasil dari pengembangan sistem informasi perekaman prestasi mahasiswa. Dapat ditarik kesimpulan sebagai berikut:

1. Proses analisa kebutuhan sistem informasi perekaman prestasi mahasiswa mendapatkan 10 kebutuhan fungsional. Dengan aktor terbagi menjadi menjadi 3 yaitu mahasiswa, kemahasiswaan, dan akademik. Serta dilakukan presentasi *prototype* kepada ketua PSIK FTP sebanyak 2 kali. Selanjutnya dibuat *use case diagram* dan *use case scenario* yang kemudian digambarkan ke dalam *activity diagram*.
2. Proses perancangan sistem informasi informasi perekaman prestasi mahasiswa dibuat berdasarkan kebutuhan yang telah dimodelkan. Dimulai dari membuat arsitektur sistem kemudian dibuat perancangan sistem dimodelkan kedalam *sequence diagram*, *class diagram*, dan *physical data model*. Selain itu dilakukan perancangan REST API.
3. Pada proses implementasi sistem informasi perekaman prestasi mahasiswa dibuat berdasarkan hasil dari rancangan sistem yang telah dibuat. Pada proses ini dilakukan implementasi *website* dengan menggunakan *framework* Laravel dan REST API dengan menggunakan *framework* Lumen.
4. Proses pengujian sistem dilakukan dalam 3 tahap. Dimulai dengan pengujian terhadap REST API pada setiap fitur yang telah diimplementasi pada API prestasi dengan bantuan aplikasi Postman. Pada pengujian *blackbox* menunjukkan bahwa 100% fitur yang diminta bisa dijalankan. Pada pengujian *usability* dilakukan kepada 8 responden dengan hasil mendapatkan kategori *Good* dengan *grade scale* C. Dengan *acceptability ranges* berada pada dapat diterima.

8.2 Saran

Dari penelitian mengenai sistem informasi perekaman prestasi mahasiswa yang telah dilakukan, saran yang diberikan untuk penelitian selanjutnya antara lain:

1. Pada penelitian ini masih belum dikembangkan keamanan pada REST API yang telah diimplementasi, harapan kedepannya dapat ditambahkan sistem keamanan pada REST API dengan menggunakan *JSON Web Token* (JWT) atau menggunakan OAuth 2.
2. Dilakukan penelitian dari sisi *mobile apps (mobile client)*, dikarenakan dari hasil wawancara dengan ketua PSIK FTP berharap sistem dapat berjalan pada *platform mobile*.



DAFTAR PUSTAKA

- Afsari K., Eastman C. M., & Lacouture D. C., 2017. *JavaScript Object Nation (JSON) data serialization for IFC schema in web-based BIM data exchange*. [e-journal]. Tersedia melalui ScienceDirect <<https://www.sciencedirect.com/>> [Diakses 30 Agustus 2019].
- Alavi Maryam: 1984. *An Assessment Of The Prototyping Approach To Information Systems Development*. *Communications of the ACM*, (Online), Volume XXVII, Nomor 6.
- Bangor A., Kortum P., & Miller J., 2009. *Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*. [e-journal]. Tersedia melalui Semantic Scholar <<https://www.semanticscholar.org/>> [Diakses 16 September 2019].
- Booch G., Ivar R., 2005. *The Unified Modelling Language User Guide*. United State: Addison Wesley Professional.
- Bhuvanewari, N.S., & Sujatha, S., 2011. *Integrating SOA and Web Services*. Denmark : River Publishers.
- Brooke J., 1986. *SUS - A quick and dirty usability scale*. [online]. Tersedia melalui <https://cui.unige.ch/isi/icle-wiki/_media/ipm:test-suschart.pdf/> [Diakses 16 September 2019].
- CASEMaker, 2000. *What is Rapid Application Development (RAD)?* Tersedia melalui <http://www.iro.umontreal.ca/~dift6803/Transparents/Chapitre1/Documents/rad_wp.pdf/> [Diakses 30 Agustus 2019].
- Christanto A. T. S. & Kurniawati R., 2015. *Penerapan Service Oriented Architecture Menggunakan Web Service Pada Aplikasi Perpustakaan Berbasis Android*. *Jurnal Buana Informatika*, VII(1), pp. 75-82.
- Cooksey B., 2014. *An Introduction to APIs*. Tersedia melalui <<https://zapier.com/learn/apis/>> [Diakses 30 Agustus 2019].
- Dharwiyanti S., & Wahono R.S., 2013. *Pengantar Unified Modeling Language (UML)*. Tersedia melalui <<http://rosni-gj.staff.gunadarma.ac.id/>> [Diakses 30 Agustus 2019].
- Doglio F., 2015. *Pro REST API Development with Node.js*. Uruguay: Apress.
- Rohman F. R., Soebroto A. A., & Kharisma A. P., 2018. *Pengembangan Perangkat Lunak Aplikasi Monitoring Klimatologi Menggunakan Metode Restful Web Service Berbasis Android (Studi Kasus : Stasiun Klimatologi Karangploso Malang)*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, II(6), pp. 2017-2023.



