

BAB 3

MODEL SISTEM

3.1 Model Pengembangan

Model pengembangan sistem yang digunakan pada aplikasi ini adalah Model *Waterfall*. Menurut (Almuttaqin, 2016) metode ini merupakan metode dengan model sekuensial, sehingga penyelesaian satu set kegiatan menyebabkan dimulainya aktivitas berikutnya. Fokus terhadap masing-masing fase dapat dilakukan maksimal karena tidak adanya pengerjaan yang sifatnya paralel. Metode *Waterfall* terbagi menjadi lima tahapan yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Seluruh kebutuhan aplikasi didapatkan pada fase ini, termasuk didalamnya kegunaan aplikasi yang diharapkan pengguna dan batasan aplikasi. Proses pengumpulan kebutuhan dilakukan dengan cara menyebar survei dan studi pustaka. Survei dilakukan kepada 81 Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang untuk mengetahui permasalahan dan solusi yang diinginkan dalam aplikasi. Penulis juga melakukan studi kepustakaan melalui referensi di perpustakaan maupun internet yang berhubungan dengan permasalahan dalam penulisan tugas akhir ini.

2. Perancangan (Desain)

Tahap ini bertujuan untuk memberikan gambaran seperti apa sistem yang akan dibuat dan bagaimana *interface* untuk setiap kegiatannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan. Pada tahap desain penulis melakukan proses yang berfokus pada desain pembuatan program mulai dari perancangan pembuatan model sistem menggunakan *Unified Modelling Language* (UML) yang terdiri dari *use case diagram* dan *activity diagram*, *Entity Relationship Diagram* (ERD), dan antarmuka. Dalam pembuatan *database* aplikasi ini menggunakan MySQL yang merupakan *Software Database Management System* (RDBMS) yang dapat mengelola *database* dengan cepat dan menampung data dalam jumlah besar.

3. Implementasi (*Coding*)

Tahap ini penulis mentranslasikan desain ke dalam bahasa pemrograman. Bahasa pemrograman yang digunakan yaitu bahasa pemrograman PHP (*Hypertext Preprocessor*) dengan *Framework CodeIgniter 3*.

4. Pengujian (*Testing*)

Pada tahap ini dilakukan pengujian untuk mengetahui apakah aplikasi telah sesuai atau belum sesuai dengan desainnya. Pengujian pada aplikasi ini berfokus pada fungsional menggunakan sistem pengujian *Black Box* dengan teknik *Equivalence Partitioning* dan kemudahan antarmuka menggunakan pengujian *usability* dengan metode kuesioner menggunakan Google Formulir kepada Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang yang diukur menggunakan skala *likert*.

5. Pemeliharaan (*Maintenance*)

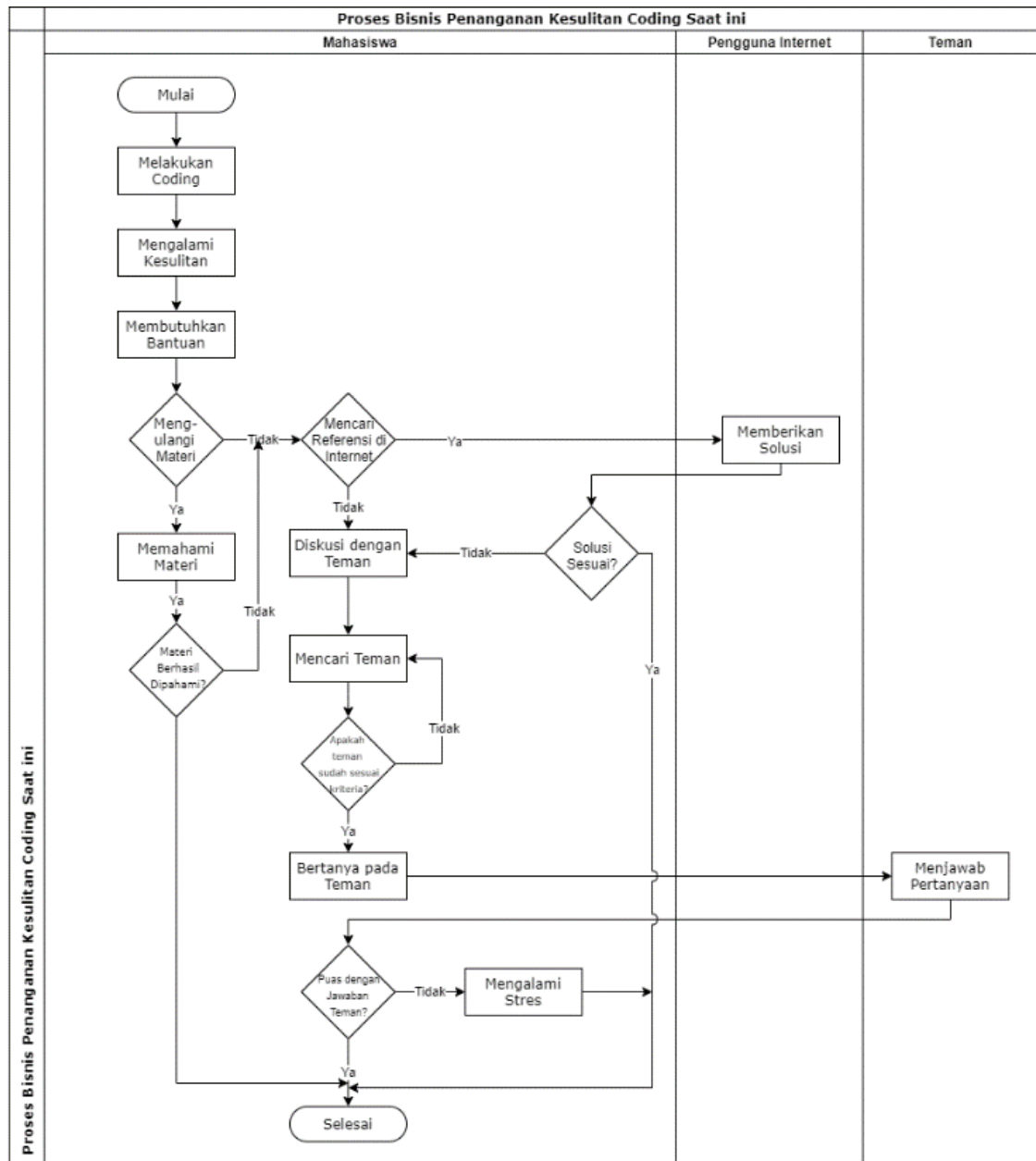
Ini merupakan tahap terakhir dalam model *Waterfall*. Aplikasi yang telah dibangun, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

3.2 Proses Bisnis

Proses bisnis adalah suatu kumpulan aktivitas atau pekerjaan terstruktur yang saling terkait untuk menyelesaikan suatu masalah tertentu atau yang menghasilkan produk atau layanan.

3.2.1 Proses Bisnis Saat Ini

Proses bisnis saat ini adalah proses yang digunakan mahasiswa dalam mengatasi kesulitan dalam pembelajaran *coding* sebelum dibangunnya aplikasi ini. Proses bisnis yang sedang berjalan saat ini dapat dilihat pada Gambar 3.1 di mana proses bisnis ini digunakan untuk analisa agar didapatkan proses bisnis yang baru setelah sistem diimplementasikan.



Gambar 3. 1 Proses Bisnis Saat Ini

Berdasarkan Gambar 3.1 dapat dijelaskan bahwa terdapat mahasiswa yang sedang kesulitan dalam melakukan *coding* dan membutuhkan bantuan untuk mengatasi masalah tersebut. Dari survei yang didapatkan terdapat tiga cara yang akan dilakukan mahasiswa dalam menangani kesulitan *coding* yaitu mempelajari ulang materi, mencari referensi di internet, dan berdiskusi dengan teman. Berikut ini merupakan alur mahasiswa dalam penanganan kesulitan *coding* saat ini.

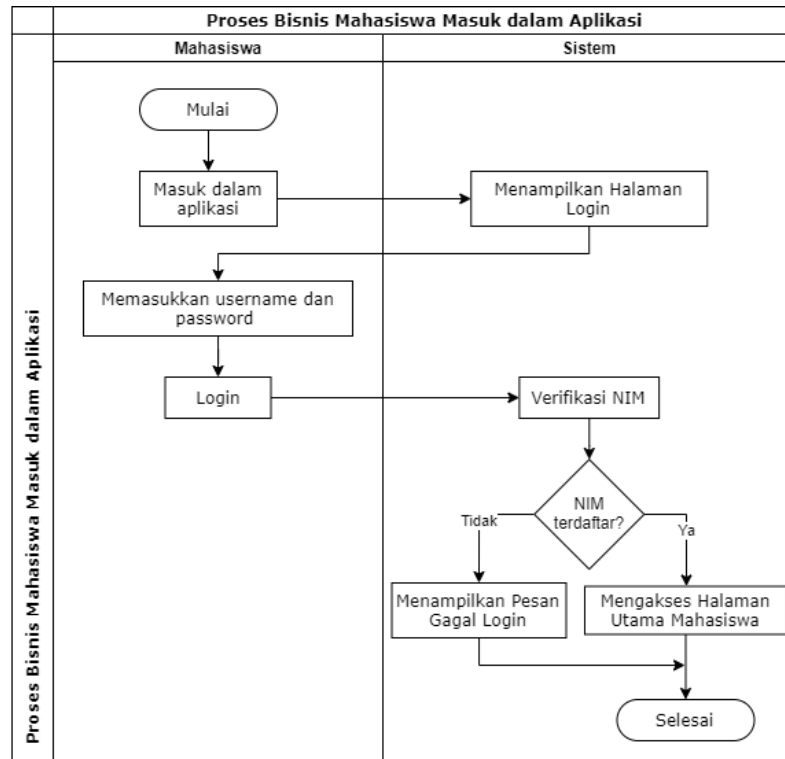
1. Mahasiswa akan mempelajari ulang materi yang tidak mereka pahami. Jika cara ini berhasil membuat mereka paham maka proses penanganan kesulitan *coding* akan berakhir. Sedangkan jika cara tersebut gagal maka akan digunakan cara yang kedua yaitu mencari referensi di internet.
2. Mahasiswa akan mencari referensi di internet dimana di dalam internet banyak pengguna internet yang akan menyediakan solusi terhadap permasalahan yang ditanyakan. Jika cara kedua menghasilkan referensi yang sesuai maka proses penanganan kesulitan *coding* akan berakhir. Sedangkan jika cara tersebut gagal maka akan digunakan cara ketiga yaitu mahasiswa akan berdiskusi dengan teman.
3. Melalui cara ketiga ini mahasiswa harus mencari teman sesuai dengan kriteria yang dibutuhkan terlebih dahulu. Jika belum menemukan kriteria teman yang sesuai maka mahasiswa tersebut akan terus mencarinya. Ketika sudah menemukan teman yang sesuai dengan kriteria maka mahasiswa akan melakukan diskusi dengan melakukan tanya jawab.
4. Setelah diskusi dilakukan, dapat diketahui apakah hasil diskusi tersebut membuahkan hasil yang baik atau tidak terhadap pemahaman mahasiswa.
5. Jika materi *coding* berhasil dipahami oleh mahasiswa maka dapat dipastikan proses penanganan kesulitan *coding* pada cara ini akan berakhir. Sedangkan setelah melakukan tiga cara di atas mahasiswa tersebut masih merasa kesulitan dalam melakukan *coding* maka mahasiswa tersebut akan menjadi stres karena tertekan.

3.2.2 Proses Bisnis Usulan

Proses bisnis usulan merupakan proses bisnis untuk mengatasi kesulitan bagi mahasiswa dalam proses pembelajaran *coding* pada aplikasi pembelajaran *coding* berbasis *website* yang dibangun. Pada aplikasi yang dibangun kami mengusulkan untuk menggabungkan ketiga cara dalam mengatasi kesulitan *coding* pada Gambar 3.1 ke dalam satu aplikasi.

Aplikasi ini terdiri atas empat pengguna yaitu admin, mahasiswa, tutor, dan calon tutor. Admin bertugas dalam pengelolaan seluruh data yang ada dalam aplikasi ini mulai dari data mahasiswa, data tutor, data kategori materi, data forum, melihat kritik dan saran, serta memverifikasi calon tutor untuk diterima atau tidak. Sedangkan mahasiswa sebagai

pengakses materi akan langsung dapat mengakses sistem karena data mahasiswa akan secara otomatis terdaftar dalam sistem yang ditambahkan oleh admin. Pada Gambar 3.2 berikut ini merupakan proses bisnis bagaimana cara mahasiswa dapat masuk ke dalam sistem.



