

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Penelitian ini dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu metode pengembangan sistem *Prototype*. Adapun tahapan *Prototype* dapat dilihat pada gambar sebagai berikut :

3.1.1 Comunnication

Tahapan komunikasi dalam penelitian ini meliputi observasi, wawancara, dan dokumentasi. Adapun penjelasan mengenai setiap tahapannya adalah seperti berikut :

1. Observasi

Metode pengumpulan data terkait penelitian ini dilakukan secara observasi (pengamatan secara langsung) dengan mengunjungi langsung tempat yang terkait dengan penelitian yaitu pada IIB Darmajaya yang berlokasi di Jl. ZA. Pagar Alam No.93, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung 35141.

2. Wawancara

Wawancara merupakan salah satu teknik yang dapat digunakan untuk mengumpulkan data penelitian. Secara sederhana dapat dikatakan bahwa wawancara adalah suatu kejadian atau suatu proses interaksi antara pewawancara dan sumber informasi atau orang yang di wawancarai melalui komunikasi langsung. Dalam metode wawancara ini penulis selaku pewawancara/orang yang sedang melakukan penelitian langsung berinteraksi dengan dosen dan mahasiswa IIB Darmajaya.

3. Dokumentasi

Metode dokumentasi dalam penelitian merupakan pelengkap dari penggunaan metode observasi dan wawancara. Studi dokumentasi yaitu mengumpulkan dokumen dan data-data yang diperlukan dalam permasalahan penelitian. Dalam hal ini saya selaku peneliti mengunjungi langsung tempat penelitian yaitu IIB Darmajaya untuk mendapatkan data-

data pendukung berupa gambar-gambar atau dokumen lainnya sebagai penjelasan untuk data-data yang diperoleh dari hasil penelitian dan pengamatan secara langsung.

3.1.2 Quick Plan

Perencanaan secara cepat adalah tahapan dengan melakukan analisis dan perencanaan yang tepat setelah data didapatkan pada tahap komunikasi dengan dosen dan mahasiswa terkait masalah yang didapat.

3.1.2.1 Analisis Kebutuhan Perangkat Lunak

Untuk mempermudah berjalanya kegiatan praktikum dengan lebih cepat, dan menjamin penyimpanan data yang lebih mudah maka Laboratorium IIB Darmajaya khususnya prodi Teknik Informatika membutuhkan suatu sistem yang dapat menangani pencatatan lembar kegiatan praktikum, program ini banyak dipakai oleh peneliti sebelumnya. Untuk mewujudkannya maka peneliti merancang suatu rancangan sistem pengisian lembar kegiatan praktikum yang dapat membantu dosen dan mahasiswa dalam melaksanakan kegiatan praktikum sehingga pencatatan laporan dapat dilakukan dengan lebih terstruktur.

1. Analisis Kebutuhan Fungsional

Kebutuhan fungsional yaitu kebutuhan yang diperlukan oleh dosen dan mahasiswa saat mengakses aplikasi sistem pengisian lembar kegiatan praktikum yang terdiri dari :

- 1). Kebutuhan dosen :
 - a. Login
 - b. Menu utama
 - c. Mengelola data kelas
 - d. Mengelola data pertemuan
 - e. Mengelola data instruksi laporan
 - f. Mengelola data nilai mahasiswa
 - g. Monitoring data laporan mahasiswa

2). Kebutuhan mahasiswa :

- a. Daftar login
- b. Menu utama
- c. Melakukan pendaftaran pada kelas
- d. Melakukan download instruksi praktikum
- e. Melakukan proses pengerjaan praktikum melalui website.
- f. Monitoring laporan praktikum
- g. Monitoring data nilai

2. Analisis Kebutuhan Non-Fungsional

Selain kebutuhan fungsional, diperlukan juga kebutuhan non-fungsional sebagai berikut:

- 1) Performa: Sistem harus mampu mengelola data dan laporan dengan cepat dan efisien.
- 2) Usability: Sistem harus mudah digunakan oleh pengguna, sehingga tidak diperlukan pelatihan khusus untuk menggunakannya.
- 3) Security: Sistem harus terlindungi dari akses yang tidak sah dan perlindungan terhadap data pengguna.
- 4) Reliability: Sistem harus stabil dan dapat diandalkan dalam mengelola data dan laporan.

3.1.2.2 Analisis Kebutuhan Teknis

Kebutuhan teknis sistem yang dibangun adalah sebagai berikut:

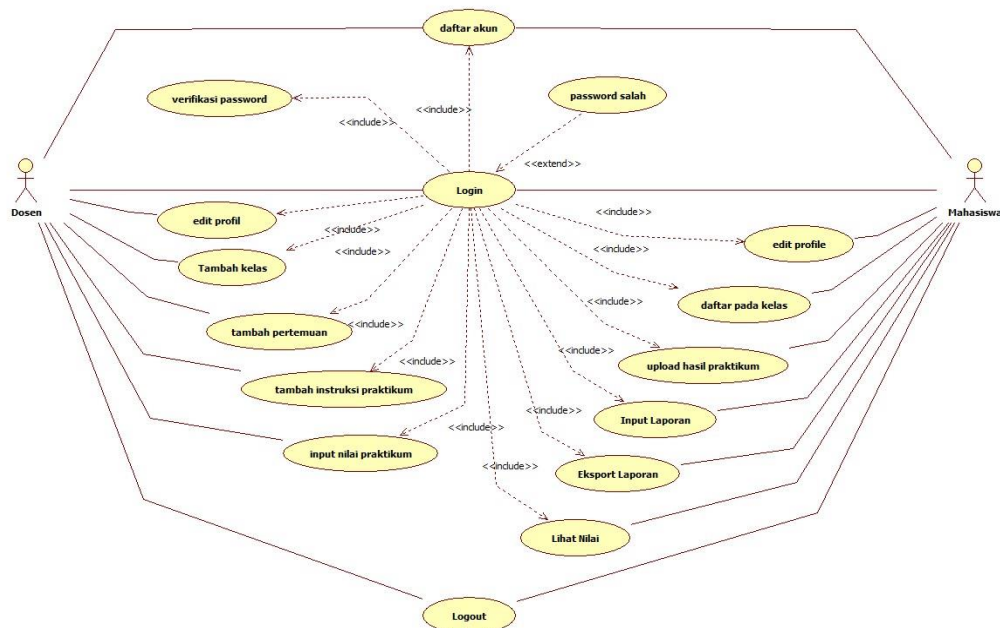
1. *Framework CodeIgniter*: *Framework* PHP yang akan digunakan sebagai dasar pembangunan sistem.
2. Basis Data: Sistem akan menggunakan MySQL sebagai basis data untuk mengelola data dan laporan kegiatan praktikum.
3. Bahasa Pemrograman: Sistem akan dibangun dengan bahasa pemrograman PHP.
4. *Web Server*: Sistem akan dijalankan pada *web server Apache*.