Hendini Ade. 2016. *Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus Distro Zhezha Pontianak)*. Jurnal Khatulistiwa Informastika, (Online), Volume IV. Nomor 2.

Kemer E., & Samli R. 2019. *Performance Comparison of Scalable Rest Application Programming Interfaces in Different Platform*. [e-journal]. Tersedia melalui ScienceDirect < <https://www.sciencedirect.com/> > [Diakses 30 Agustus 2019].

Laravel, 2019. *Laravel - The PHP Framework for Web Artisans*. [Online] Tersedia di: <https://laravel.com/> [Diakses 30 Agustus 2019].

Laudon, K. C. a. J. P., 2014. *Management Information Systems: Managing the Digital Firm. 13th ed.* 13th penyunt. United States of America: Pearson Education.

Miles, R. & Hamilton, K., 2006. *Learning UML 2.0*. First penyunt. California: O'Reilly Media, Inc.

Mumbaikarr S., & Padiya P., 2013. *Web Services Based On SOAP and REST Principles. International Journal of Scientific and Research Publications*, III(5), pp. 2250-3153.

MySQL, 2020. *What is MySQL?*. [Online] Tersedia melalui mysql <<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>> [Diakses 24 Februari 2020].

Nauman J. D., & Jenkins M., 1982. *Prototyping: The New Paradigm for Systems Development. MIS Quarterly*, VI(3), pp. 29-44.

Redmond P., 2016. *Writing APIs with Lumen A Hands-on Guide to Writing API Services With PHP*. [e-book]. Tersedia melalui leanpub < <https://leanpub.com/lumen-apis> > [Diakses 30 Agustus 2019].

Postman, 2020. *What is Postman?*. [Online] Tersedia melalui postman < <https://www.postman.com/> > [Diakses 23 April 2020].

Pressman, R. S., 2010. *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH, SEVENTH EDITION*. Seventh penyunt. New York: McGraw-Hill.

PSIK TP UB., 2019. *Fakultas Teknologi Pertanian - Universitas Brawijaya*. [Online] Tersedia di: <https://tp.ub.ac.id/> [Diakses 30 Agustus 2019].

Ristekdikti., 2019. *Panduan Sistem Informasi Manajemen - Peningkatan Kemahasiswaan (Simkatmawa) Tahun 2019*. [Online] Tersedia di: <https://belmawa.ristekdikti.go.id/uploads/2019/03> [Diakses 04 September 2019].

Sharfina Z., & Santoso H. B., 2016. *An Indonesian Adaptation of the System Usability Scale (SUS)*. International Conference on Advanced Computer Science and Information Systems, ICACSI 2016, pp. 145-148.