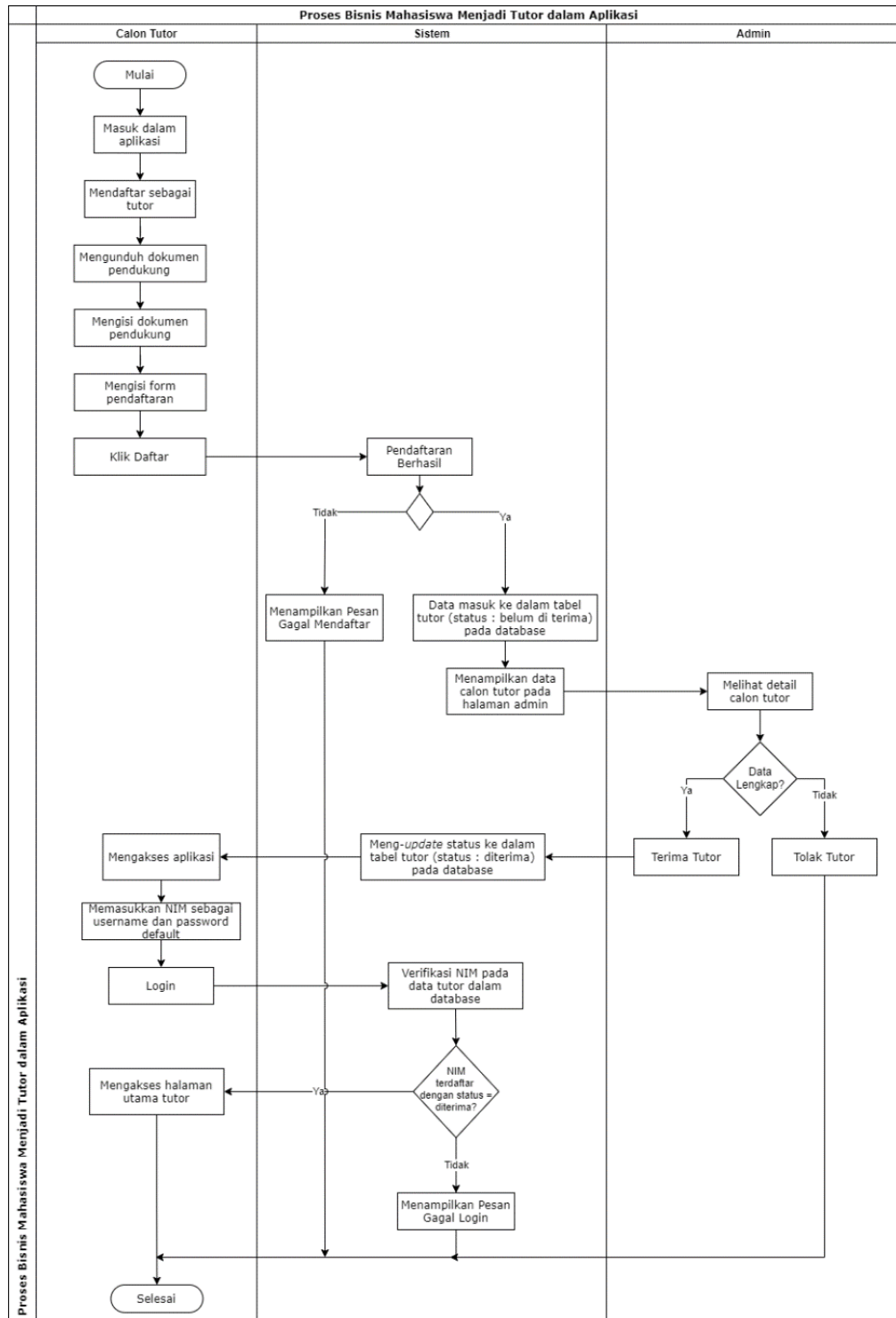
Gambar 3. 2 Proses Bisnis Cara Mahasiswa Masuk dalam Aplikasi

Berdasarkan Gambar 3.2 dapat dijelaskan bahwa proses bisnis untuk mahasiswa agar dapat masuk ke dalam aplikasi yaitu:

1. Mahasiswa masuk ke dalam aplikasi dan sistem akan menampilkan halaman *login*.
2. Mahasiswa memasukkan NIM sebagai *username* dan *password* kemudian *login*.
3. Ketika proses *login*, sistem akan memeriksa NIM yang dimasukkan, apakah NIM tersebut terdaftar pada data mahasiswa atau tidak. Kemudian jika NIM tersebut terdaftar maka mahasiswa berhasil dalam mengakses sistem. Apabila pada proses *login* NIM yang dimasukkan tidak terdaftar di dalam sistem maka sistem akan menampilkan pesan gagal dalam melakukan *login*.

Sedangkan yang bertugas sebagai tutor juga merupakan mahasiswa yang dinilai sudah memiliki kemampuan dalam melakukan *coding*. Berikut ini merupakan proses bisnis

bagaimana cara mahasiswa dalam mendaftarkan dirinya sebagai tutor di dalam sistem yang ditunjukkan pada Gambar 3.3 di bawah ini.



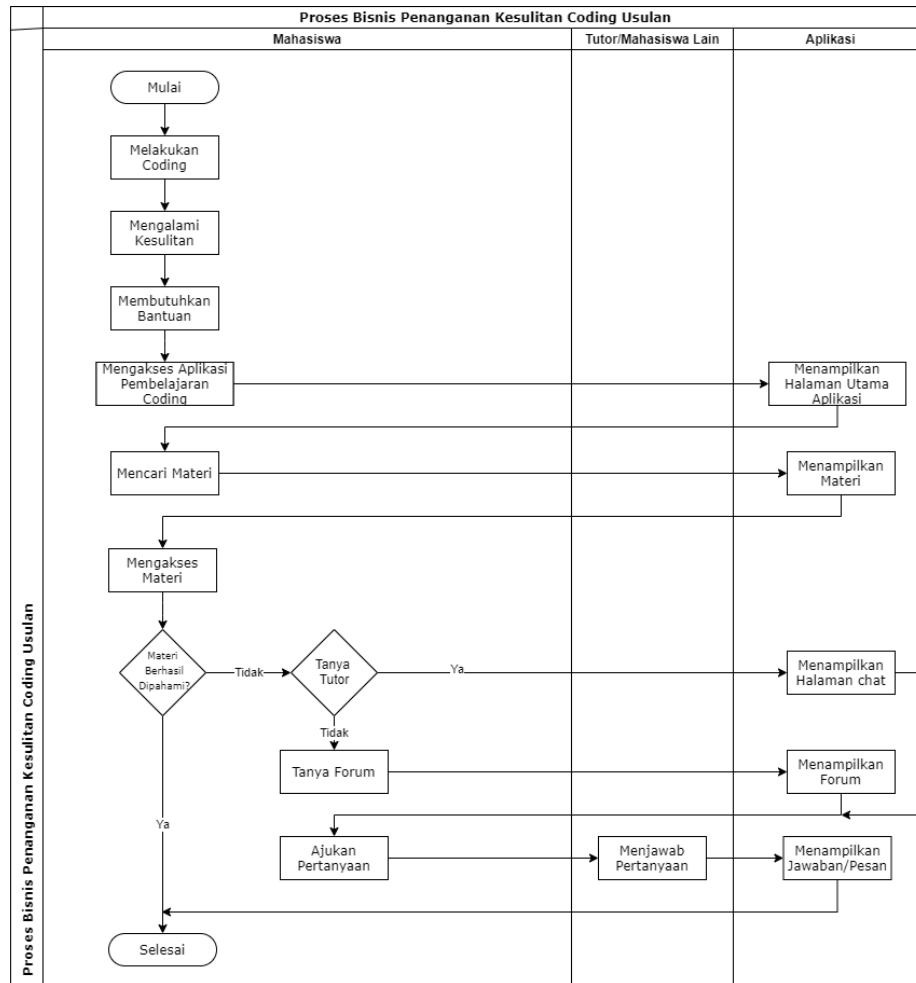
Gambar 3. 3 Proses Bisnis Cara Mahasiswa Menjadi Tutor

Berdasarkan Gambar 3.3 dapat dijelaskan bahwa proses bisnis mahasiswa atau calon tutor dalam mendaftar sebagai tutor yaitu:

1. Calon tutor harus mengakses aplikasi terlebih dahulu kemudian memilih menu daftar tutor.
2. Sebelum melanjutkan mendaftar, calon tutor diarahkan untuk mengunduh dokumen pendukung terlebih dahulu. Dokumen ini berisi surat bukti kesediaan bahwa orang tersebut benar-benar ingin menjadi tutor pada aplikasi ini serta *draft* nilai mata kuliah sesuai dengan materi yang akan diambil.
3. Setelah mengisi dokumen tersebut, calon tutor melanjutkan untuk mengisi *form* pendaftaran pada sistem lalu klik daftar.
4. Sistem akan memeriksa kelengkapan data pendaftaran. Jika proses pemeriksaan ini berhasil data calon tutor akan masuk ke dalam *database* dengan status masih belum disetujui, data ini akan ditampilkan di halaman verifikasi pada admin. Jika dalam proses pemeriksaan ini tidak berhasil maka calon tutor gagal dalam melakukan pendaftaran.
5. Kemudian admin bertugas untuk memeriksa kelengkapan dokumen serta data diri calon tutor. Jika dokumen lengkap dan benar maka admin berhak untuk menerimanya. Ketika calon tutor tersebut diterima maka status calon tutor pada *database* akan berubah menjadi “diterima”. Hal ini berarti bahwa calon tutor tersebut sudah mendapatkan hak akses sebagai tutor.
6. Tutor melakukan *login* dengan NIM lalu sistem akan memeriksa NIM tersebut apakah telah terdaftar ke dalam data tutor dengan status diterima atau tidak, jika kondisi tersebut benar maka tutor berhasil dalam mengakses sistem. Jika proses ini gagal maka sistem akan menampilkan pesan bahwa proses *login* gagal.

Seperti yang telah dijelaskan di awal, aplikasi ini akan menggabungkan tiga cara penanganan kesulitan *coding* pada Gambar 3.1 dalam satu aplikasi. Hal ini berarti bahwa tiga cara tersebut akan dijadikan sebagai fitur utama dalam aplikasi ini. Fitur-fitur tersebut yaitu penyediaan materi oleh tutor, melakukan diskusi dengan mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang beserta tutor pada forum yang disediakan, serta dapat melakukan komunikasi dengan tutor secara langsung melalui *private chat* dalam aplikasi ini.

Proses bisnis cara penanganan kesulitan *coding* yang dilakukan oleh mahasiswa dalam aplikasi yang dibangun dapat dilihat pada Gambar 3.4 di bawah ini.



Gambar 3. 4 Proses Bisnis Sistem Penanganan Kesulitan *Coding* Usulan

Pada Gambar 3.4 menunjukkan bahwa proses penanganan kesulitan *coding* sudah terhubung ke dalam sebuah sistem atau aplikasi. Ketika mahasiswa mengalami kesulitan dalam melakukan *coding* dan membutuhkan bantuan, mahasiswa hanya perlu mengakses aplikasi pembelajaran *coding* yang telah dibangun. Aplikasi ini menggabungkan tiga cara penanganan kesulitan *coding* pada proses bisnis lama ke dalam satu sistem sehingga hanya dengan mengakses aplikasi ini mahasiswa dapat melakukan tiga cara penanganan kesulitan *coding* sekaligus. Untuk mengatasi kesulitan *coding* menggunakan aplikasi ini yang dilakukan mahasiswa yaitu:

1. Mahasiswa dapat mencari materi yang disediakan langsung oleh tutor.

2. Sistem akan menampilkan materi yang di *request* oleh mahasiswa tersebut.
3. Ketika mahasiswa masih belum memahami materi, maka mahasiswa dapat melakukan tanya jawab dengan tutor atau melakukan diskusi dengan forum untuk membantu dalam menangani kesulitan *coding*.
4. Setelah ada jawaban atau pesan masuk dari tutor/mahasiswa lain, sistem akan menampilkan jawaban atau pesan tersebut kepada mahasiswa.

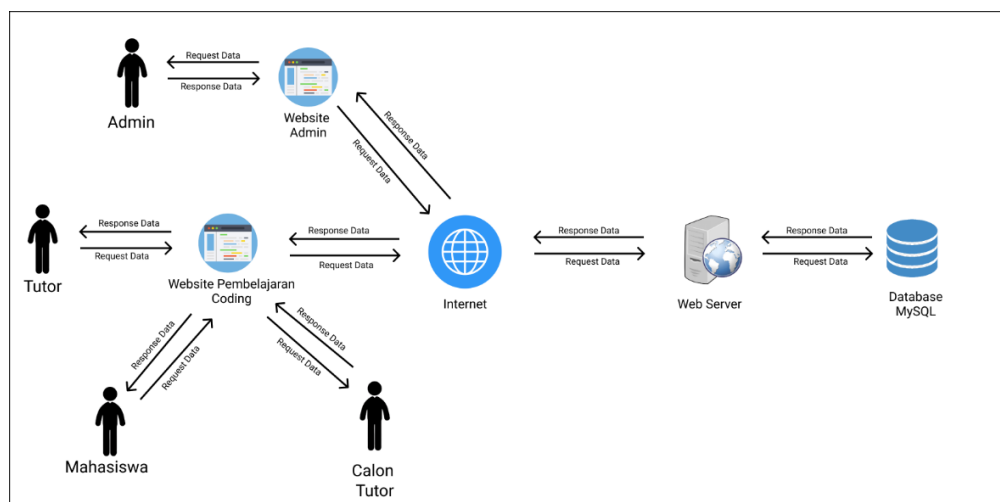
3.3 Analisis dan Desain Usulan

Analisis yang dilakukan adalah mengidentifikasi masalah yang terdapat di dalam kasus yang diambil. Dari perumusan masalah yang telah ditentukan, permasalahan yang terjadi adalah:

1. Mahasiswa mengalami kesulitan dalam melakukan *coding*.
2. Mahasiswa membutuhkan fasilitas yang membantu mereka mengatasi kesulitan dalam melakukan pembelajaran *coding*.
3. Mahasiswa membutuhkan penyajian materi *coding* yang mudah dipahami.

3.3.1 Arsitektur Sistem

Arsitektur sistem adalah sebuah arsitektur yang menggambarkan komponen-komponen penyusun yang saling terhubung di dalam suatu aplikasi. Perancangan arsitektur sistem aplikasi pembelajaran *coding* dapat dilihat pada Gambar 3.5 di bawah ini.



Gambar 3.5 Arsitektur Sistem

Berdasarkan Gambar 3.5 sistem terdiri dari empat pengguna yaitu admin, tutor, mahasiswa, dan calon tutor. Di antara keempat pengguna yang telah disebutkan, satu dari ketiga pengguna memiliki halaman *website* yang berbeda yaitu *website* admin hanya dapat diakses oleh admin, sedangkan untuk tiga pengguna lainnya yaitu tutor, calon tutor, dan mahasiswa mengakses *website* pembelajaran *coding*.