3.1.3 Modeling Quick Design

Tahap ini merupakan tahapan untuk perancangan alur program Aplikasi Pengisian kegiatan praktikum Berbasis mobile. Perancangan ini terdiri dari dua aktor, yaitu aktor mahasiswa dan aktor dosen, spesifikasi Perancangan Sistem dilakukan dengan UML (*Unified Modelling Language*) yang menerapkan *Use Case Diagram* dan *Activity Diagram*.

a. Use Case Diagram

Pada *Use Case Diagram* ini menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang akan menjelaskan kerja sistem secara garis besar dengan mempresentasikan interaksi aktor dengan sistem yang dibuat serta memberikan gambaran fungsi dari sistem. Dalam penelitian ini, aktor yang terlibat dalam aplikasi Silaprak adalah Mahasiswa sebagai pihak pengguna layanan tersebut yang dapat dilihat pada gambar 3.1



Gambar 3. 1 Use Case Diagram Aplikasi

Gambar 3.1 merupakan *Use Case Diagram* Sistem, skenario pendeskripsian dan pendefinisiannya dapat dilihat pada Tabel 3.1 dan Tabel 3.2

Tabel 3. 1 Deskripsi Aktor

No	Aktor	Deskripsi
1	Dosen	Dosen Pengajar yang akan memberikan materi dan penilaian praktikum
2	Mahasiswa	Mahasiswa yang akan melaksanakan kegiatan praktikum

Tabel 3. 2 Skenario Use Case Diagram Aplikasi

Aktor	Aksi	Aksi Aktor	Sistem
Dosen	Login	1. Dosen mengisi form login	1. Validasi username dan password 2. Masuk ke halaman dashboard dosen
	Edit profil	1. Memilih menu “my profile” 2. Mengisi form update profil	1. Menampilkan data profil dosen 2. Melakukan update data dari <i>database</i>
	Tambah data kelas	1. Memilih menu tambah kelas	1. Melakukan insert data kelas baru kedalam <i>database</i>
	Tambah data pertemuan	1. Memilih menu “tambah pertemuan”	1. Melakukan insert data pertemuan baru kedalam <i>database</i>
	Tambah instruksi praktikum	1. Memilih menu “kelola pertemuan”	1. Melakukan insert data instruksi praktikum kedalam <i>database</i>
	Input nilai laporan	1. Memilih menu “input nilai”	1. Melakukan insert data nilai kedalam <i>database</i>
	Monitoring nilai	1. Dosen memilih menu daftar nilai 2. Memilih menu lihat mahasiswa 3. Memilih lihat detail nilai	1. Menampilkan data seluruh kelas yang dimiliki beserta jumlah mahasiswa di dalamnya 2. Menampilkan seluruh daftar mahasiswa yang ada dalam kelas dosen yang dipilih

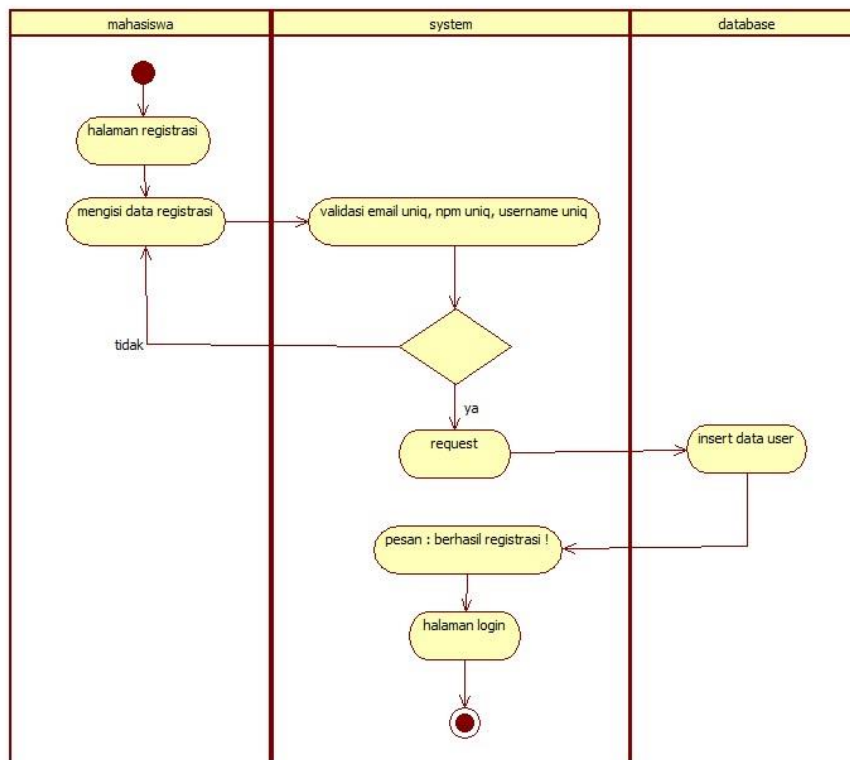
			3. Menampilkan detail nilai dari mahasiswa yang dipilih secara keseluruhan.
Mahasiswa	Login	1. Mahasiswa mengisi form login	1. Melakukan validasi apakah username dan password yang dimasukan benar 2. Masuk ke halaman dashboard mahasiswa
	Edit profil	1. Memilih menu “my profile” 2. Mengisi form update profil	1. Menampilkan data profil dosen 2. Melakukan update data dari <i>database</i>
	Melakukan pencarian kelas	1. Mahasiswa memasukan nama mata kuliah dalam search box	1. Mencari kelas berdasarkan nama mata kuliah yang telah di inputkan oleh mahasiswa
	Registrasi pada kelas	1. Mahasiswa menekan tombol “Daftarkan Saya” 2. Mahasiswa menekan tombol “Yakin”	1. Memunculkan pop up konfirmasi “apakah yakin mendaftar”. 2. Melakukan insert data kelas baru kedalam <i>database</i> .
	Mengerjakan laporan	1. Menekan tombol “Kerjakan Laporan” 2. Menekan tombol “Kumpulkan Laporan”	1. Menampilkan text box untuk mengerjakan laporan 2. Melakukan insert data laporan kedalam <i>database</i>
	Monitoring nilai	1. Menekan tombol “Daftar Nilai” 2. Memilih mata kuliah yang akan dilihat	1. Menampilkan daftar kelas yang diikuti oleh mahasiswa 2. Menampilkan data nilai keseluruhan dari kelas yang dipilih

b. *Activity Diagram*

Activity diagram menggambarkan rangkaian aliran dan aktivitas yang digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi. Pembuatan *activity diagram* pada awal proses dapat membantu memahami keseluruhan proses dari berjalannya suatu sistem.