LAMPIRAN

Laporan Penelitian Hasil Wawancara

Tanggal : 26 Agustus 2019
Waktu : 18.00-19.00
Narasumber : Mas'ud Effendi, STP., MP.
Jabatan : ketua PSIK Fakultas Teknologi Pertanian

Wiku : Permissi pak Mas'ud maaf mengganggu waktunya, sebelumnya perkenalkan nama saya Wiku saya mahasiswa Fakultas Ilmu Komputer. Tujuan saya menemui bapak selaku ketua PSIK (Perencanaan Sistem Informasi dan Kehumasan) FTP (Fakultas Teknologi Pertanian) karena saya mendapat informasi bahwa di FTP terdapat suatu permasalahan terkait perekaman prestasi dari mahasiswa yang ada di FTP dan membutuhkan sistem yang dapat mengatasi permasalahan tersebut. Apakah memang benar pak? Jika iya, mungkin apakah bapak berkenan untuk menjelaskan lebih detail tentang permasalahan tersebut.

Mas'ud : Iya benar, jadi di FTP sendiri ingin memberikan penghargaan terhadap mahasiswa FTP yang berprestasi dengan memberikan pengakuan terhadap prestasi yang telah didapat oleh mahasiswa FTP agar nantinya bisa digunakan sebagai *curriculum vitae* yang valid dalam bentuk SKPI (Surat Keterangan Pendamping Ijazah). Namun pada pelaksanaannya terdapat beberapa kendala mulai dari prestasi yang belum terekam, kemudian dari pihak mahasiswa FTP yang merasa kesulitan untuk melakukan pendataan prestasi yang telah didapat, hingga pihak akademik FTP yang merasa kesulitan untuk melakukan rekap data prestasi yang telah terekam. Dari permasalahan ini, pihak FTP merencanakan untuk merancang sebuah sistem informasi yang dapat mengatasi permasalahan ini.

Wiku : baik pak saya mengerti, jika boleh tau untuk harapan dibuatnya sistem informasi perekaman prestasi mahasiswa ini apakah bapak berkenan untuk menjelaskan juga pak?

Mas'ud : Harapannya dengan adanya sistem informasi ini prestasi – prestasi yang telah didapatkan mahasiswa FTP ini bisa secara langsung terekam. Dan juga pihak akademik mempunyai data dari prestasi yang ada dalam bentuk *excel*.

Wiku : Kemudian untuk alur dari sistemnya kira – kira apa bapak bersedia untuk menjelaskannya juga?

Mas'ud : yang pertama mahasiswa memasukkan data prestasinya ke dalam sistemnya, setelah itu tugas dari kemahasiswaan adalah melakukan verifikasi prestasi yang telah dimasukkan oleh mahasiswa, setelah diverifikasi baru bagian akademik bisa melihat data prestasi mahasiswa dan akademik dapat mengunduh rekapan prestasi mahasiswa dalam format *excel* untuk keperluan di dimasukkan di siacad ub. Kemudian mahasiswa juga bisa mengunduh *draft* prestasi yang berisi



Mas'ud : O begitu, berarti apa bisa juga mas dari pihak lain ketika ingin melihat data prestasi mahasiswa tinggal mengakses dari API yang mas maksud? Semisal kasus dari suatu perusahaan ingin melakukan *cross check* dan melihat data riwayat prestasi mahasiswa ketika ada alumni dari FTP yang melamar pekerjaan disana.

Wiku : Bisa pak.

Wiku : Jadi dari yang bisa saya simpulkan untuk aktor yang akan menggunakan sistem ini ada tiga ya pak, yaitu mahasiswa, kemahasiswaan, dan akademik. Kemudian untuk membedakan ketiga aktor ini pada sistem jika dibuatkan fungsi *login* berdasarkan jabatan dari masing – masing aktor bagaimana pak, apakah tidak keberatan?