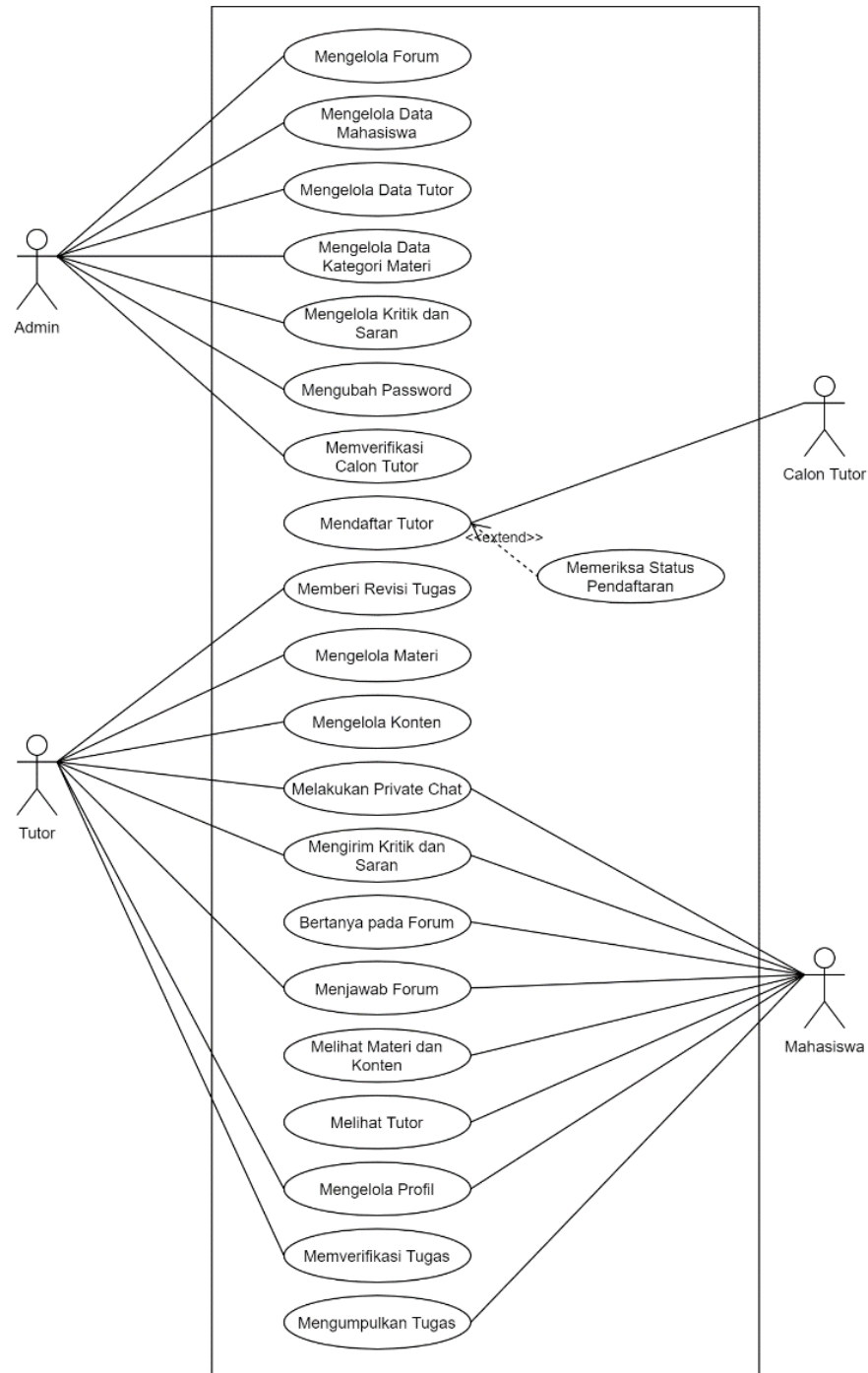
Admin merupakan pengguna yang memiliki akses untuk mengelola semua data yang ada di dalam *website* admin mulai dari data mahasiswa, data tutor, forum, data kategori materi, memverifikasi data calon tutor, serta dapat melihat kritik dan saran dari mahasiswa atau tutor. Pengguna kedua adalah tutor, merupakan pengguna yang memiliki akses untuk menyajikan dan mengelola data materi dan konten yang mereka miliki, memverifikasi tugas, tutor juga dapat menjawab pertanyaan yang ada pada forum, serta dapat berkomunikasi langsung dengan para mahasiswa melalui *private chat* yang disediakan oleh sistem. Untuk pengguna ketiga yaitu Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang, merupakan pengguna yang memiliki akses untuk melihat materi yang disediakan oleh tutor, melakukan diskusi pada forum, dan berkomunikasi dengan tutor. Pengguna terakhir adalah calon tutor yang juga merupakan mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang yang berpeluang dapat menjadi seorang tutor dalam aplikasi ini. Calon tutor hanya memiliki akses untuk mendaftarkan diri sebagai tutor dan memeriksa status pendaftaran mereka.

Melalui *website* tersebut, baik *website* admin maupun *website* aplikasi *coding* masing-masing pengguna menggunakannya sebagai jalan dalam melakukan permintaan (*request*) dan pengiriman data kepada *web server* jika pengguna dalam keadaan terkoneksi dengan internet. *Web server* akan meneruskan permintaan pengguna ke dalam *database server*. *Database server* yang digunakan adalah MySQL sebagai tempat penyimpanan semua data pada aplikasi ini. Kemudian *database server* melalui *web server* mengirim kembali data yang diminta kepada pengguna atau dapat disebut dengan *response data* ke dalam masing-masing *website* pengguna.

3.3.2 Use Case Diagram

Use case diagram merupakan gambaran skenario serta interaksi antara pengguna (aktor) dengan aktivitas atau *behavior* apa saja yang dapat dilakukan dalam sistem atau

aplikasi yang dibangun. *Use case diagram* juga merupakan bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case diagram* pada aplikasi *coding* dapat dilihat pada Gambar 3.6 di bawah ini.



Gambar 3. 6 Use Case Diagram

Berdasarkan *use case diagram* pada Gambar 3.6 didapatkan definisi aktor dan definisi *use case*. Penjabaran definisi aktor dapat dilihat pada Tabel 3.1 dimana pada tabel ini dijelaskan peran dari aktor tersebut dalam aplikasi yang dibangun.

Tabel 3. 1 Definisi Aktor

No	Aktor	Keterangan
1.	Admin	Admin merupakan aktor yang memiliki hak akses yang lebih tinggi dari aktor-aktor lainnya. Karena admin pada aplikasi ini dapat melakukan pengelolaan seluruh data pada sistem seperti melihat, menambah, menghapus, dan mengubahnya. Mulai dari data mahasiswa, data tutor, data kategori materi, dan data forum. Selain itu admin juga berhak melihat data kritik dan saran yang dikirimkan oleh tutor dan mahasiswa serta memverifikasi pendaftaran tutor dimana admin dapat menerima dan menolak pendaftar.
2.	Tutor	Tutor merupakan mahasiswa yang berperan sebagai pengajar dengan cara memberikan materi kepada mahasiswa dalam aplikasi ini. Tutor juga memiliki hak akses dalam pengelolaan data miliknya seperti mengelola data materi, konten, dan data profil. Selain itu tutor dapat melakukan verifikasi tugas dan berkomunikasi dengan mahasiswa lain melalui fasilitas <i>private chat</i> yang disediakan oleh sistem serta tutor juga dapat menjawab pertanyaan dalam forum yang diajukan oleh mahasiswa lain. Tutor juga dapat mengirim kritik dan saran sebagai penilaian untuk aplikasi ini.
3.	Mahasiswa	Mahasiswa merupakan aktor yang berperan sebagai pengakses materi yang diberikan oleh tutor. Mahasiswa dapat melihat daftar materi dan konten, daftar tutor, mengumpulkan tugas yang diberikan oleh tutor, dan dapat melakukan komunikasi dengan tutor melalui fasilitas <i>private chat</i> yang disediakan oleh

		sistem serta mahasiswa dapat melakukan diskusi pada forum. Untuk pengelolaan data, mahasiswa hanya dapat mengelola data diri dengan memperbarui data mahasiswa tersebut. Mahasiswa juga dapat mengirim kritik dan saran sebagai penilaian untuk aplikasi ini.
4.	Calon Tutor	Calon tutor merupakan mahasiswa yang akan mendaftar sebagai tutor dalam aplikasi. Aktor ini hanya dapat mengakses sistem untuk melakukan pendaftaran dan memeriksa status pendaftaran mereka.

Sedangkan definisi *use case* menjelaskan proses yang terjadi di dalam suatu *use case* tersebut. Penjabaran definisi *use case* dapat dilihat pada Tabel 3.2 sebagai berikut

Tabel 3. 2 Definisi *Use Case*

No	<i>Use Case</i>	Keterangan
1.	Mengelola Forum	Proses untuk melakukan pengelolaan forum dengan melihat dan menghapus forum.
2.	Mengelola Data Mahasiswa	Proses untuk melakukan pengelolaan data mahasiswa dengan menambah, mengubah, melihat, dan menghapus data mahasiswa.
3.	Mengelola Data Tutor	Proses untuk melakukan pengelolaan data tutor dengan melihat dan menghapus data tutor.
4.	Mengelola Data Kategori Materi	Proses untuk melakukan pengelolaan data kategori materi dengan menambah, mengubah, melihat, dan menghapus data kategori materi.
5.	Mengelola Kritik dan Saran	Proses untuk melihat detail dan menghapus kritik dan saran yang dikirimkan oleh tutor dan mahasiswa.
6.	Mengubah <i>Password</i>	Proses untuk melakukan perubahan <i>password</i> .
7.	Memverifikasi Calon Tutor	Proses untuk melakukan penerimaan atau penolakan terhadap calon tutor.

8.	Mendaftar Tutor	Proses mendaftarkan diri mahasiswa untuk mendapatkan hak akses sebagai tutor.
9.	Memeriksa Status Pendaftaran	Proses untuk melihat status pendaftaran dalam sistem apakah status pendaftar sudah diterima atau belum.
10.	Mengelola Materi	Proses untuk melakukan pengelolaan data materi dengan menambah, mengubah, melihat, dan menghapus data materi
11.	Mengelola Konten	Proses untuk melakukan pengelolaan konten dengan menambah, mengubah, melihat, dan menghapus konten yang ada di setiap materi.
12.	Melakukan <i>Private Chat</i>	Proses untuk melakukan balas kirim pesan antara mahasiswa dengan tutor di dalam aplikasi.
13.	Mengirim Kritik dan Saran	Proses untuk melakukan pengiriman kritik dan saran untuk penilaian kinerja sistem.
14.	Bertanya pada Forum	Proses untuk melakukan pembuatan forum dalam sistem.
15.	Menjawab Forum	Proses untuk menjawab pertanyaan pada forum dalam sistem.
16.	Melihat Materi dan Konten	Proses untuk melihat semua daftar materi dan konten yang ada dalam sistem.
17.	Melihat Tutor	Proses untuk melihat semua daftar tutor yang ada dalam sistem.
18.	Mengelola Profil	Proses untuk memperbarui data profil yang sudah ada.
19.	Mengumpulkan Tugas	Proses untuk mengumpulkan tugas di setiap konten yang telah diberikan.
20.	Memverifikasi Tugas	Proses untuk memverifikasi tugas yang telah dikumpulkan.
21.	Memberi Revisi Tugas	Proses untuk memberi revisi pada tugas mahasiswa yang masih kurang tepat.

Berikut ini merupakan deskripsi *use case* dari masing-masing *use case* yang telah didefinisikan yaitu:

a. Nama *Use Case* : Mengelola Forum

Deskripsi *use case* mengelola forum ditunjukkan pada Tabel 3.3 sebagai berikut.

Tabel 3. 3 Deskripsi *Use Case* Mengelola Forum

Nama <i>Use Case</i>	Mengelola Forum
Prioritas	Tinggi
Aktor Utama	Admin
Deskripsi	Aktor dapat melakukan pengelolaan forum dengan melihat detail forum dan melakukan hapus forum.
Skenario Normal	<ol style="list-style-type: none"> 1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu forum 5. Sistem menampilkan halaman forum 6. Admin memilih forum yang akan dilihat atau dihapus 7. Jika admin ingin melihat detail forum maka admin klik ikon mata dan sistem akan menampilkan detail forum, sedangkan jika admin ingin menghapus forum maka admin klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus.
Skenario Alternatif	-
Tujuan	Jika admin memilih melihat forum maka sistem akan menampilkan halaman detail forum sedangkan jika admin memilih hapus forum maka data forum akan terhapus dari <i>database</i> .

b. Nama *Use Case* : Mengelola Data Mahasiswa

Deskripsi *use case* mengelola data mahasiswa ditunjukkan pada Tabel 3.4 sebagai berikut.

Tabel 3. 4 Deskripsi *Use Case* Mengelola Data Mahasiswa

Nama Use Case	Mengelola Data Mahasiswa
Prioritas	Tinggi
Aktor Utama	Admin
Deskripsi	Aktor dapat melakukan pengelolaan data dengan melihat detail, menghapus, menambah, dan mengubah data mahasiswa.
Skenario Normal	<ol style="list-style-type: none"> 1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu mahasiswa 5. Sistem menampilkan halaman yang berisi daftar data mahasiswa 6. Jika admin ingin melihat detail mahasiswa maka admin klik ikon mata dan sistem akan menampilkan halaman detail dari mahasiswa yang dipilih 7. Jika admin ingin menambah data mahasiswa maka admin klik <i>button</i> tambah mahasiswa atau <i>button import</i> data dan sistem akan menampilkan <i>form</i> penambahan/<i>import</i> data. Admin mengisi seluruh data yang diperlukan lalu klik tambah/<i>import</i> dan sistem akan menampilkan pesan berhasil menambah data 8. Jika admin ingin mengubah data mahasiswa maka admin klik ikon pensil dan sistem akan menampilkan <i>form</i> perubahan data. Admin mengisi seluruh data yang diubah lalu klik <i>edit</i> dan sistem akan menampilkan pesan berhasil mengubah data

	9. Jika admin ingin menghapus data mahasiswa maka admin klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus
Skenario Alternatif	<ol style="list-style-type: none"> 1. Jika pada aliran menambah data mahasiswa terjadi <i>exception</i> maka admin tidak dapat menambah data mahasiswa dan admin dapat mengisi kembali data dengan benar sampai data tersimpan dalam <i>database</i> 2. Jika pada aliran ubah data mahasiswa terjadi <i>exception</i> maka admin tidak dapat mengubah data mahasiswa dan admin dapat mengisi kembali data dengan benar sehingga data yang tersimpan pada <i>database</i> dapat diperbarui
Tujuan	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman detail mahasiswa 2. Data hasil dari penambahan dan perubahan tersimpan dalam <i>database</i> 3. Data yang dihapus hilang dari <i>database</i>

c. Nama *Use Case* : Mengelola Data Tutor

Deskripsi *use case* mengelola data tutor ditunjukkan pada Tabel 3.5 sebagai berikut.