1. *Activity Diagram* Registrasi

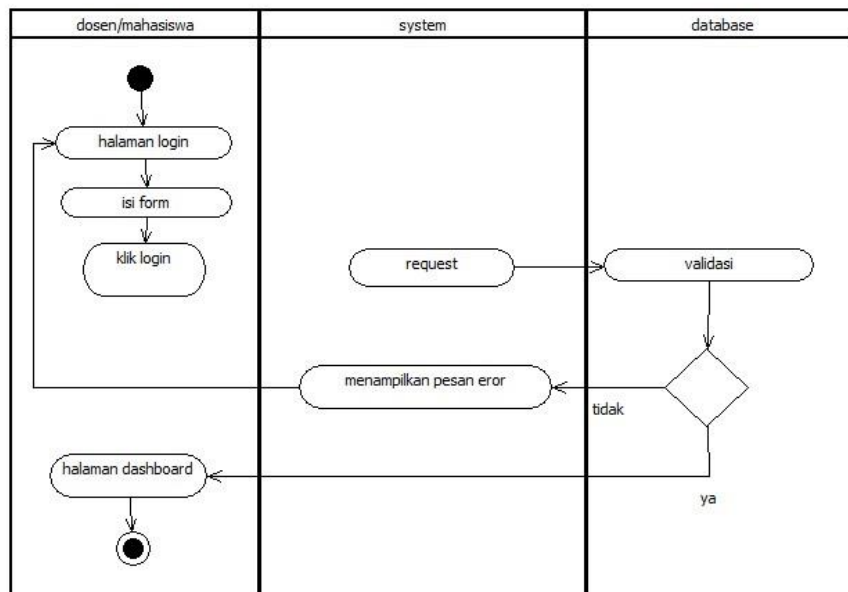
Activity diagram registrasi menjelaskan proses awal untuk mendaftar kedalam sistem menggunakan aplikasi, Proses *activity diagram* registrasi dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Activity Diagram Registrasi Mahasiswa

2. *Activity Diagram* Login Mahasiswa dan Dosen

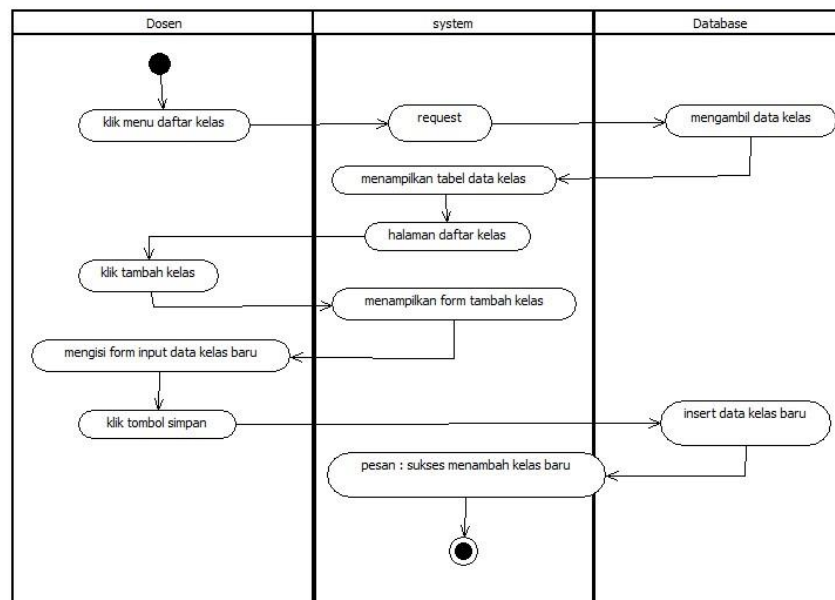
Activity diagram login menjelaskan proses awal untuk masuk kedalam sistem menggunakan aplikasi baik itu sebagai Dosen maupun sebagai Mahasiswa. Proses *activity diagram* login dapat dilihat pada gambar Gambar 3.3.



Gambar 3.3 Activity Diagram Login Mahasiswa

3. Activity diagram tambah kelas untuk Dosen

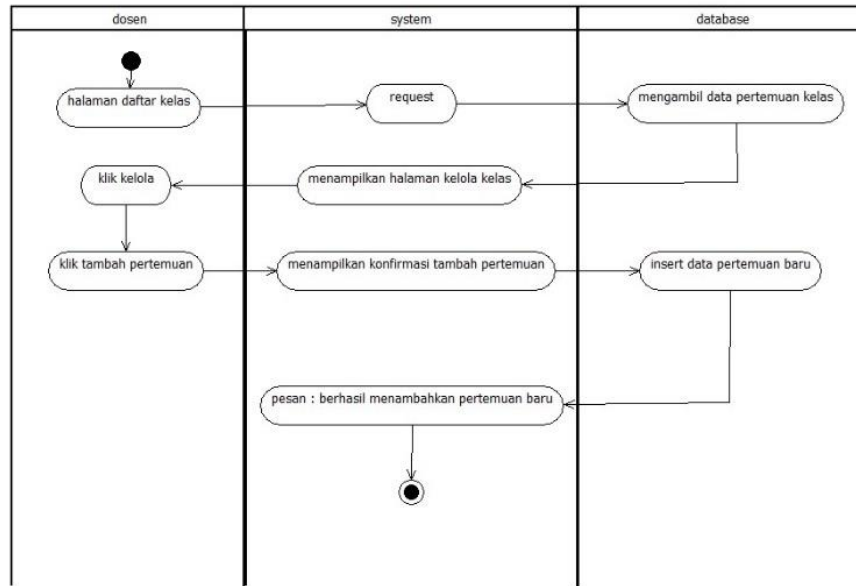
Activity diagram tambah kelas menjelaskan proses dosen dalam menambahkan kelas baru. Proses activity diagram tambah kelas dapat dilihat pada Gambar 3.4.