Mas'ud : Iya benar jadi nanti ketiga orang tersebut dapat mengakses sistem ini. Iya mas mungkin bisa ditambahkan fungsi *login* dan ketika mahasiswa *login* akan diarahkan kepada halaman mahasiswa, begitu juga dengan kemahasiswaan dan akademik dan juga nanti dibedakan antara *login* sebagai mahasiswa dan juga karyawan (kemahasiswaan dan akademik FTP).

Wiku : Baik pak, saya akan mencoba untuk membuat prototype dari sistem sesuai dari kebutuhan yang bapak uraikan tadi. Saya akan menghubungi bapak kembali setelah prototype selesai saya buat.

Peneliti

Wiku Galindra Wardhana
165150701111020

Ketua PSIK FTP UB

Mas'ud Effendi, STP, MP

Lampiran 1. Dokumentasi wawancara analisis kebutuhan (lanjutan)

Laporan Penelitian Hasil Wawancara

Tanggal : 26 September 2019
Waktu : 13.45-14.15
Narasumber : Mas'ud Effendi, STP., MP.
Jabatan : ketua PSIK Fakultas Teknologi Pertanian

Wiku : Permissi pak Mas'ud maaf mengganggu waktunya, saya telah menyelesaikan *prototype* dari sistem perekaman prestasi mahasiswa yang sempat saya diskusikan dengan bapak kemarin dan tujuan saya menemui bapak adalah untuk memperlihatkan hasilnya kepada bapak.

Mas'ud : Iya silahkan.

Wiku : jadi sistem dimulai dengan *login*. Untuk aktor kemahasiswaan dan akademik nantinya menggunakan *data dummy*, untuk mahasiswa nanti akan menggunakan teknologi *login* dengan menggunakan NIM dan *password* SIAM dari mahasiswa. Terdapat dua halaman *login* yaitu halaman *login* mahasiswa dan *login* karyawan, ketika berada pada halaman *login* mahasiswa terdapat tombol *login* karyawan yang mengarah pada halaman *login* karyawan dan juga pada halaman *login* karyawan terdapat tombol *login* mahasiswa yang mengarah pada halaman *login* mahasiswa. Pada aktor nantinya terdapat jabatan dan ketika *login* dengan jabatan mahasiswa akan diarahkan pada halaman mahasiswa, begitu juga dengan kemahasiswaan dan akademik.

Mas'ud : baik

Wiku : Kemudian untuk halaman mahasiswa sendiri terdiri dari beberapa bagian yaitu terdapat tombol tambah prestasi dan unduh *draft*, yang mana unduh *draft* disini akan mengunduh daftar prestasi yang telah dimasukkan mahasiswa dan telah diverifikasi oleh kemahasiswaan, kemudian juga terdapat tabel yang berisi prestasi yang telah dimasukkan mahasiswa yang telah masuk ke dalam sistem dengan status prestasi verifikasi dan menunggu, dan jika status menunggu maka terdapat tombol hapus, sehingga mahasiswa dapat menghapus prestasi jika terdapat kesalahan ketika memasukkan prestasi.

Mas'ud : Baik saya mengerti.

Wiku : Kemudian, ketika tombol tambah prestasi ditekan maka akan muncul form masukan data prestasi yang terdiri dari nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kejuaraan, posisi/jabatan, uraian kegiatan/prestasi, serta *upload file* gambar. Dan untuk bagian tingkat kejuaraan dan posisi/jabatan terdapat sebuah pilihan. Pada tingkat kejuaraan terdapat pilihan internasional, nasional, regional, universitas, fakultas, jurusan/program studi, komunitas studi, dan juga strip (-). Pada posisi/jabatan terdapat pilihan juara 1, juara 2, juara 3 jika prestasi berupa