Tabel 3. 5 Deskripsi *Use Case* Mengelola Data Tutor

Nama Use Case	Mengelola Data Tutor
Prioritas	Tinggi
Aktor Utama	Admin
Deskripsi	Aktor dapat melakukan pengelolaan data tutor berupa melihat detail tutor dan menghapus tutor.
Skenario Normal	1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i>

	<ol style="list-style-type: none"> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu tutor 5. Sistem menampilkan halaman data tutor 6. Admin memilih tutor yang akan dilihat atau dihapus 7. Jika admin ingin melihat detail tutor maka admin klik ikon mata dan sistem akan menampilkan detail tutor, sedangkan jika admin ingin menghapus tutor maka admin klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus.
Skenario Alternatif	-
Tujuan	Jika admin memilih melihat tutor maka sistem akan menampilkan halaman detail tutor sedangkan jika admin memilih hapus tutor maka data tutor akan terhapus dari <i>database</i> .

d. Nama *Use Case* : Mengelola Data Kategori Materi

Deskripsi *use case* mengelola data kategori materi ditunjukkan pada Tabel 3.6 sebagai berikut.

Tabel 3. 6 Deskripsi *Use Case* Mengelola Data Kategori Materi

Nama <i>Use Case</i>	Mengelola Data Kategori Materi
Prioritas	Tinggi
Aktor Utama	Admin
Deskripsi	Aktor dapat melakukan pengelolaan data dengan melihat detail, menghapus, menambah, dan mengubah kategori materi.
Skenario Normal	1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i>

	<ol style="list-style-type: none"> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu kategori materi 5. Sistem menampilkan halaman semua data kategori materi 6. Jika admin ingin menambah data kategori materi maka admin klik <i>button</i> tambah data dan sistem akan menampilkan <i>form</i> penambahan data. Admin mengisi seluruh data yang diperlukan lalu klik tambah dan sistem akan menampilkan pesan berhasil menambah data 7. Jika admin ingin mengubah data kategori materi maka admin klik ikon pensil dan sistem akan menampilkan <i>form</i> perubahan data. Admin mengisi seluruh data yang diubah lalu klik <i>edit</i> dan sistem akan menampilkan pesan berhasil mengubah data 8. Jika admin ingin menghapus data kategori materi maka admin klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus
Skenario Alternatif	<ol style="list-style-type: none"> 1. Jika pada aliran menambah data kategori materi terjadi <i>exception</i> maka admin tidak dapat menambah data kategori materi dan admin dapat mengisi kembali data dengan benar sampai data tersimpan dalam <i>database</i> 2. Jika pada aliran ubah data kategori materi terjadi <i>exception</i> maka admin tidak dapat mengubah data kategori materi dan admin dapat mengisi kembali data dengan benar sehingga data yang tersimpan pada <i>database</i> dapat diperbarui
Tujuan	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman data kategori materi

	<ol style="list-style-type: none"> 2. Data hasil dari penambahan dan perubahan tersimpan dalam <i>database</i> 3. Data yang dihapus hilang dari <i>database</i>
--	---

e. Nama *Use Case* : Mengelola Kritik dan Saran

Deskripsi *use case* melihat kritik dan saran ditunjukkan pada Tabel 3.7 sebagai berikut.

Tabel 3. 7 Deskripsi *Use Case* Melihat Kritik dan Saran

Nama <i>Use Case</i>	Mengelola Kritik dan Saran
Prioritas	Sedang
Aktor Utama	Admin
Deskripsi	Aktor dapat melihat kritik dan saran yang dikirimkan oleh tutor dan mahasiswa.
Skenario Normal	<ol style="list-style-type: none"> 1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu kritik dan saran 5. Sistem menampilkan halaman kritik dan saran 6. Jika ingin menghapus data admin klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus
Skenario Alternatif	-
Tujuan	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman kritik dan saran 2. Data yang dihapus hilang dari <i>database</i>

f. Nama *Use Case* : Mengubah *Password*

Deskripsi *use case* mengubah *password* ditunjukkan pada Tabel 3.8 sebagai berikut.

Tabel 3. 8 Deskripsi *Use Case* Mengubah *Password*

Nama <i>Use Case</i>	Mengubah <i>Password</i>
Prioritas	Sedang
Aktor Utama	Admin
Deskripsi	Aktor dapat memperbarui data <i>password</i> yang dimiliki.
Skenario Normal	<ol style="list-style-type: none"> 1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu ubah <i>password</i> 5. Sistem menampilkan halaman <i>form</i> perubahan <i>password</i> 6. Admin memperbarui <i>password</i> yang telah ada dengan <i>password</i> baru kemudian klik simpan dan sistem akan menampilkan pesan berhasil mengubah <i>password</i>
Skenario Alternatif	Jika pada aliran ubah <i>password</i> terjadi <i>exception</i> maka admin tidak dapat mengubah <i>password</i> tersebut. Hal yang harus dilakukan yaitu admin harus mengisi kembali data dengan benar sampai data baru tersimpan di dalam <i>database</i> .
Tujuan	Data <i>password</i> yang baru dapat tersimpan dalam <i>database</i> menggantikan data <i>password</i> lama.

g. Nama *Use Case* : Memverifikasi Calon Tutor

Deskripsi *use case* memverifikasi calon tutor ditunjukkan pada Tabel 3.9 sebagai berikut.

Tabel 3. 9 Deskripsi *Use Case* Memverifikasi Calon Tutor

Nama <i>Use Case</i>	Memverifikasi Calon Tutor
Prioritas	Tinggi

Aktor Utama	Admin
Deskripsi	Aktor dapat menerima atau menolak calon tutor.
Skenario Normal	<ol style="list-style-type: none"> 1. Admin melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel admin pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> admin 4. Admin memilih menu verifikasi calon tutor 5. Sistem menampilkan halaman semua data pendaftar 6. Admin memeriksa kelengkapan berkas calon tutor 7. Jika admin akan menerima calon tutor maka admin klik ikon <i>checklist</i> dan sistem menampilkan pesan berhasil menerima tutor, sedangkan jika admin akan menolak tutor maka admin klik ikon X dan sistem menampilkan pesan berhasil menolak tutor
Skenario Alternatif	-
Tujuan	<ol style="list-style-type: none"> 1. Jika admin menerima calon tutor maka status calon tutor pada <i>database</i> akan berganti menjadi “diterima” dan calon tutor dapat melakukan <i>login</i> dengan hak akses sebagai tutor 2. Jika admin menolak calon tutor maka data calon tutor akan terhapus dalam <i>database</i>

h. Nama *Use Case* : Mendaftar Tutor

Deskripsi *use case* mendaftar tutor ditunjukkan pada Tabel 3.10 sebagai berikut.

Tabel 3. 10 Deskripsi *Use Case* Mendaftar Tutor

Nama <i>Use Case</i>	Mendaftar Tutor
Prioritas	Tinggi
Aktor Utama	Calon Tutor

Deskripsi	Proses pendaftaran untuk mendapatkan hak akses sebagai tutor.
Skenario Normal	<ol style="list-style-type: none"> 1. Calon tutor mengakses sistem 2. Calon tutor memilih tombol daftar tutor 3. Sistem menampilkan <i>form</i> pendaftaran tutor 4. Calon tutor mengunduh dokumen pendukung pendaftaran 5. Calon tutor mengisi dokumen pendukung pendaftaran 6. Calon tutor memasukkan data diri dan mengunggah dokumen pendukung pada <i>form</i> pendaftaran 7. Calon tutor klik <i>button</i> daftar dan sistem menampilkan pesan berhasil mendaftar
Skenario Alternatif	Jika pada aliran pendaftaran tutor terjadi <i>exception</i> maka proses pendaftaran tutor gagal. Hal yang harus dilakukan calon tutor yaitu mengisi kembali data dengan lengkap dan benar sampai pendaftaran berhasil
Tujuan	Data calon tutor tersimpan dalam <i>database</i> dengan status “belum diterima”.

i. Nama *Use Case* : Memeriksa Status Pendaftaran

Deskripsi *use case* memeriksa status pendaftaran ditunjukkan pada Tabel 3.11 sebagai berikut.

Tabel 3. 11 Deskripsi *Use Case* Memeriksa Status Pendaftaran

Nama <i>Use Case</i>	Memeriksa Status Pendaftaran
Prioritas	Tinggi
Aktor Utama	Calon Tutor
Deskripsi	Proses pemeriksaan status calon tutor.
Skenario Normal	<ol style="list-style-type: none"> 1. Calon tutor mengakses sistem 2. Calon tutor memilih tombol cek status pendaftaran 3. Sistem menampilkan <i>form</i> pencarian 4. Calon tutor memasukkan NIM yang didaftarkan

	<p>5. Calon tutor klik <i>button</i> cek status</p> <p>6. Sistem menampilkan informasi status pendaftaran</p>
Skenario Alternatif	Jika pada proses pencarian tidak ditemukan data yang cocok atau data salah maka sistem akan menampilkan pesan “NIM tidak terdaftar”. Hal yang harus dilakukan calon tutor yaitu memasukkan kembali NIM dengan benar.
Tujuan	Sistem menampilkan informasi status pendaftaran calon tutor.

j. Nama *Use Case* : Mengelola Materi

Deskripsi *use case* Mengelola materi ditunjukkan pada tabel 3.12 sebagai berikut.

Tabel 3. 12 Deskripsi *Use Case* Mengelola Materi

Nama Use Case	Mengelola Data Materi
Prioritas	Tinggi
Aktor Utama	Tutor
Deskripsi	Aktor dapat melakukan pengelolaan data yaitu dengan melihat, menghapus, menambah, dan mengubah data materi.
Skenario Normal	<ol style="list-style-type: none"> 1. Tutor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel tutor dengan status diterima pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> tutor 4. Tutor memilih menu materi 5. Sistem menampilkan halaman semua daftar materi 6. Jika admin ingin melihat detail materi maka admin klik tombol konten dan sistem akan menampilkan halaman detail dari materi yang dipilih 7. Jika admin ingin menambah data materi maka admin klik <i>button</i> tambah data dan sistem akan menampilkan <i>form</i> penambahan data. Admin mengisi seluruh data yang diperlukan

	<p>lalu klik tambah dan sistem akan menampilkan pesan berhasil menambah data</p> <p>8. Jika admin ingin mengubah data materi maka admin klik tombol <i>edit</i> dan sistem akan menampilkan <i>form</i> perubahan data. Admin mengisi seluruh data yang diubah lalu klik <i>edit</i> dan sistem akan menampilkan pesan berhasil mengubah data</p> <p>9. Jika admin ingin menghapus data materi maka admin klik tombol hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus</p>
Skenario Alternatif	<p>1. Jika pada aliran menambah data materi terjadi <i>exception</i> maka tutor tidak dapat menambah data materi dan tutor dapat mengisi kembali data dengan benar sampai data tersimpan dalam <i>database</i></p> <p>2. Jika pada aliran ubah data materi terjadi <i>exception</i> maka tutor tidak dapat mengubah data materi dan aktor dapat mengisi kembali data dengan benar sehingga data yang tersimpan pada <i>database</i> dapat diperbarui</p>
Tujuan	<p>1. Sistem menampilkan halaman detail materi</p> <p>2. Data hasil dari penambahan dan perubahan tersimpan dalam <i>database</i></p> <p>3. Data yang dihapus hilang dari <i>database</i></p>

k. Nama *Use Case* : Melakukan *Private Chat*

Deskripsi *use case* melakukan *private chat* ditunjukkan pada Tabel 3.13 sebagai berikut.

Tabel 3. 13 Deskripsi *Use Case* Melakukan *Private Chat*

Nama Use Case	Melakukan <i>Private Chat</i>
Prioritas	Tinggi

Aktor Utama	Tutor, Mahasiswa
Deskripsi	Aktor dapat saling mengirim dan menerima pesan dalam sistem.
Skenario Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa pada <i>database</i> 3. Sistem menampilkan halaman utama masing-masing aktor 4. Aktor memilih menu <i>private chat</i> 5. Sistem menampilkan halaman yang berisi daftar mahasiswa/tutor 6. Aktor memilih penerima pesan 7. Aktor menulis pesan 8. Aktor klik <i>button</i> kirim 9. Sistem menampilkan pesan dalam <i>chatbox</i>
Skenario Alternatif	Jika pada proses kirim atau balas pesan terjadi <i>exception</i> maka aktor tidak dapat mengirim balasan pesan dan solusi yang dapat dilakukan yaitu mengirim ulang pesan tersebut.
Tujuan	Penerima dapat melihat pesan di dalam sistem serta pengirim pesan dapat melakukan pengiriman pesan di dalam sistem.

1. Nama *Use Case* : Mengirim Kritik dan Saran

Deskripsi *use case* mengirim kritik dan saran ditunjukkan pada Tabel 3.14 sebagai berikut.

Tabel 3. 14 Deskripsi *Use Case* Kritik dan Saran

Nama <i>Use Case</i>	Mengirim Kritik dan Saran
Prioritas	Sedang
Aktor Utama	Mahasiswa, Tutor
Deskripsi	Kritik dan saran merupakan aktivitas digunakan untuk meningkatkan kinerja aplikasi kedepannya.

Skenario Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke <i>database</i> 3. Sistem menampilkan halaman utama masing-masing aktor 4. Aktor menuju bagian kritik dan saran 5. Sistem menampilkan halaman <i>form</i> kritik dan saran 6. Aktor mengisi kritik dan saran 7. Aktor klik <i>button</i> kirim dan sistem menampilkan pesan berhasil mengirim kritik dan saran
Skenario Alternatif	Jika pada proses pengajuan kritik dan saran terjadi <i>exception</i> gagal untuk mengirim maka sistem akan menampilkan pesan bahwa kritik dan saran gagal untuk dikirim. Maka yang harus dilakukan aktor yaitu mengisi dan mengirim kembali kritik dan saran mereka sampai pengajuan terkirim.
Tujuan	Kritik dan saran yang dikirim akan tersimpan dalam <i>database</i> .

m. Nama *Use Case* : Bertanya pada Forum

Deskripsi *use case* bertanya pada forum ditunjukkan pada Tabel 3.15 sebagai berikut.

Tabel 3. 15 Deskripsi *Use Case* Bertanya pada Forum

Nama Use Case	Bertanya pada Forum
Prioritas	Tinggi
Aktor Utama	Mahasiswa
Deskripsi	Aktor dapat mengajukan pertanyaan pada forum dalam sistem sesuai dengan kategori materi yang dipilih.
Skenario Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel mahasiswa pada <i>database</i>

	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman utama mahasiswa 4. Mahasiswa memilih kategori materi kemudian menu forum 5. Mahasiswa klik <i>button</i> buat forum 6. Sistem menampilkan <i>form</i> pembuatan forum 7. Mahasiswa mengisi <i>form</i> tersebut dengan menuliskan topik dan pertanyaan, kemudian klik <i>button</i> kirim dan sistem akan menampilkan pesan berhasil membuat forum
Skenario Alternatif	Jika pada proses pengajuan pertanyaan pada forum terjadi <i>exception</i> gagal untuk mengirim pertanyaan maka sistem akan menampilkan pesan bahwa pertanyaan gagal untuk dikirim. Maka yang harus dilakukan mahasiswa yaitu mengisi dan mengirim kembali pertanyaan mereka sampai forum berhasil dibuat.
Tujuan	Forum akan tersimpan dalam <i>database</i> dan ditampilkan dalam halaman forum.

n. Nama *Use Case* : Menjawab Forum

Deskripsi *use case* menjawab forum ditunjukkan pada Tabel 3.16 sebagai berikut.