Gambar 3.4 Activity Diagram Tambah Kelas Dosen

4. Activity diagram tambah pertemuan baru untuk Dosen.

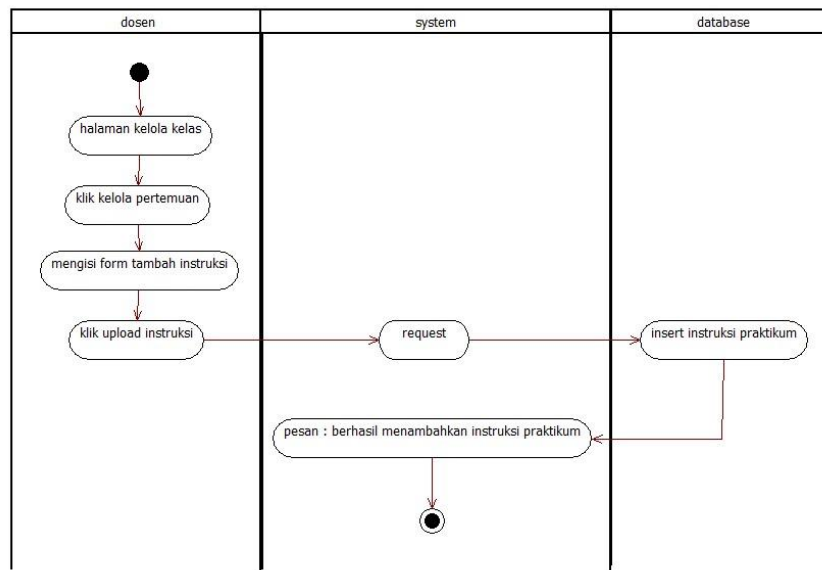
Activity diagram tambah pertemuan menjelaskan proses dosen dalam menyusun jadwal pertemuan dengan mahasiswa. Proses *activity diagram* tambah pertemuan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Tambah Pertemuan Baru

5. *Activity diagram* form upload instruksi praktikum dosen

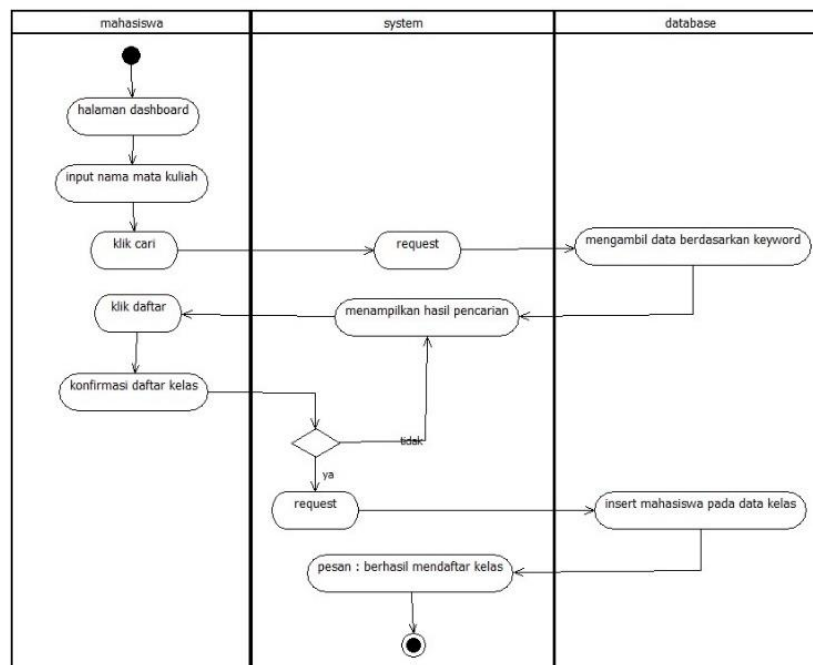
Activity diagram form upload instruksi menjelaskan proses dosen mengupload instruksi praktikum. Proses *activity diagram* form upload instruksi praktikum dapat dilihat pada Gambar 3.6.



Gambar 3.6 Activity Diagram Upload Instruksi Praktikum

6. *Activity diagram* pencarian kelas untuk Mahasiswa

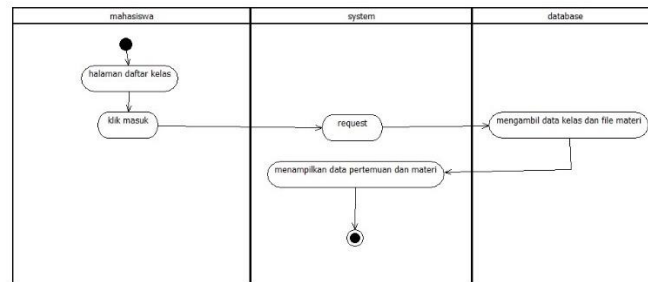
Activity diagram pencarian kelas menjelaskan bagaimana mahasiswa dalam mencari kelas praktikum. Proses *Activity diagram* pencarian kelas dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity Diagram Pencarian Kelas Mahasiswa

7. Activity diagram masuk kelas Mahasiswa

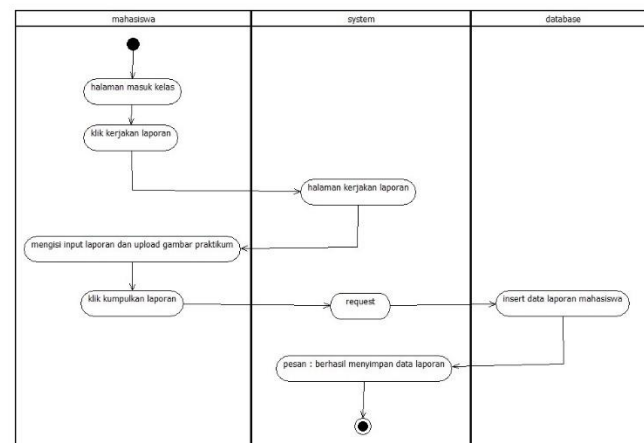
Activity diagram masuk kelas menjelaskan bagaimana mahasiswa masuk kelas dan dapat melihat materi pertemuan yang diberikan oleh dosen. Proses activity diagram masuk kelas dapat dilihat pada gambar Gambar 3.8



Gambar 3.8 Activity Diagram Masuk Kelas Mahasiswa

8. Activity diagram halaman pengerjaan laporan

Activity diagram halaman pengerjaan laporan menjelaskan bagaimana mahasiswa mengerjakan laporan praktikum secara langsung pada halaman pengerjaan. Proses activity diagram halaman pengerjaan laporan dapat dilihat pada Gambar 3.9



Gambar 3.9 Activity Diagram Halaman Pengerjaan Laporan

3.1.4 Construction of Prototype

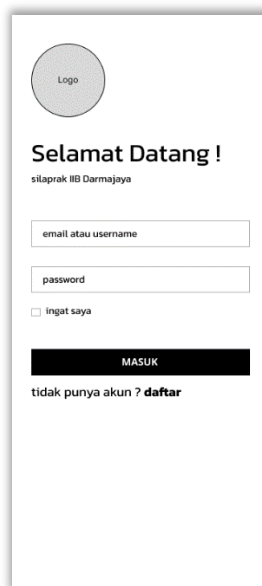
Tahap ini adalah pembentukan *prototype*, yaitu pembuatan *prototype* termasuk dalam pengujian dan penyempurnaan. Rancangan *Interface*

merupakan desain awal sebelum membangun suatu perangkat lunak, hasil dari perangkat lunak yang dibangun nantinya tidak jauh berbeda dengan perancangan *interface* yang dibuat.