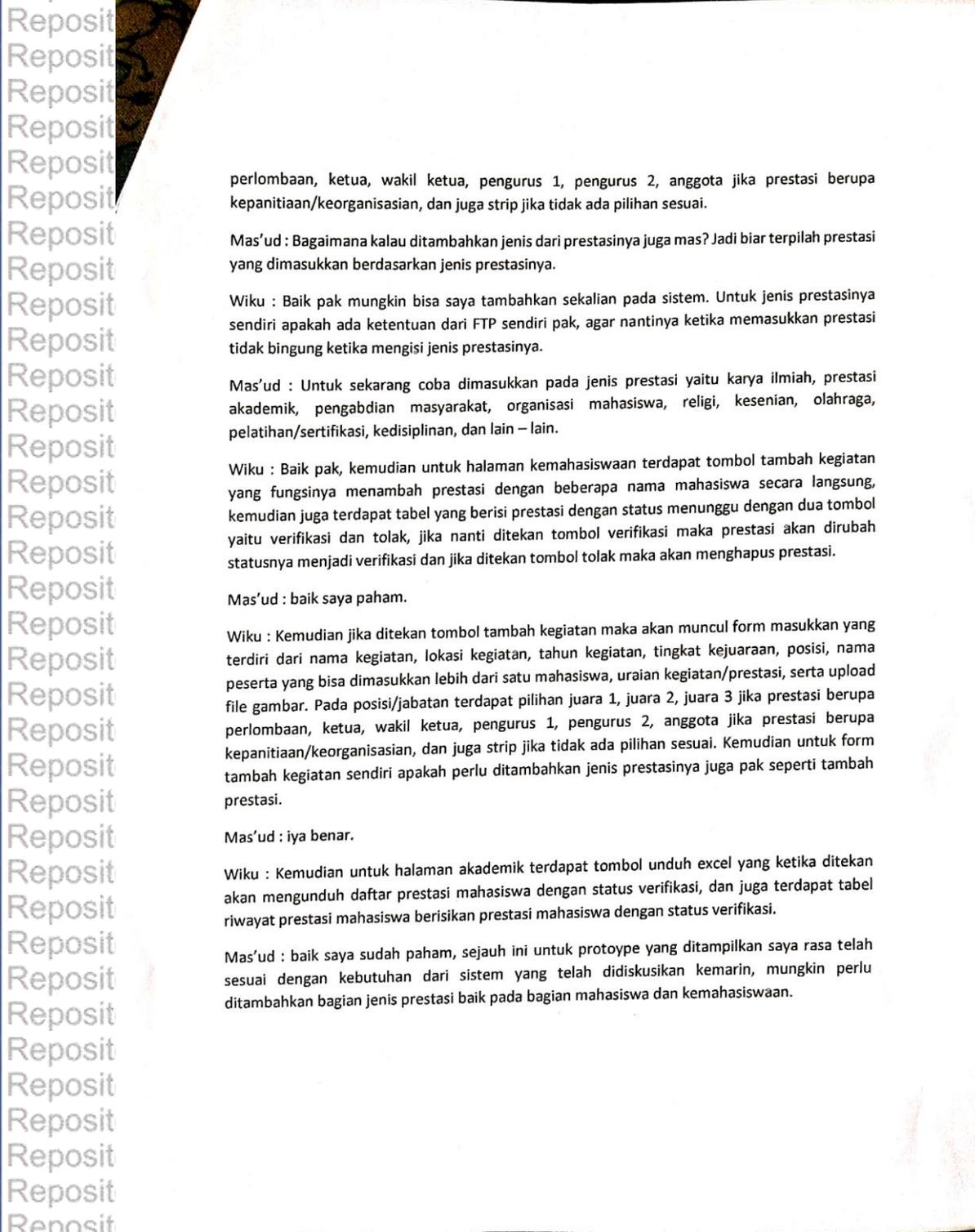
Lampiran 2 Dokumentasi wawancara presentasi *prototype* pertama



Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya



perlombaan, ketua, wakil ketua, pengurus 1, pengurus 2, anggota jika prestasi berupa kepanitiaan/keorganisasian, dan juga strip jika tidak ada pilihan sesuai.