Tabel 3. 16 Deskripsi *Use Case* Menjawab Forum

Nama Use Case	Menjawab Forum
Prioritas	Tinggi
Aktor Utama	Mahasiswa, Tutor
Deskripsi	Aktor dapat menjawab pertanyaan pada forum dalam sistem.
Skenario Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke <i>database</i> 3. Sistem menampilkan halaman utama masing-masing aktor 4. Aktor memilih menu forum 5. Sistem menampilkan halaman forum

	6. Aktor memilih pertanyaan yang akan dijawab 7. Aktor menjawab pertanyaan forum dengan mengisi di kolom balas 8. Aktor klik <i>button</i> kirim
Skenario Alternatif	Jika pada proses balas forum terjadi <i>exception</i> gagal untuk membalas forum maka sistem akan menampilkan pesan bahwa pertanyaan gagal untuk dikirim. Maka yang harus dilakukan aktor yaitu mengisi dan mengirim kembali jawaban mereka sampai jawaban berhasil tampil di halaman detail forum.
Tujuan	Jawaban yang dikirim dalam forum akan tersimpan dalam <i>database</i> dan ditampilkan dalam halaman forum.

o. Nama *Use Case* : Melihat Materi dan Konten

Deskripsi *use case* melihat materi ditunjukkan pada Tabel 3.17 sebagai berikut.

Tabel 3. 17 Deskripsi *Use Case* Melihat Daftar Materi

Nama <i>Use Case</i>	Melihat Materi dan Konten
Prioritas	Tinggi
Aktor Utama	Mahasiswa
Deskripsi	Aktor dapat melihat daftar materi dan konten yang diberikan oleh tutor.
Skenario Normal	1. Mahasiswa melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel mahasiswa pada <i>database</i> 3. Sistem menampilkan halaman utama 4. Mahasiswa memilih kategori materi klik <i>button</i> mulai 5. Sistem menampilkan materi sesuai dengan kategori yang dipilih

	6. Mahasiswa memilih materi yang akan dilihat dan sistem menampilkan detail dari materi tersebut yang berisi daftar konten di setiap materinya
Skenario Alternatif	-
Tujuan	Sistem menampilkan halaman daftar materi sesuai dengan kategori yang dipilih.

p. Nama *Use Case* : Melihat Tutor

Deskripsi *use case* melihat tutor ditunjukkan pada Tabel 3.18 sebagai berikut.

Tabel 3. 18 Deskripsi *Use Case* Melihat Tutor

Nama Use Case	Melihat Tutor
Prioritas	Sedang
Aktor Utama	Mahasiswa
Deskripsi	Mahasiswa dapat melihat semua daftar tutor yang telah terdaftar dalam aplikasi.
Skenario Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel mahasiswa pada <i>database</i> 3. Sistem menampilkan halaman utama 4. Mahasiswa memilih kategori materi klik <i>button</i> mulai 5. Mahasiswa memilih menu tutor 6. Sistem menampilkan semua daftar tutor 7. Mahasiswa memilih tutor yang akan dilihat dan sistem menampilkan detail dari tutor yang dipilih
Skenario Alternatif	-

Tujuan	Sistem menampilkan halaman detail tutor yang telah terdaftar dalam <i>database</i> .
---------------	--

q. Nama *Use Case* : Mengelola Profil

Deskripsi *use case* mengelola profil ditunjukkan pada Tabel 3.19 sebagai berikut.

Tabel 3. 19 Deskripsi *Use Case* Mengubah Profil

Nama Use Case	Mengelola Profil
Prioritas	Sedang
Aktor Utama	Tutor, Mahasiswa
Deskripsi	Proses pengelolaan akun profil yang dimiliki.
Skenario Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke <i>database</i> 3. Sistem menampilkan halaman utama masing-masing aktor 4. Aktor memilih menu profil 5. Sistem menampilkan halaman profil masing-masing aktor 6. Jika aktor ingin mengubah profil maka klik <i>button edit</i> profil dan sistem akan menampilkan halaman perubahan profil. Aktor mengisi form tersebut dan klik <i>edit</i> kemudian sistem akan menampilkan pesan berhasil mengubah data
Skenario Alternatif	Jika pada proses perubahan profil terjadi <i>exception</i> data yang dimasukkan salah atau tidak sesuai dengan ketentuan maka sistem akan menampilkan pesan bahwa proses perubahan profil gagal. Maka hal yang harus dilakukan aktor adalah memasukkan kembali data profil dengan benar dan menyimpannya kembali hingga sistem menampilkan bahwa proses perubahan berhasil.
Tujuan	Sistem menyimpan perubahan data profil dalam <i>database</i> .

r. Nama *Use Case* : Mengumpulkan Tugas

Deskripsi *use case* mengumpulkan tugas ditunjukkan pada Tabel 3.20 sebagai berikut

Tabel 3. 20 Deskripsi *Use Case* Mengumpulkan Tugas

Nama <i>Use Case</i>	Mengumpulkan Tugas
Prioritas	Tinggi
Aktor Utama	Mahasiswa
Deskripsi	Mahasiswa mengumpulkan tugas yang telah diberikan tutor di setiap konten yang diberikan.
Skenario Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel mahasiswa pada <i>database</i> 3. Sistem menampilkan halaman utama 4. Mahasiswa memilih kategori materi dan klik <i>button</i> mulai 5. Sistem menampilkan daftar materi sesuai dengan kategori 6. Mahasiswa memilih menu lihat materi dan menampilkan daftar konten di setiap materi 7. Mahasiswa memilih konten 8. Mahasiswa klik <i>button</i> kumpulkan tugas dan sistem menampilkan halaman <i>form</i> pengumpulan tugas 9. Mahasiswa mengisi form tersebut dan klik <i>submit</i> kemudian sistem akan menampilkan pesan tugas berhasil dikirim
Skenario Alternatif	Jika pada proses mengumpulkan tugas terjadi <i>exception</i> gagal dalam pengiriman tugas maka sistem akan menampilkan pesan gagal untuk dikirim. Maka yang harus dilakukan mahasiswa yaitu mengisi dan mengirim kembali tugas mereka sampai data terkirim.
Tujuan	Tugas yang dikirim tersimpan dalam <i>database</i> dengan status “diajukan”.

s. Nama *Use Case* : Memverifikasi Tugas

Deskripsi *use case* memverifikasi tugas ditunjukkan pada Tabel 3.21 sebagai berikut

Tabel 3. 21 Deskripsi *Use Case* Memverifikasi Tugas

Nama Use Case	Memverifikasi Tugas
Prioritas	Tinggi
Aktor Utama	Tutor
Deskripsi	Aktor dapat menyetujui atau menolak tugas yang dikumpulkan oleh mahasiswa pada halaman pengumpulan tugas.
Skenario Normal	<ol style="list-style-type: none"> 1. Tutor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel tutor pada <i>database</i> 3. Sistem menampilkan halaman utama tutor 4. Tutor memilih menu pengumpulan tugas 5. Sistem menampilkan halaman yang berisi daftar tugas yang telah dikumpulkan oleh mahasiswa 6. Tutor klik <i>link</i> Github dan memeriksa tugas tersebut 7. Jika tutor memilih untuk menyetujui tugas maka klik tombol verifikasi dan sistem akan menampilkan pesan berhasil memverifikasi tugas, sedangkan jika tutor ingin memberikan revisi maka tutor klik tombol revisi
Skenario Alternatif	-
Tujuan	<ol style="list-style-type: none"> 1. Jika tutor menyetujui tugas yang dikumpulkan maka status tugas pada <i>database</i> akan berganti menjadi “disetujui” 2. Jika tutor merevisi tugas yang dikumpulkan maka status tugas pada <i>database</i> akan menjadi yaitu “revisi”

t. Nama *Use Case* : Mengelola Konten

Deskripsi *use case* mengelola konten ditunjukkan pada Tabel 3.22 sebagai berikut.

Tabel 3. 22 Deskripsi *Use Case* Mengelola Konten

Nama <i>Use Case</i>	Mengelola Konten
Prioritas	Tinggi
Aktor Utama	Tutor
Deskripsi	Aktor dapat melakukan pengelolaan data yaitu dengan melihat, menghapus, menambah, dan mengubah konten di setiap materi.
Skenario Normal	<ol style="list-style-type: none"> 1. Tutor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel tutor dengan status diterima pada <i>database</i> 3. Sistem menampilkan halaman <i>dashboard</i> tutor 4. Tutor memilih menu materi 5. Sistem menampilkan halaman semua daftar materi 6. Admin klik tombol konten dan sistem menampilkan semua daftar konten yang ada pada materi yang dipilih 7. Jika admin ingin menambah konten maka tutor klik <i>button</i> tambah konten dan sistem akan menampilkan <i>form</i> penambahan data. Admin mengisi seluruh data yang diperlukan lalu klik tambah dan sistem akan menampilkan pesan berhasil menambah data 8. Jika tutor ingin mengubah konten maka tutor klik ikon pensil dan sistem akan menampilkan <i>form</i> perubahan data. Tutor mengisi seluruh data yang diubah lalu klik <i>edit</i> dan sistem akan menampilkan pesan berhasil mengubah data 9. Jika tutor ingin menghapus konten maka tutor klik ikon hapus dan sistem akan menampilkan peringatan apakah data yang dipilih benar-benar dihapus. Jika “OK” maka sistem

	menghapus data dari <i>database</i> dan menampilkan pesan berhasil menghapus
Skenario Alternatif	<ol style="list-style-type: none"> 1. Jika pada aliran menambah konten terjadi <i>exception</i> maka tutor tidak dapat menambah konten dan tutor dapat mengisi kembali data dengan benar sampai data tersimpan dalam <i>database</i> 2. Jika pada aliran ubah konten terjadi <i>exception</i> maka tutor tidak dapat mengubah konten dan aktor dapat mengisi kembali data dengan benar sehingga data yang tersimpan pada <i>database</i> dapat diperbarui
Tujuan	<ol style="list-style-type: none"> 1. Sistem menampilkan detail konten 2. Data hasil dari penambahan dan perubahan tersimpan dalam <i>database</i> 3. Data yang dihapus hilang dari <i>database</i>

u. *Use Case* Memberi Revisi Tugas

Deskripsi *use case* memberi revisi tugas ditunjukkan pada Tabel 3.23 sebagai berikut.

Tabel 3. 23 Deskripsi *Use Case* Memberi Revisi Tugas

Nama Use Case	Memberi Revisi Tugas
Prioritas	Tinggi
Aktor Utama	Tutor
Deskripsi	Aktor memberikan revisi pada tugas mahasiswa yang masih kurang tepat.
Skenario Normal	<ol style="list-style-type: none"> 1. Tutor melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> 2. Sistem memeriksa valid tidaknya data yang dimasukkan dengan memeriksa ke tabel tutor pada <i>database</i> 3. Sistem menampilkan halaman utama tutor 4. Tutor memilih menu tugas mahasiswa 5. Sistem menampilkan halaman yang berisi daftar tugas

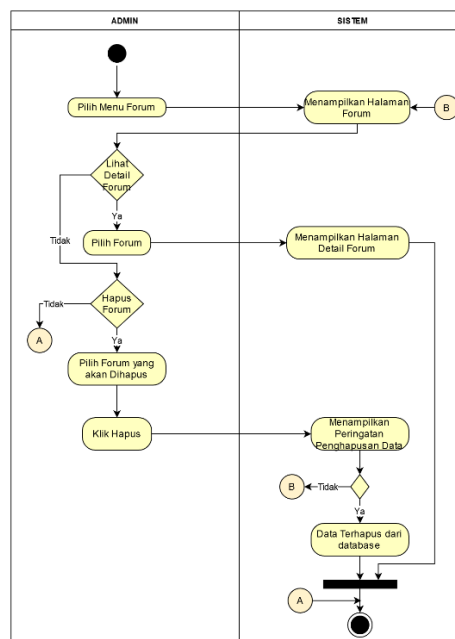
	<p>6. Tutor klik ikon revisi dan sistem akan menampilkan halaman <i>form</i> revisi</p> <p>7. Tutor mengisi revisi dan klik button kirim dan sistem akan menampilkan pesan revisi berhasil dikirim</p>
Skenario Alternatif	Jika pada aliran menambah revisi terjadi <i>exception</i> maka tutor tidak dapat menambah revisi dan tutor dapat mengisi kembali data dengan benar sampai data tersimpan dalam <i>database</i> .
Tujuan	Revisi tersimpan dalam <i>database</i> dan status tugas menjadi “revisi”.

3.3.3 Activity Diagram

Activity diagram adalah sebuah diagram yang menjelaskan sebuah alur kerja atau kegiatan di dalam program yang dibangun. *Activity diagram* juga menggambarkan bagaimana alur sistem berawal, pilihan (*decision*) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut. Berikut ini *activity diagram* pada aplikasi yang dibangun.

1. Activity Diagram Mengelola Forum

Activity diagram mengelola forum pada Gambar 3.7 menggambarkan semua alur proses pengelolaan forum yang dilakukan oleh admin.