Berikut adalah rancangan *interface* pada aplikasi SILAPRAK yang akan dibangun:

3.1.4.1 Tampilan Rancangan *Interface* Halaman Login Dosen

Gambar 3.10 berikut merupakan rancangan halaman login yang akan digunakan untuk memasuki aplikasi :



The image shows a vertical rectangular login form. At the top, there is a circular placeholder labeled 'Logo'. Below it, the text 'Selamat Datang !' is displayed in a bold font, followed by 'silaprak IIB Darmajaya' in a smaller font. The form contains two input fields: 'email atau username' and 'password'. Below these fields is a checkbox labeled 'ingat saya'. A black button with the white text 'MASUK' is positioned below the checkbox. At the bottom of the form, there is a link that reads 'tidak punya akun ? **daftar**'.

Gambar 3. 10 Interface Halaman Login Dosen

3.1.4.2 Tampilan Rancangan *Interface* Halaman Dashboard Dosen

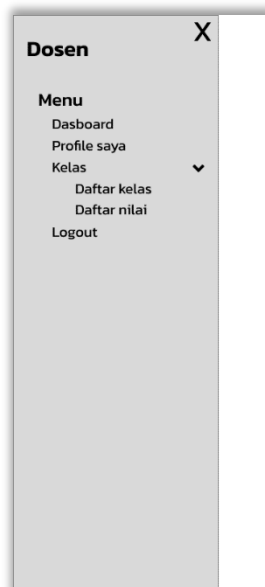
Gambar 3.11 rancangan halaman dashboard atau halaman yang akan muncul pertama kali ketika dosen berhasil melakukan login dengan benar.



Gambar 3.11 Interface Halaman Dashboard Dosen

3.1.4.3 Tampilan Rancangan *Interface* Halaman NavBar Dosen

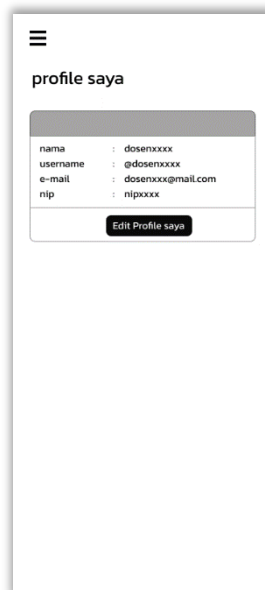
Gambar 3.12 merupakan rancangan untuk tampilan navbar (*navigation bar*) yang akan digunakan dosen untuk berpindah pindah halaman.



Gambar 3.12 Interface Halaman NavBar Dosen

3.1.4.4 Tampilan Rancangan *Interface* Halaman Profil Dosen

Gambar 3.13 merupakan rancangan halaman profile dosen yang akan digunakan untuk melihat data profil dosen saat ini.



☰

profile saya

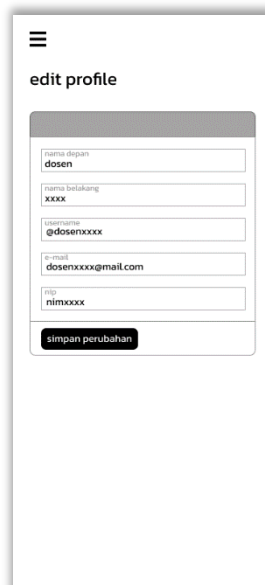
nama	: dosenxxxx
username	: @dosenxxxx
e-mail	: dosenxxxx@mail.com
nip	: nipxxxx

Edit Profile saya

Gambar 3.13 Interface Halaman Profil Dosen

3.1.4.5 Tampilan Rancangan *Interface* Halaman Edit Profil Dosen

Gambar 3.14 merupakan rancangan halaman untuk edit profile dosen yang akan digunakan untuk melakukan update data profil (edit).



☰

edit profile

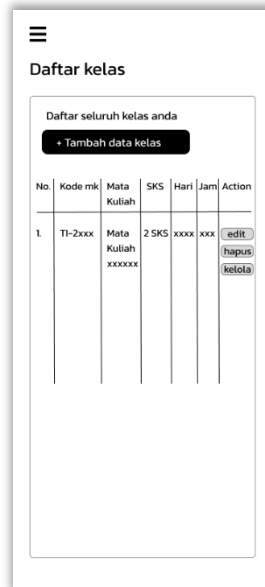
nama depan	dosen
nama belakang	xxxx
username	@dosenxxxx
e-mail	dosenxxxx@mail.com
nip	nipxxxx

simpan perubahan

Gambar 3.14 Interface Halaman Edit Profile Dosen

3.1.4.6 Tampilan Rancangan *Interface* Halaman Daftar Kelas Dosen

Gambar 3.15 merupakan rancangan untuk tampilan halaman daftar kelas, pada halaman ini dosen akan melihat daftar kelas yang dimiliki.



Gambar 3.15 Interface Halaman Daftar Kelas Dosen

3.1.4.7 Tampilan Rancangan *Interface* Halaman Tambah Kelas Dosen

Gambar 3.16 merupakan rancangan tampilan halaman tambah kelas pada daftar kelas dosen, yang akan digunakan oleh dosen untuk menambahkan kelas baru.

Gambar 3.16 Interface Halaman Tambah Kelas Dosen

3.1.4.8 Tampilan Rancangan *Interface* Halaman Kelola Kelas Dosen

Gambar 3.17 merupakan rancangan tampilan untuk halaman kelola kelas, pada halaman ini dosen dapat menambahkan pertemuan baru.

No.	Nama	Aksi
1.	pertemuan 1	kelola pertemuan
2.	pertemuan 2	kelola pertemuan
3.	pertemuan 3	kelola pertemuan
4.	pertemuan 4	kelola pertemuan
5.	pertemuan 5	kelola pertemuan

Gambar 3.17 Interface Halaman Kelola Kelas Dosen

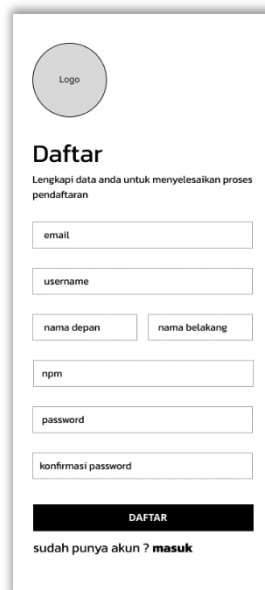
3.1.4.9 Tampilan Rancangan *Interface* Halaman Kelola Pertemuan Dosen

Gambar 3.18 merupakan rancangan tampilan untuk halaman kelola pertemuan dosen, pada halaman ini dosen dapat mengelola pertemuan yang sudah dibuat untuk dapat melakukan upload file instruksi praktikum dalam format pdf/doc.