Mas'ud : Bagaimana kalau ditambahkan jenis dari prestasinya juga mas? Jadi biar terpilah prestasi yang dimasukkan berdasarkan jenis prestasinya.

Wiku : Baik pak mungkin bisa saya tambahkan sekalian pada sistem. Untuk jenis prestasinya sendiri apakah ada ketentuan dari FTP sendiri pak, agar nantinya ketika memasukkan prestasi tidak bingung ketika mengisi jenis prestasinya.

Mas'ud : Untuk sekarang coba dimasukkan pada jenis prestasi yaitu karya ilmiah, prestasi akademik, pengabdian masyarakat, organisasi mahasiswa, religi, kesenian, olahraga, pelatihan/sertifikasi, kedisiplinan, dan lain – lain.

Wiku : Baik pak, kemudian untuk halaman kemahasiswaan terdapat tombol tambah kegiatan yang fungsinya menambah prestasi dengan beberapa nama mahasiswa secara langsung, kemudian juga terdapat tabel yang berisi prestasi dengan status menunggu dengan dua tombol yaitu verifikasi dan tolak, jika nanti ditekan tombol verifikasi maka prestasi akan dirubah statusnya menjadi verifikasi dan jika ditekan tombol tolak maka akan menghapus prestasi.

Mas'ud : baik saya paham.

Wiku : Kemudian jika ditekan tombol tambah kegiatan maka akan muncul form masukkan yang terdiri dari nama kegiatan, lokasi kegiatan, tahun kegiatan, tingkat kejuaraan, posisi, nama peserta yang bisa dimasukkan lebih dari satu mahasiswa, uraian kegiatan/prestasi, serta upload file gambar. Pada posisi/jabatan terdapat pilihan juara 1, juara 2, juara 3 jika prestasi berupa perlombaan, ketua, wakil ketua, pengurus 1, pengurus 2, anggota jika prestasi berupa kepanitiaan/keorganisasian, dan juga strip jika tidak ada pilihan sesuai. Kemudian untuk form tambah kegiatan sendiri apakah perlu ditambahkan jenis prestasinya juga pak seperti tambah prestasi.

Mas'ud : iya benar.

Wiku : Kemudian untuk halaman akademik terdapat tombol unduh excel yang ketika ditekan akan mengunduh daftar prestasi mahasiswa dengan status verifikasi, dan juga terdapat tabel riwayat prestasi mahasiswa berisikan prestasi mahasiswa dengan status verifikasi.

Mas'ud : baik saya sudah paham, sejauh ini untuk prototype yang ditampilkan saya rasa telah sesuai dengan kebutuhan dari sistem yang telah didiskusikan kemarin, mungkin perlu ditambahkan bagian jenis prestasi baik pada bagian mahasiswa dan kemahasiswaan.

Lampiran 2 Dokumentasi wawancara presentasi *prototype* pertama (lanjutan)

Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya



LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : JESSICA YOLANDA (195100601111007)
 Jabatan : MAHASISWA FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (✓) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu-ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

1. Saya berpikir akan menggunakan sistem ini lagi.

				✓
--	--	--	--	---
2. Saya merasa sistem ini rumit untuk digunakan.

	✓			
--	---	--	--	--
3. Saya merasa sistem ini mudah digunakan.

			✓	
--	--	--	---	--
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

	✓			
--	---	--	--	--
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

			✓	
--	--	--	---	--
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).

✓				
---	--	--	--	--
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

			✓	
--	--	--	---	--
8. Saya merasa sistem ini membingungkan.