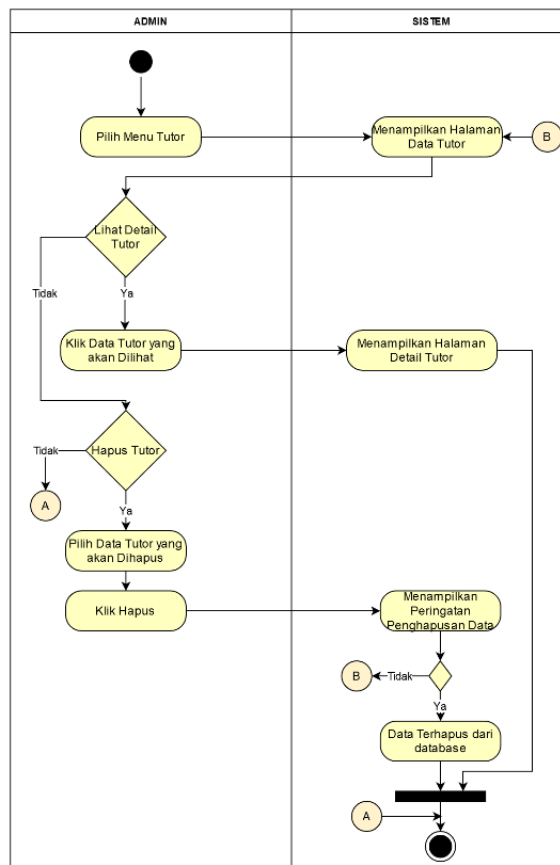
Gambar 3. 7 Activity Diagram Mengelola Forum

Berdasarkan Gambar 3.7 dapat dijelaskan bahwa alur proses untuk aktivitas pengelolaan forum yaitu:

- 1) Admin memilih menu forum kemudian sistem menampilkan halaman forum.
- 2) Setelah itu terdapat dua kondisi yaitu melihat atau menghapus forum.
- 3) Jika admin ingin melihat forum maka admin memilih forum yang akan dilihat kemudian sistem menampilkan detail forum.
- 4) Sedangkan jika admin akan menghapus forum, maka admin memilih forum mana yang akan dihapus kemudian sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut kemudian sistem akan menghapus data dari *database*.

2. Activity Diagram Mengelola Data Tutor

Activity diagram mengelola data tutor pada Gambar 3.8 menggambarkan semua alur proses pengelolaan data tutor yang dilakukan oleh admin.



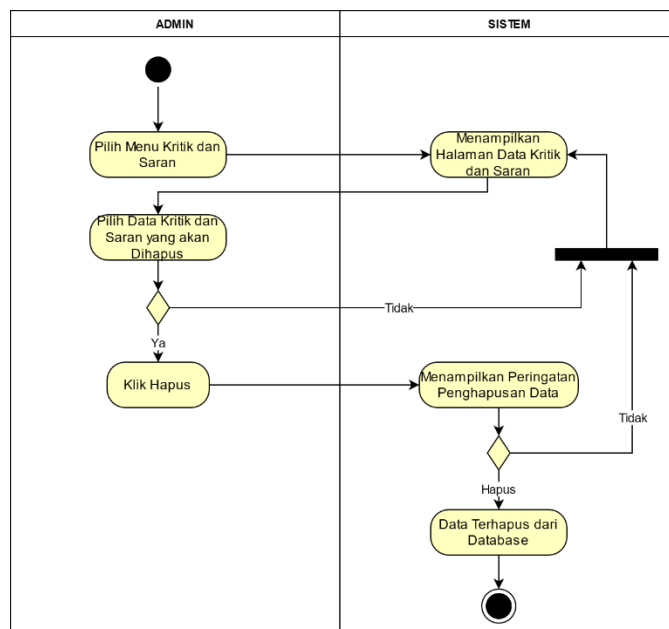
Gambar 3. 8 Activity Diagram Mengelola Data Tutor

Berdasarkan Gambar 3.8 dapat dijelaskan bahwa alur proses untuk aktivitas pengelolaan data tutor yaitu:

- 1) Admin memilih menu tutor.
- 2) Sistem menampilkan halaman data tutor.
- 3) Setelah itu terdapat dua kondisi yaitu melihat atau menghapus data tutor.
- 4) Jika admin ingin melihat data tutor maka admin memilih data tutor mana yang akan dilihat kemudian sistem menampilkan detail informasi dari tutor tersebut.
- 5) Sedangkan jika admin akan menghapus data tutor, maka admin memilih data tutor mana yang akan dihapus kemudian sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut kemudian sistem akan menghapus data dari *database*.

3. Activity Diagram Mengelola Kritik dan Saran

Activity diagram mengelola kritik dan saran pada Gambar 3.9 menggambarkan semua alur proses untuk mengelola kritik dan saran yang dilakukan oleh admin.



Gambar 3. 9 Activity Diagram Mengelola Kritik dan Saran

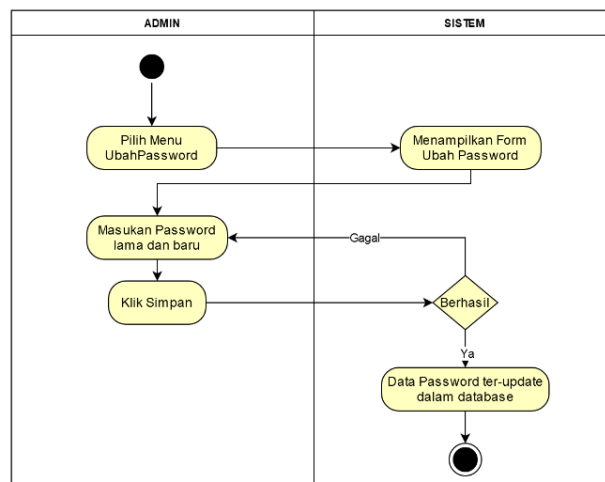
Berdasarkan Gambar 3.9 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengelola kritik dan saran yaitu:

- 1) Admin memilih menu kritik dan saran.

- 2) Sistem akan menampilkan halaman data kritik dan saran yang telah dikirim oleh tutor dan mahasiswa.
- 3) Jika admin memilih untuk menghapus data kritik dan saran maka admin memilih data mana yang akan dihapus lalu sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut, kemudian sistem akan menghapus data dari *database*.

4. Activity Diagram Mengubah Password

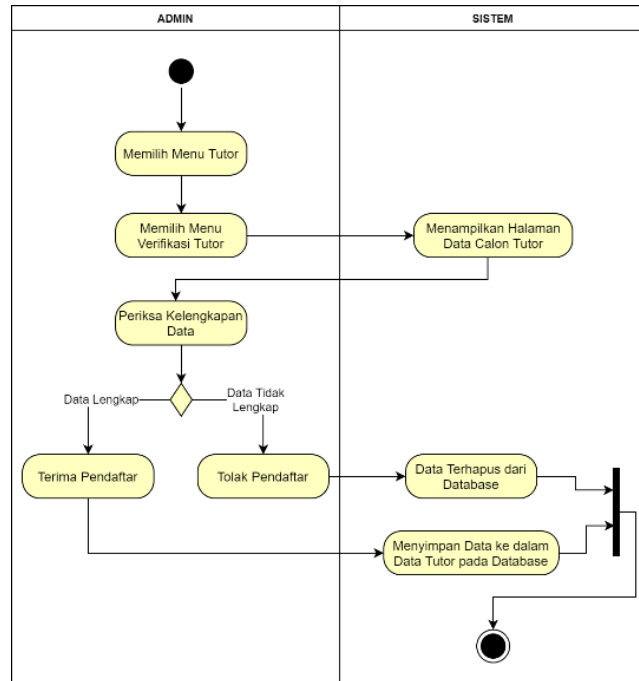
Activity diagram mengubah password pada Gambar 3.10 menggambarkan semua alur proses untuk mengubah password yang dilakukan oleh admin.



Gambar 3. 10 Activity Diagram Mengubah Password

Berdasarkan Gambar 3.10 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengubah password yaitu:

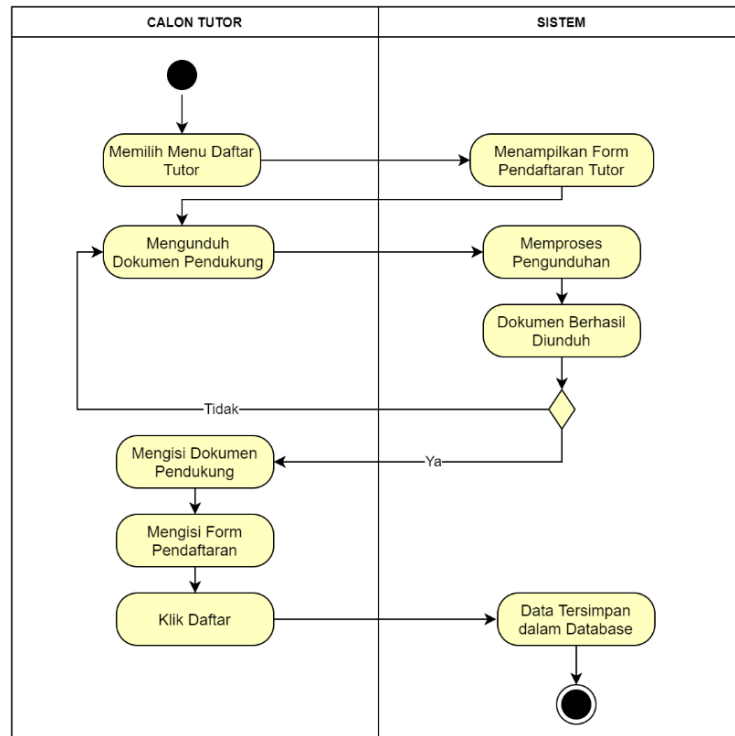
- 1) Admin memilih menu ubah password.
 - 2) Sistem akan menampilkan form perubahan password.
 - 3) Kemudian admin memperbarui password dengan memasukkan password baru dan admin klik button simpan.
 - 4) Sistem akan menyimpan perubahan data password dalam database.
- #### 5. Activity Diagram Memverifikasi Calon Tutor
- Activity diagram memverifikasi calon tutor pada Gambar 3.11 menggambarkan semua alur proses dalam memverifikasi calon tutor yang dilakukan oleh admin.



Gambar 3. 11 *Activity Diagram* Memverifikasi Calon Tutor

Berdasarkan Gambar 3.11 dapat dijelaskan bahwa alur proses untuk aktivitas dalam memverifikasi calon tutor yaitu:

- 1) Admin memilih menu tutor.
 - 2) Sistem akan menampilkan halaman verifikasi data calon tutor.
 - 3) Admin memilih data pendaftar mana yang akan dilihat dan melakukan pemeriksaan kelengkapan data yang diunggah oleh calon tutor.
 - 4) Apabila data lengkap maka admin berhak untuk menerima pendaftar tersebut dengan klik *button* terima dan sistem akan menyimpan data pendaftar sebagai data tutor dalam *database*.
 - 5) Sedangkan jika data tidak lengkap maka admin berhak untuk melakukan penolakan dengan klik *button* tolak sehingga sistem akan menghapus data pendaftar dari *database*.
6. *Activity Diagram* Mendaftar Tutor
- Activity diagram* mendaftar tutor pada Gambar 3.12 menggambarkan semua alur proses dalam mendaftar sebagai tutor yang dilakukan oleh calon tutor.



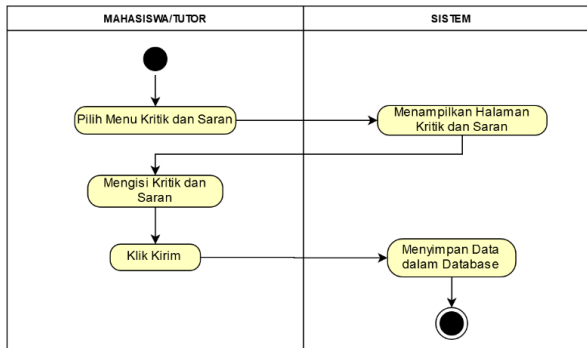
Gambar 3. 12 Activity Diagram Mendaftar Tutor

Berdasarkan Gambar 3.12 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mendaftar tutor yaitu:

- 1) Calon tutor memilih menu daftar tutor.
- 2) Sistem akan menampilkan *form* pendaftaran tutor.
- 3) Sebelum mengisi *form*, calon tutor harus mengunduh dokumen pendukung sebagai syarat dalam pendaftaran sebagai tutor dalam aplikasi.
- 4) Setelah mengunduh dokumen calon tutor mengisi dokumen tersebut.
- 5) Selanjutnya calon tutor memasukkan data yang diminta pada *form* pendaftaran termasuk mengunggah dokumen tersebut dan klik *button* daftar, data pendaftaran calon tutor akan tersimpan dalam *database*.

7. Activity Diagram Mengirim Kritik dan Saran

Activity diagram mengirim kritik dan saran pada Gambar 3.13 menggambarkan semua alur proses dalam mengirim kritik dan saran yang dilakukan oleh tutor dan mahasiswa.



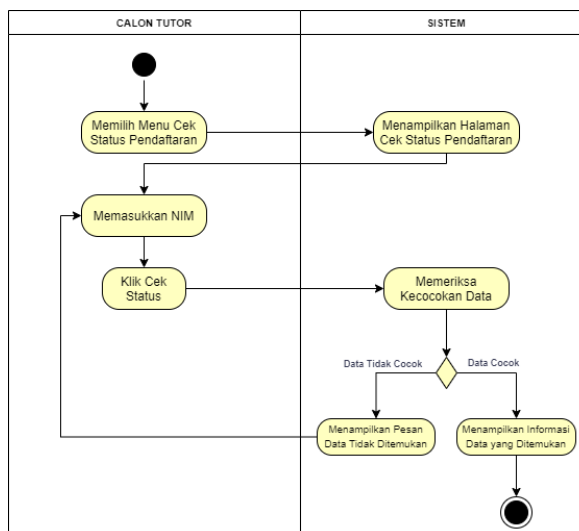
Gambar 3. 13 *Activity Diagram* Mengirim Kritik dan Saran

Berdasarkan Gambar 3.13 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengirim kritik dan saran yaitu:

- 1) Mahasiswa dan tutor memilih menu kritik dan saran.
- 2) Kemudian sistem menampilkan halaman *form* kritik dan saran.
- 3) Tutor dan mahasiswa mengisi kritik dan saran.
- 4) Tutor dan mahasiswa klik *button* kirim dan sistem menyimpan data dalam *database*.

8. *Activity Diagram* Memeriksa Status Pendaftaran

Activity diagram memeriksa status pendaftaran pada Gambar 3.14 menggambarkan semua alur proses dalam memeriksa status pendaftaran yang dilakukan oleh calon tutor.