Gambar 3.18 Interface Halaman Kelola Pertemuan Dosen

3.1.4.10 Tampilan Rancangan *Interface* Halaman Registrasi Mahasiswa

Gambar 3.19 merupakan rancangan tampilan untuk halaman registrasi mahasiswa, pada halaman ini mahasiswa dapat melakukan pendaftaran untuk digunakan pada halaman login.



The image shows a mobile application registration screen. At the top, there is a circular logo placeholder labeled 'Logo'. Below it, the title 'Daftar' is displayed in a bold font. Under the title, a subtitle reads 'Lengkapi data anda untuk menyelesaikan proses pendaftaran'. The form consists of several input fields: 'email', 'username', 'nama depan' (first name), 'nama belakang' (last name), 'npm', 'password', and 'konfirmasi password'. At the bottom of the form is a black button with the text 'DAFTAR' in white. Below the button, there is a link that says 'sudah punya akun ? masuk'.

Logo

Daftar

Lengkapi data anda untuk menyelesaikan proses pendaftaran

email

username

nama depan nama belakang

npm

password

konfirmasi password

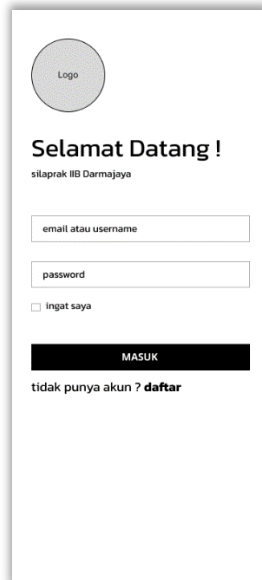
DAFTAR

sudah punya akun ? **masuk**

Gambar 3.19 Interface Halaman Registrasi Mahasiswa

3.1.4.11 Tampilan Rancangan *Interface* Halaman Login Mahasiswa

Gambar 3.20 merupakan rancangan tampilan halaman login mahasiswa, yang akan digunakan oleh mahasiswa untuk melakukan login pada aplikasi SILAPRAK.

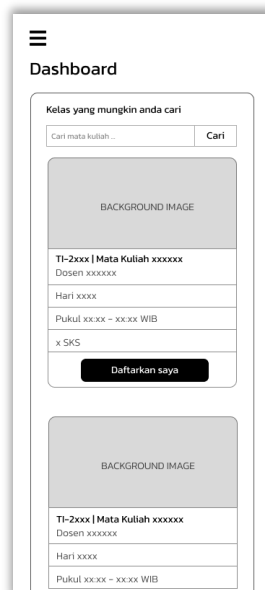


The image shows a vertical rectangular login form. At the top, there is a circular placeholder labeled 'Logo'. Below this, the text 'Selamat Datang !' is displayed in a bold font, followed by 'silaprak IB Darmajaya' in a smaller font. The form contains two input fields: the first is labeled 'email atau username' and the second is labeled 'password'. Below these fields is a checkbox labeled 'ingat saya'. A black button with the white text 'MASUK' is positioned below the checkbox. At the bottom of the form, the text 'tidak punya akun ? **daftar**' is shown, where 'daftar' is in bold.

Gambar 3.20 Interface Halaman Login Mahasiswa

3.1.4.12 Tampilan Rancangan *Interface* Halaman Dashboard Mahasiswa

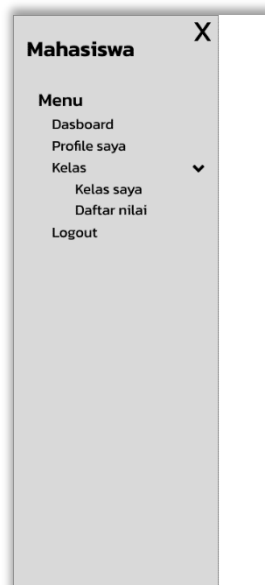
Gambar 3.21 merupakan rancangan tampilan halaman dashboard, halaman ini adalah halaman yang pertama kali tampil ketika mahasiswa berhasil melakukan login dengan benar, pada halaman ini mahasiswa dapat melakukan pencarian kelas yang diinginkan dengan memasukkan nama mata kuliah yang dicari.



Gambar 3.21 Interface Halaman Dashboard Mahasiswa

3.1.4.13 Tampilan Rancangan Interface Halaman NavBar Mahasiswa

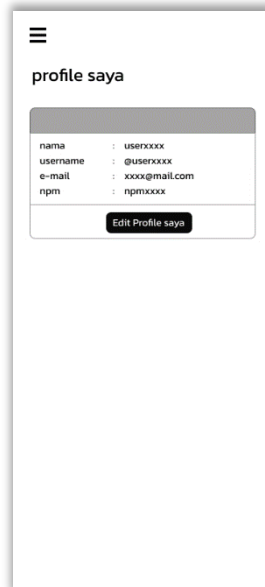
Gambar 3.22 merupakan rancangan tampilan halaman *navigation bar* (navbar), pada bagian tersedia beberapa menu agar mahasiswa dapat melakukan perpindahan ke halaman lain.



Gambar 3.22 Interface Halaman NavBar Mahasiswa

3.1.4.14 Tampilan Rancangan *Interface* Halaman Profil Mahasiswa

Gambar 3.23 merupakan rancangan tampilan halaman profil mahasiswa, pada halaman ini mahasiswa dapat melihat detail data profilnya.



Gambar 3.23 Interface Halaman Profil Mahasiswa

3.1.4.15 Tampilan Rancangan *Interface* Halaman Edit Profil Mahasiswa

Gambar 3.24 merupakan adalah rancangan tampilan halaman profil mahasiswa, pada halaman ini mahasiswa dapat melakukan *update* data profile (*edit*).

The screenshot shows a mobile application interface for editing a profile. At the top, there is a hamburger menu icon and the title "edit profile". Below the title is a form with several input fields: "nama depan" (first name) with the value "userxxxx", "nama belakang" (last name) with the value "xxxx", "username" with the value "@userxxxx", "e-mail" with the value "xxxx@mail.com", and "npm" with the value "npmxxxx". At the bottom of the form is a button labeled "simpan perubahan" (save changes).

Gambar 3.24 Interface Halaman Edit Profil Mahasiswa

3.1.4.16 Tampilan Rancangan *Interface* Halaman Kelas Mahasiswa

Gambar 3.25 merupakan rancangan tampilan halaman kelas mahasiswa, pada halaman ini mahasiswa dapat melihat kelas yang mereka miliki.

The screenshot shows a mobile application interface for a list of classes. At the top, there is a hamburger menu icon and the title "Daftar kelas". Below the title is a section header "kelas anda" (your classes). The list contains two identical class entries. Each entry has a "BACKGROUND IMAGE" placeholder, followed by the course name "TI-2xxx | Mata Kuliah xxxxxxx", the lecturer "Dosen xxxxxx", the day "Hari xxxxx", and the time "Pukul xxxxx - xxxxx WIB". At the bottom of each entry are two buttons: "Masuk" (enter) and "Hapus kelas" (delete class).