✓				
---	--	--	--	--
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

				✓
--	--	--	--	---
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

			✓	
--	--	--	---	--

Malang, 4 FEBRUARI 2020
 Responden

JESSICA YOLANDA



LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : ADELIA ARUM NURMAWATI (19510010111023)
 Jabatan : MAHASISWA FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (v) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu - ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

1. Saya berpikir akan menggunakan sistem ini lagi.

			✓	
1	2	3	4	5
2. Saya merasa sistem ini rumit untuk digunakan.

v				
1	2	3	4	5
3. Saya merasa sistem ini mudah digunakan.

				v
1	2	3	4	5
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

		v		
1	2	3	4	5
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

			v	
1	2	3	4	5
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).

	v			
1	2	3	4	5
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

				v
1	2	3	4	5
8. Saya merasa sistem ini membingungkan.

v				
1	2	3	4	5
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

			v	
1	2	3	4	5
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

			v	
1	2	3	4	5

Malang, 04 Feb 2020
 Responden

Adelia Arum
ADELIA ARUM

Lampiran 6 Kuisiонер SUS responden 3

LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : Siti Masrurah (17510010111022)
Jabatan : Mahasiswa FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (v) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu - ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

- 1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Malang, 04 Feb 2020
Responden

[Signature]
Siti Masrurah

Lampiran 7 Kuisisioner SUS responden 4

LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : KIRARA JAUZA MILLENIA (185100107111018)
Jabatan : MAHASISWA FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (✓) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu - ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

- Saya berpikir akan menggunakan sistem ini lagi.

				✓
1	2	3	4	5
- Saya merasa sistem ini rumit untuk digunakan.

	✓			
1	2	3	4	5
- Saya merasa sistem ini mudah digunakan.

			✓	
1	2	3	4	5
- Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

	✓			
1	2	3	4	5
- Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

			✓	
1	2	3	4	5
- Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).

✓				
1	2	3	4	5
- Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

				✓
1	2	3	4	5
- Saya merasa sistem ini membingungkan.

	✓			
1	2	3	4	5
- Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

			✓	
1	2	3	4	5
- Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

✓				
1	2	3	4	5

Malang, 5 Februari 2020

Responden

KIRARA JAUZA M.



LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : Jihan Nada Sulabilla Erha (185100601111008)
Jabatan : Mahasiswa FTTP.

Jawablah pertanyaan berikut dengan memberi satu tanda centang (v) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu-ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

- 1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Malang, 5 Februari 2020
Responden

Handwritten signature: Nade
Jihan Nada Sulabilla Erha

Lampiran 9 Kuisiomer SUS responden 6



LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : YEFRI M.
 Jabatan : STAF PEMAHASISWAAN FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (✓) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu - ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

1. Saya berpikir akan menggunakan sistem ini lagi.

1	2	3	4	5
---	---	---	---	---
2. Saya merasa sistem ini rumit untuk digunakan.

1	2	3	4	5
---	---	---	---	---
3. Saya merasa sistem ini mudah digunakan.

1	2	3	4	5
---	---	---	---	---
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

1	2	3	4	5
---	---	---	---	---
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

1	2	3	4	5
---	---	---	---	---
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).

1	2	3	4	5
---	---	---	---	---
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

1	2	3	4	5
---	---	---	---	---
8. Saya merasa sistem ini membingungkan.

1	2	3	4	5
---	---	---	---	---
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

1	2	3	4	5
---	---	---	---	---
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

1	2	3	4	5
---	---	---	---	---

Malang, 27 JANUARI 2020
 Responden

Lampiran 10 Kuisisioner SUS responden 7



LEMBAR KUESIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI PEREKAMAN PRESTASI MAHASISWA

(Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)

Nama : Dani P.
Jabatan : Staf Akademik FTP

Jawablah pertanyaan berikut dengan memberi satu tanda centang (v) pada setiap pertanyaan pada kolom jawaban yang tersedia. Dengan nilai 1 = sangat tidak setuju, nilai 2 = tidak setuju, nilai 3 = ragu - ragu, nilai 4 = setuju, dan nilai 5 = sangat setuju.

- 1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Malang, 29 Januari 2020
Responden

[Signature]
Dani P.

Lampiran 11 Kuisisioner SUS responden 8