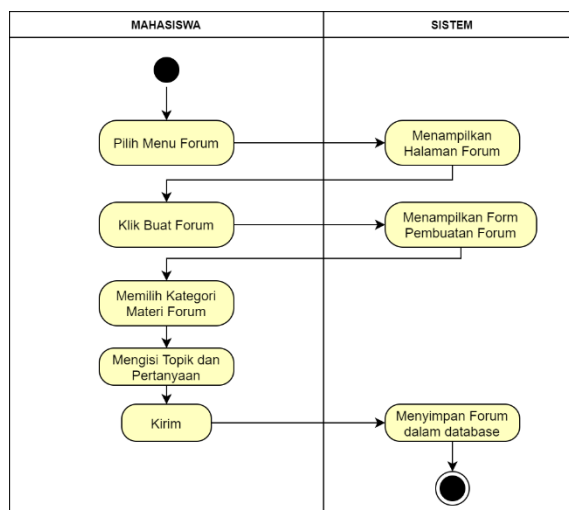
Gambar 3. 14 *Activity Diagram* Memeriksa Status Pendaftaran

Berdasarkan Gambar 3.14 dapat dijelaskan bahwa alur proses untuk aktivitas dalam memeriksa status pendaftaran yaitu:

- 1) Calon tutor memilih menu cek status pendaftaran.
- 2) Sistem akan menampilkan halaman cek status pendaftaran.
- 3) Pada halaman tersebut calon tutor memasukkan NIM yang didaftarkan saat melakukan pendaftaran kemudian klik *button* cek status.
- 4) Selanjutnya sistem akan memeriksa kecocokan data ke dalam *database*, jika data yang dimasukkan oleh calon tutor cocok maka sistem akan menampilkan informasi dari data yang dimasukkan sedangkan jika data tidak cocok maka sistem akan menampilkan pesan bahwa data yang dicari tidak ditemukan.

9. *Activity Diagram* Bertanya pada Forum

Activity diagram bertanya pada forum pada Gambar 3.15 menggambarkan semua alur proses dalam mengajukan pertanyaan pada forum yang dilakukan oleh mahasiswa.



Gambar 3. 15 *Activity Diagram* Bertanya pada Forum

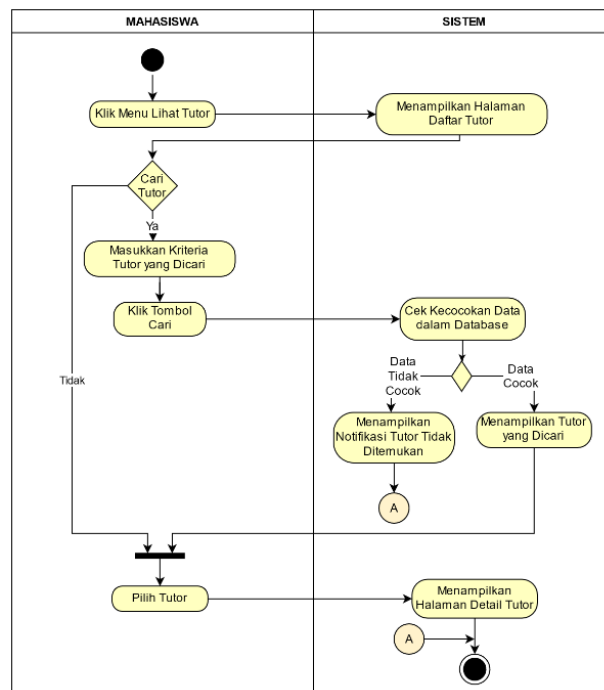
Berdasarkan Gambar 3.15 dapat dijelaskan bahwa alur proses untuk aktivitas dalam bertanya pada forum yaitu:

- 1) Mahasiswa memilih menu forum.
- 2) Sistem menampilkan halaman forum.
- 3) Kemudian mahasiswa klik *button* buat forum dan sistem menampilkan *form* pembuatan forum.
- 4) Di dalam *form* tersebut mahasiswa memilih kategori materi pada forum sesuai dengan topik dan pertanyaan yang akan diajukan.

- 5) Mahasiswa mengisi pertanyaan pada *form* dengan menjelaskan detail maksud dari topik yang telah dimasukkan.
- 6) Mahasiswa klik *button* kirim.
- 7) Selanjutnya sistem akan menyimpan forum dalam *database*.

10. Activity Diagram Melihat Tutor

Activity diagram melihat tutor pada Gambar 3.16 menggambarkan semua alur proses dalam melihat tutor yang dilakukan oleh mahasiswa.



Gambar 3. 16 *Activity Diagram* Melihat Tutor

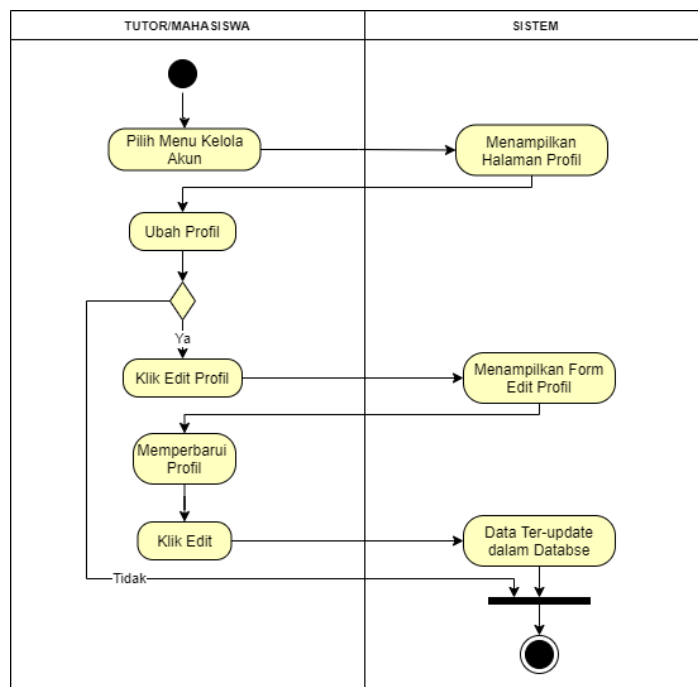
Berdasarkan Gambar 3.16 dapat dijelaskan bahwa alur proses untuk aktivitas dalam melihat tutor yaitu:

- 1) Mahasiswa memilih menu tutor.
- 2) Sistem akan menampilkan halaman daftar tutor secara keseluruhan.
- 3) Terdapat kondisi apakah mahasiswa ingin melakukan proses pencarian tutor atau langsung memilih tutor yang akan dilihat. Jika “Ya” maka mahasiswa memasukkan kriteria nama atau kategori tutor yang dicari kemudian klik *button* cari. Sistem akan melakukan pemeriksaan kecocokan data pada *database*.

- 4) Jika data yang dimasukkan cocok dalam *database* maka sistem akan menampilkan data tutor yang dicari sedangkan jika data yang dimasukkan tidak cocok dalam *database* maka sistem akan menampilkan pesan bahwa data tutor tidak ditemukan.
- 5) Selanjutnya mahasiswa memilih tutor yang akan dilihat dan sistem akan menampilkan informasi detail dari tutor tersebut.

11. Activity Diagram Mengelola Profil

Activity diagram mengelola tutor pada Gambar 3.17 menggambarkan semua alur proses dalam mengelola profil yang dilakukan oleh tutor atau mahasiswa.



Gambar 3. 17 Activity Diagram Mengelola Profil

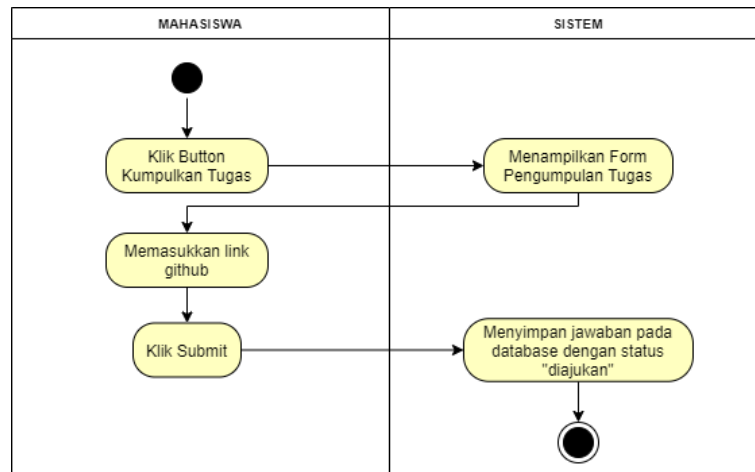
Berdasarkan Gambar 3.17 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengelola profil yaitu:

- 1) Mahasiswa memilih menu kelola akun.
- 2) Sistem menampilkan halaman profil berupa detail data diri.
- 3) Jika tutor atau mahasiswa ingin melakukan ubah profil maka klik ikon *edit* profil dan sistem akan menampilkan *form* perubahan profil dan tutor atau mahasiswa dapat melakukan perubahan data.

- 4) Kemudian tutor atau mahasiswa menyimpan perubahan tersebut dengan klik *button edit*.
- 5) Apabila tutor atau mahasiswa tidak ingin melakukan ubah data maka sistem akan berakhir.

12. Activity Diagram Mengumpulkan Tugas

Activity diagram mengumpulkan tugas pada Gambar 3.18 menggambarkan semua alur proses dalam pengumpulan tugas yang dilakukan oleh mahasiswa.



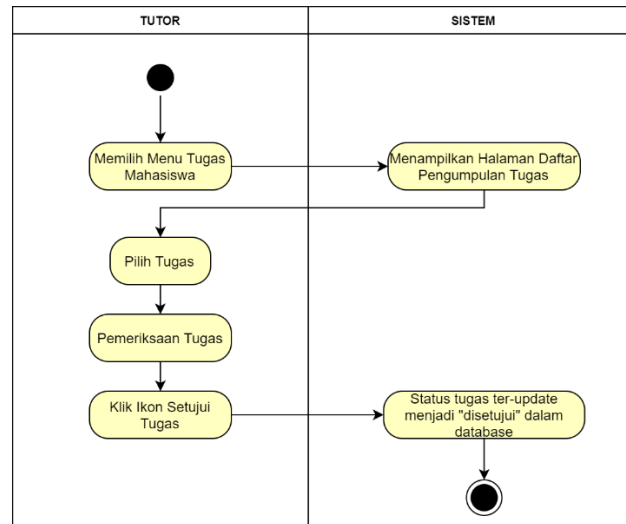
Gambar 3. 18 Activity Diagram Mengumpulkan Tugas

Berdasarkan Gambar 3.18 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengumpulkan tugas yaitu:

- 1) Mahasiswa yang awalnya masuk ke dalam menu materi yang dipilih terlebih dahulu seperti pada Gambar 3.21. Di dalam setiap materi tersebut terdapat tugas yang harus dikerjakan oleh mahasiswa.
- 2) Setelah pengerjaan, proses yang dilakukan mahasiswa dalam mengumpulkan tugas yaitu mahasiswa klik kumpulan tugas.
- 3) Sistem akan menampilkan halaman *form* pengumpulan tugas.
- 4) Kemudian mahasiswa mengumpulkan alamat URL Github di mana di dalam pada alamat tersebut berisi tugas yang telah dikerjakan oleh mahasiswa.
- 5) Setelah itu mahasiswa klik *button submit*.
- 6) Sistem akan menyimpan jawaban tugas mahasiswa ke dalam *database* dengan status "Diajukan" yang akan ditampilkan pada halaman tutor.

13. Activity Diagram Memverifikasi Tugas

Activity diagram memverifikasi tugas pada Gambar 3.19 menggambarkan semua alur proses dalam verifikasi tugas yang dilakukan oleh tutor.



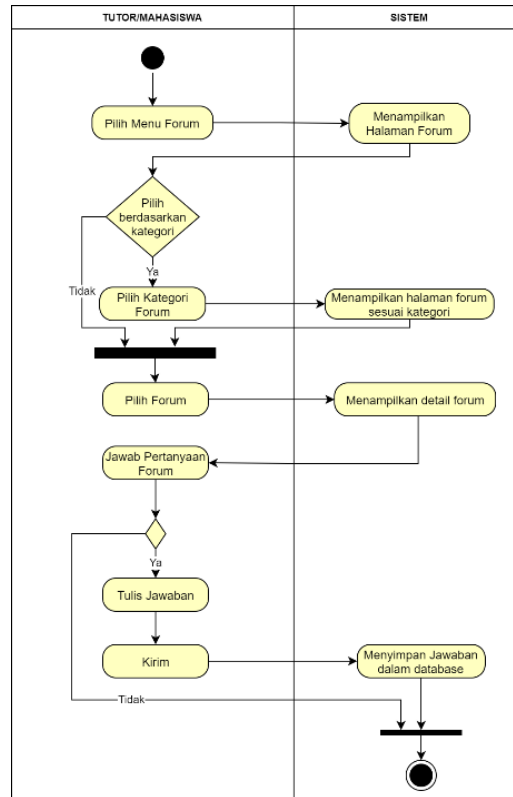
Gambar 3. 19 Activity Diagram Memverifikasi Tugas

Berdasarkan Gambar 3.19 dapat dijelaskan bahwa alur proses untuk aktivitas dalam memverifikasi tugas yang telah dikumpulkan oleh mahasiswa yaitu:

- 1) Tutor memilih menu pengumpulan tugas.
- 2) Sistem akan menampilkan halaman daftar pengumpulan tugas.
- 3) Tutor memilih tugas mana yang akan diperiksa.
- 4) Apabila jawaban telah sesuai maka tutor berhak untuk menyetujui tugas tersebut dengan klik ikon ceklis dan sistem akan meng-*update* status tugas menjadi "Disetujui" dalam *database*.

14. Activity Diagram Menjawab Forum

Activity diagram menjawab forum pada Gambar 3.20 menggambarkan semua alur proses dalam menjawab forum yang dilakukan oleh tutor dan mahasiswa.