Gambar 3.25 Interface Halaman Kelas Mahasiswa

3.1.4.17 Tampilan Rancangan *interface* Halaman Laporan Praktikum Mahasiswa

Gambar 3.26 merupakan rancangan tampilan halaman laporan praktikum mahasiswa, pada halaman ini mahasiswa dapat mengerjakan laporan dan mengumpulkan laporan.

Gambar 3.26 Interface Halaman Laporan Praktikum Mahasiswa

3.1.4.18 Tampilan Rancangan *Interface* Halaman Laporan Lengkap Mahasiswa

Gambar 3.27 merupakan rancangan tampilan halaman laporan lengkap mahasiswa, pada halaman ini mahasiswa dapat melihat seluruh laporan yang telah mereka buat dan melakukan ekspor menjadi pdf/doc.

The screenshot displays a mobile application interface for completing a report. At the top, there is a hamburger menu icon and the title 'kerjakan laporan'. Below this is a header bar with the text 'Form Laporan Pertemuan x'. The main content area contains a form with the following elements:

- A sub-header 'Eksport PDF | Header Editor Text'.
- Input fields for 'Nama' (value: userxxxx), 'Npm' (value: 123xxxxxxx), and 'E-mail' (value: user@mail.com).
- A section titled 'hasil kegiatan pertemuan xxxxxx' containing a placeholder box for 'foto hasil praktikum'.
- A section titled 'deskripsi xxxxxxxx' containing another placeholder box for 'foto hasil praktikum'.
- A button at the bottom labeled 'Kumpulkan Laporan'.

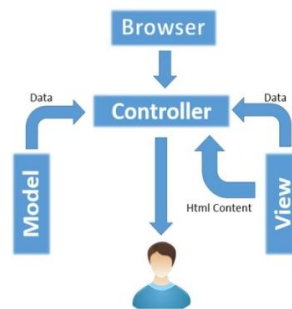
Gambar 3. 27 Interface Halaman Laporan Lengkap Mahasiswa

3.1.5 Development Delevery and Feedback

Tahapan ini melakukan proses produksi perangkat secara benar dan baik sehingga dapat digunakan oleh pengguna. //tambah lagi

3.2 Penerapan *Framework CodeIgniter*

CodeIgniter adalah sebuah *web application Framework* yang bersifat *open source* digunakan untuk membangun aplikasi PHP dinamis (Abidilah, n.d.). Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. *CodeIgniter* mengikuti pola kode MVC, sehingga dapat dibagi menjadi 3 bagian, seperti gambar 3.1:



Gambar 3.28 Konsep MVC (Model View Controller)

Uraian dari konsep MVC adalah sebagai berikut :

1. Model

Objek model adalah bagian dari aplikasi yang mengimplementasi logika untuk *domain* data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari *database* atau menyimpan data ke dalam *database*.

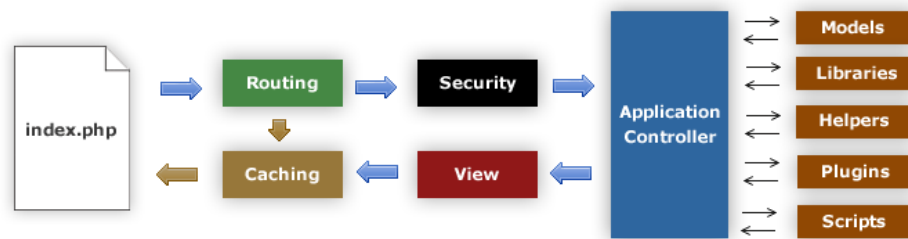
2. View

View adalah komponen yang menampilkan antarmuka untuk pengguna (*user interface*, UI) aplikasi. Antarmuka ini dibuat berdasarkan data dari *model*.

3. Controller

Controller merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan model, dan memilih view mana yang digunakan untuk merender data.

Untuk cara kerja *Framework CodeIgniter* dalam memproses permintaan dari pengguna (user) pada sebuah aplikasi web. Diagram alur ini mengilustrasikan langkah-langkah yang dilakukan oleh *framework* mulai dari menerima permintaan, memproses permintaan, mengembalikan *respons*, dan melakukan *caching* jika diperlukan.



Gambar 3. 29 Diagram Alur Kerja Framework CodeIgniter.

Secara umum, *application flowchart CodeIgniter* mengikuti langkah-langkah berikut:

1. *Index.php*: Ketika permintaan dari pengguna diterima, *framework CodeIgniter* akan memulai proses di file *index.php*, yang merupakan titik masuk pertama untuk setiap permintaan aplikasi web.
2. *Routing*: Setelah menerima permintaan, *CodeIgniter* akan mencocokkan URL permintaan dengan konfigurasi *routing* untuk menentukan *controller* mana yang harus dipanggil untuk menangani permintaan tersebut.
3. *Security*: Setelah *routing*, *CodeIgniter* akan memeriksa keamanan permintaan, termasuk memeriksa apakah permintaan valid, apakah terdapat upaya *hacking*, dan apakah pengguna memiliki hak akses yang sesuai.
4. *Application Controller*: Setelah tahap keamanan, *CodeIgniter* akan memanggil *controller* yang sesuai dengan permintaan. *Controller* akan mengambil data dari model, melakukan pemrosesan dan persiapan data, dan mengembalikan *view* yang sesuai dengan permintaan.
5. *View*: Setelah *controller* selesai memproses data, *view* akan digunakan untuk menampilkan data yang dihasilkan kepada pengguna. *View* akan menampilkan antarmuka pengguna (UI) yang sesuai dengan data yang diterima dari *controller*.
6. *Caching*: Setelah *view* ditampilkan, *CodeIgniter* dapat melakukan *caching* untuk mempercepat pengiriman *respons* pada permintaan selanjutnya dengan URL yang sama. *Cache* dapat disimpan dalam memori atau *disk*, tergantung pada konfigurasi.
7. *Index.php (lagi)*: Setelah semua proses selesai, *respons* akan dikirimkan ke pengguna dan aplikasi kembali ke titik masuk pada *index.php*.

Diagram alur aplikasi *CodeIgniter* akan memperjelas bagaimana setiap langkah saling terhubung dan membentuk alur kerja yang jelas.

3.3 User Experience Questionnaire (UEQ)

Evaluasi *user experience* pada sistem informasi akademik mahasiswa dilakukan dengan metode *User Experience Questionnaire* (UEQ). Faktor-faktor yang diukur dengan menggunakan UEQ adalah *attractive*, *efficiency*, *perspicuity*, *dependability*, dan *stimulation*. Metode yang dilakukan pada penelitian ini sebagai berikut:

a) Studi literatur

Pada penelitian ini tahap awal yang dilakukan adalah studi terhadap literatur yang berkaitan dengan *user experience*, *user experience questionnaire* serta penelitian terdahulu yang dapat dijadikan sebagai rujukan dari penelitian ini.

b) Penentuan Populasi dan Sampel Penelitian

Responden penelitian ini adalah dosen IIB Darmajaya yang memiliki status aktif pada semester ganjil tahun akademik 2022/2023. Pada semester ganjil tahun akademik 2022/2023 ini rata-rata dosen sudah menggunakan sistem akademik lebih kurang 2,5 tahun. Jumlah sampel ditentukan dengan menggunakan rumus Slovin pada persamaan 1 berikut:

$$n = \frac{N}{1 + (N \times e^2)}$$

Dimana:

n = jumlah sampel

N = jumlah populasi

e = *margin error*

Pada penelitian ini jumlah dosen yang mengajar mata kuliah praktikum pada program studi teknik informatika IIB Darmajaya adalah 10 orang dengan *margin of error* 10% atau 0,1. Sehingga dengan menggunakan persamaan 1 dihitung jumlah sampel minimal sebagai berikut:

$$n = \frac{10}{1 + (10 \times (0,1)^2)}$$

$$n = 9.09$$

Berdasarkan hasil perhitungan dengan menggunakan persamaan 1, maka jumlah populasi minimal adalah 9.09 yang dibulatkan menjadi 9.

c) Pengumpulan data menggunakan UEQ

Setelah menghitung jumlah sampel minimal, selanjutnya dilakukan pengumpulan data dengan menggunakan UEQ. Proses pengumpulan data menggunakan UEQ dapat dilihat pada Gambar 1 berikut:



Gambar 3.30 Proses Pengumpulan Data Menggunakan UEQ

Proses evaluasi menggunakan UEQ dimulai dari responden diminta untuk menggunakan semua fitur yang ada pada sistem informasi akademik mahasiswa. Tujuannya adalah agar pengguna mendapatkan umpan balik yang baik dan hasil UEQ yang akurat. Pengguna dapat mengisi UEQ sesuai dengan impresi yang mereka rasakan ketika menggunakan website, baik dari dalam segi fungsi, warna, jenis tulisan, tata letak dan lain-lain.

Setelah menggunakan sistem tersebut, pengguna mengisi kuesioner yang diberikan. Pengguna mengisi kuesioner UEQ, melalui form yang sudah disebarkan kepada pengguna. Pengisian UEQ akan diarahkan sesuai dengan ketentuan UEQ, dimana tidak selamanya poin 7 merupakan nilai tertinggi pada kuesioner ini.

Pada penelitian ini pengukuran *user experience* dilakukan dengan menggunakan kuesioner UEQ yang terdiri dari 26 pernyataan seperti pada Gambar 2. Terdapat 6 faktor *user experience* (UX) yang diukur menggunakan UEQ, yaitu daya tarik (*attractiveness*), kejelasan (*perspicuity*), efisiensi (*efficiency*), ketepatan (*dependability*), stimulasi (*stimulation*) dan kebaruan (*novelty*).

	1	2	3	4	5	6	7		
menyusahkan	○	○	○	○	○	○	○	menyenangkan	1
tak dapat dipahami	○	○	○	○	○	○	○	dapat dipahami	2
kreatif	○	○	○	○	○	○	○	monoton	3
mudah dipelajari	○	○	○	○	○	○	○	sulit dipelajari	4
bermanfaat	○	○	○	○	○	○	○	kurang bermanfaat	5
membosankan	○	○	○	○	○	○	○	mengasyikkan	6
tidak menarik	○	○	○	○	○	○	○	menarik	7
tak dapat diprediksi	○	○	○	○	○	○	○	dapat diprediksi	8
cepat	○	○	○	○	○	○	○	lambat	9
berdaya cipta	○	○	○	○	○	○	○	konvensional	10
menghalangi	○	○	○	○	○	○	○	mendukung	11
baik	○	○	○	○	○	○	○	buruk	12
rumit	○	○	○	○	○	○	○	sederhana	13
tidak disukai	○	○	○	○	○	○	○	menggembirakan	14
lazim	○	○	○	○	○	○	○	terdepan	15
tidak nyaman	○	○	○	○	○	○	○	nyaman	16
aman	○	○	○	○	○	○	○	tidak aman	17
memotivasi	○	○	○	○	○	○	○	tidak memotivasi	18
memenuhi ekspektasi	○	○	○	○	○	○	○	tidak memenuhi ekspektasi	19
tidak efisien	○	○	○	○	○	○	○	efisien	20
jelas	○	○	○	○	○	○	○	membingungkan	21
tidak praktis	○	○	○	○	○	○	○	praktis	22
terorganisasi	○	○	○	○	○	○	○	berantakan	23
atraktif	○	○	○	○	○	○	○	tidak atraktif	24
ramah pengguna	○	○	○	○	○	○	○	tidak ramah pengguna	25
konservatif	○	○	○	○	○	○	○	inovatif	26

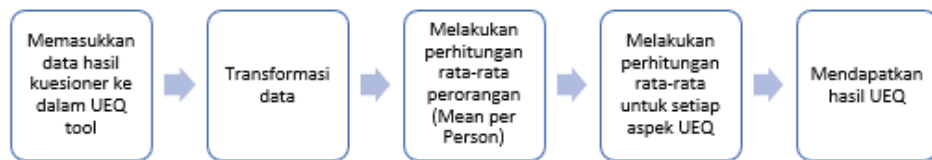
Gambar 3. 31 Pertanyaan UEQ

Data yang telah dikumpulkan dari responden kemudian dimasukan ke dalam UEQ *tool* data *analysis* dan dijadikan sebagai nilai masukan untuk menghitung nilai pengalaman pengguna dari aplikasi pengisian lembar kegiatan praktikum (SILAPRAK).

Pengujian realibilitas terhadap data kuesioner dilakukan menggunakan koefisien *cornbach alpha*. Koefisien *cornbach alpha* menggambarkan konsistensi untuk semua skala faktor UX. Hasil pengujian menunjukkan nilai *cornbach alpha* dari seluruh faktor UX diatas skala yang dapat diterima ($>0,7$), yaitu daya tarik (0,9), kejelasan (0,79), efisiensi (0,88), ketepatan (0,83), stimulasi (0,81), dan kebaruan (0,83). Hal ini menunjukkan bahwa data yang dianalisis menggunakan UEQ reliabel.

d) Pengolahan data dan analisis

Pengolahan data yang diperoleh dari responden kemudian diolah menggunakan UEQ *tools*. Hasil pengolahan data kemudian analisa untuk memperoleh informasi *user experience* dari sistem informasi akademik mahasiswa. Proses pengolahan data dapat dilihat pada Gambar 3 berikut:



Gambar 3.32 Proses Pengolahan Data UEQ