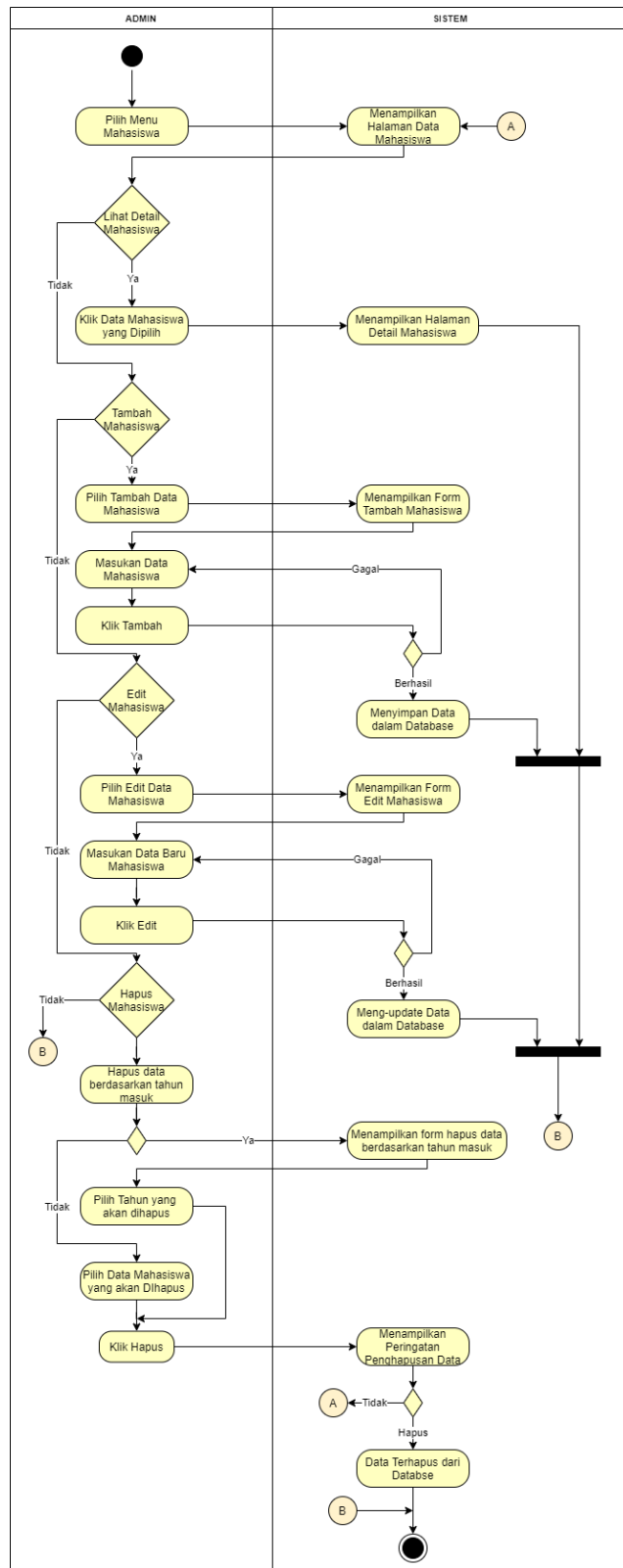
Gambar 3. 20 Activity Diagram Menjawab Forum

Berdasarkan Gambar 3.20 dapat dijelaskan bahwa alur proses untuk aktivitas dalam menjawab forum yaitu:

- 1) Tutor atau mahasiswa memilih menu forum.
- 2) Sistem menampilkan halaman forum.
- 3) Kemudian terdapat kondisi apakah tutor atau mahasiswa ingin mencari forum sesuai dengan kategori dan kata kunci atau langsung memilih forum yang tersedia.
- 4) Tutor atau mahasiswa klik forum yang dipilih dan sistem menampilkan detail forum.
- 5) Setelah itu tutor atau mahasiswa menuliskan jawaban mereka dalam kolom balas forum.
- 6) Tutor atau mahasiswa klik *button* kirim.
- 7) Selanjutnya sistem akan menyimpan jawaban dalam *database*.

15. Activity Diagram Mengelola Data Mahasiswa

Activity diagram mengelola data mahasiswa pada Gambar 3.21 menggambarkan semua alur proses pengelolaan data mahasiswa yang dilakukan oleh admin.



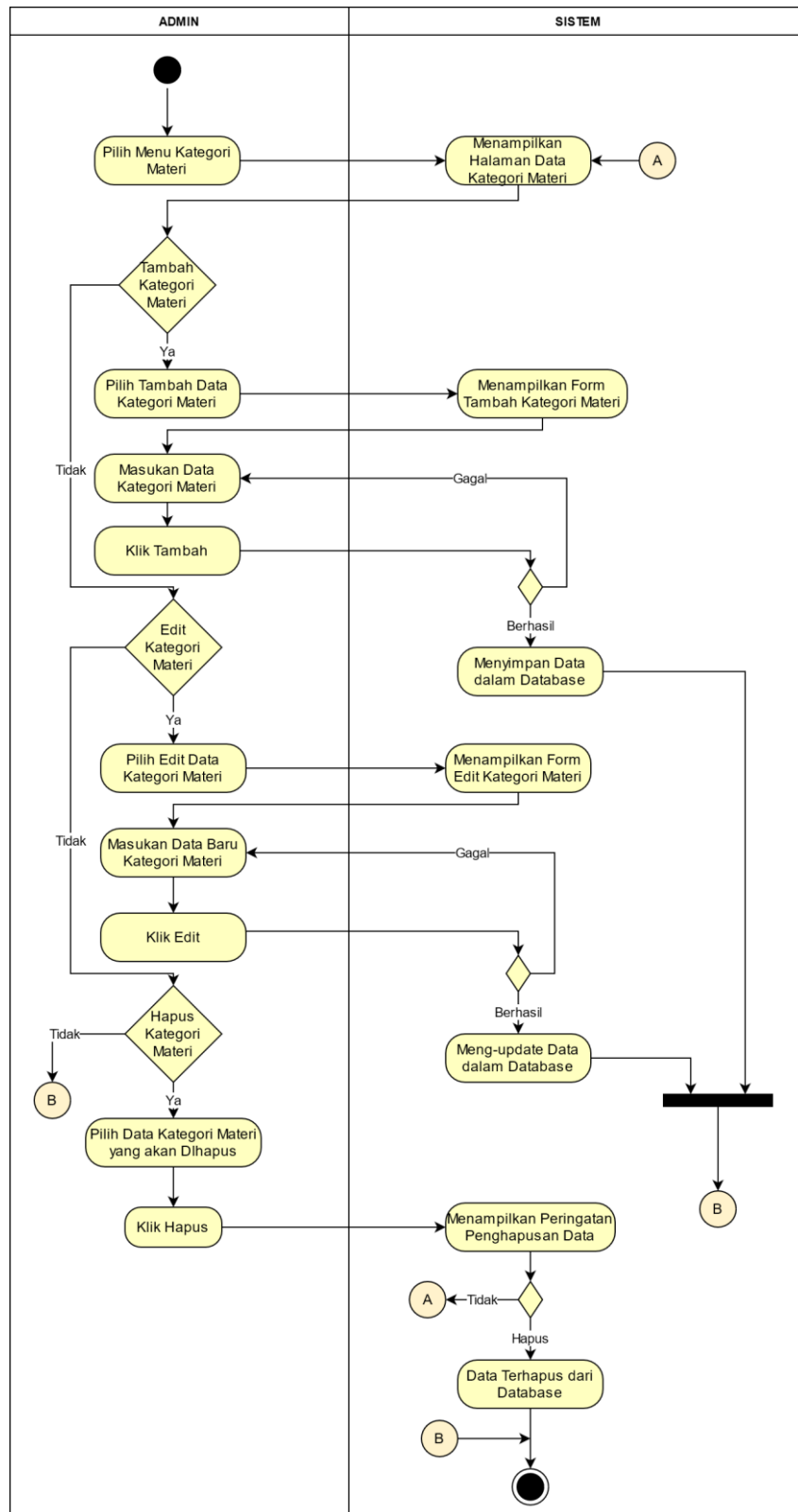
Gambar 3. 21 Activity Diagram Mengelola Data Mahasiswa

Berdasarkan Gambar 3.21 dapat dijelaskan bahwa alur proses untuk aktivitas pengelolaan data mahasiswa yaitu:

- 1) Admin memilih menu mahasiswa.
- 2) Sistem menampilkan halaman data mahasiswa. Di dalam halaman data mahasiswa terdapat empat kondisi yaitu admin dapat melihat detail mahasiswa, menambah, data, dan menghapus data mahasiswa. Berikut ini merupakan penjelasan masing-masing kondisi yaitu:
 - a. Jika admin memilih untuk melihat detail mahasiswa maka admin memilih data mahasiswa yang akan dilihat kemudian sistem akan menampilkan detail dari data mahasiswa tersebut.
 - b. Jika admin memilih untuk menambah data mahasiswa maka admin memilih *button* tambah data atau *button import* data dan sistem akan menampilkan *form* penambahan/*import* data. Selanjutnya admin memasukkan/meng-*import* data mahasiswa setelah itu klik *button* tambah/*import*. Kemudian sistem akan menyimpan data tersebut ke dalam *database*.
 - c. Jika admin memilih untuk mengubah data mahasiswa maka admin memilih data mahasiswa mana yang akan diubah setelah itu klik *button* ubah data. Sistem akan menampilkan *form* pembaruan data mahasiswa. Kemudian admin mengubah data mahasiswa pada *form* tersebut lalu klik *edit*. Sistem akan menyimpan pembaruan data tersebut dalam *database*.
 - d. Jika admin memilih untuk menghapus data mahasiswa maka admin dapat memilih untuk menghapus data berdasarkan tahun masuk dengan memasukkan tahun mana yang akan dihapus atau memilih data mahasiswa mana yang akan dihapus. Lalu sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut. Ketika diklik hapus maka sistem akan menghapus data dari *database*.

16. *Activity Diagram* Mengelola Data Kategori Materi

Activity diagram mengelola data kategori materi pada Gambar 3.22 menggambarkan semua alur proses pengelolaan data kategori materi yang dilakukan oleh admin.



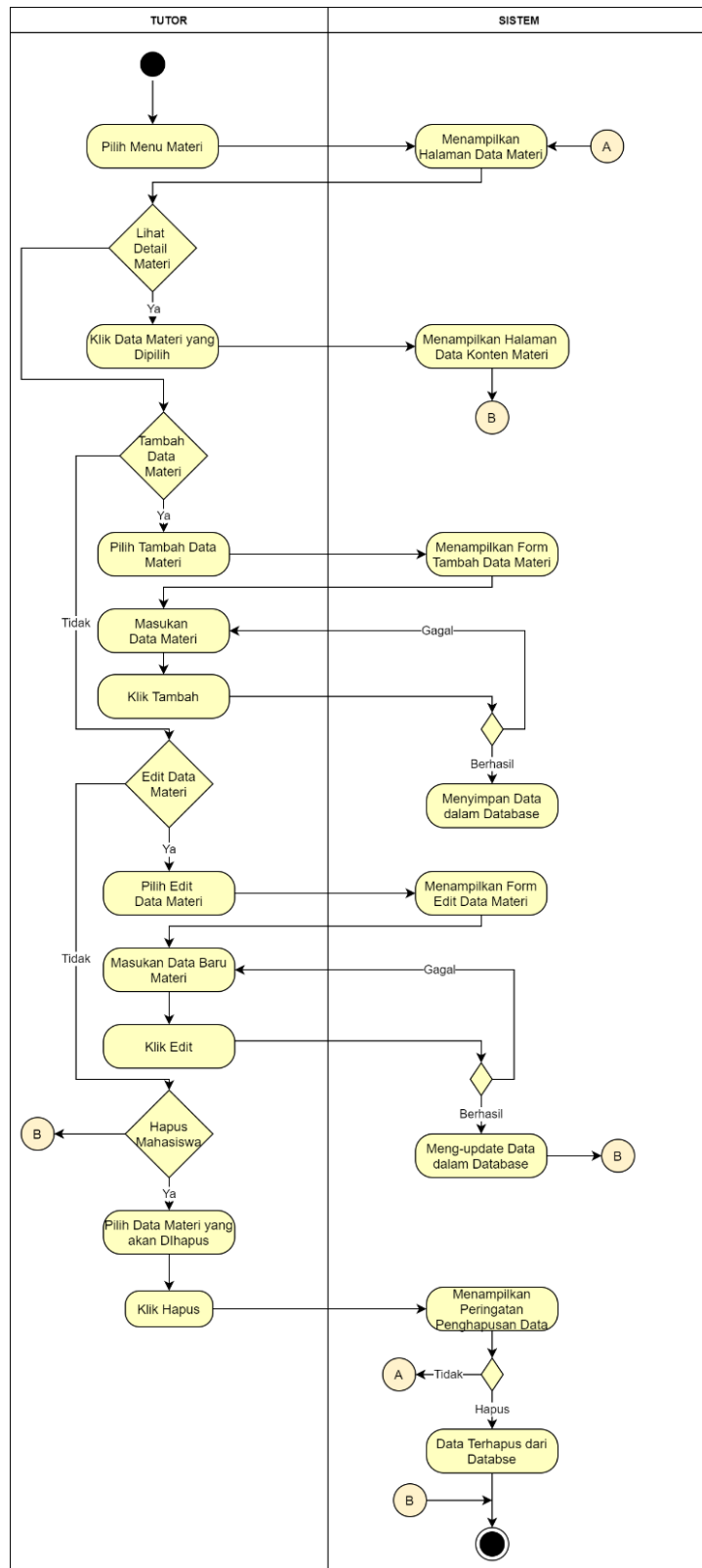
Gambar 3. 22 Activity Diagram Mengelola Data Kategori Materi

Berdasarkan Gambar 3.22 dapat dijelaskan bahwa alur proses untuk aktivitas pengelolaan data kategori materi yaitu:

- 1) Admin memilih menu kategori materi.
- 2) Sistem akan menampilkan halaman data kategori materi. Di dalam halaman data kategori materi terdapat empat kondisi yaitu admin dapat melihat, menambah, mengubah, dan menghapus data kategori materi. Berikut ini merupakan penjelasan masing-masing kondisi yaitu:
 - a. Jika admin memilih untuk menambah data kategori materi maka admin memilih *button* tambah data dan sistem akan menampilkan *form* penambahan data. Selanjutnya admin memasukkan data kategori materi setelah itu klik *button* tambah. Kemudian sistem akan menyimpan data tersebut ke dalam *database*.
 - b. Jika admin memilih untuk mengubah data kategori materi maka admin memilih data kategori materi mana yang akan diubah setelah itu klik ikon pensil. Sistem akan menampilkan *form* pengubahan data kategori materi. Kemudian admin mengubah data kategori materi pada *form* tersebut lalu klik *edit*. Sistem akan menyimpan pembaruan data tersebut dalam *database*.
 - c. Jika admin memilih untuk menghapus data kategori materi maka admin memilih data kategori materi mana yang akan dihapus. Ketika klik ikon hapus lalu sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut, kemudian sistem akan menghapus data dari *database*.

17. Activity Diagram Mengelola Materi

Activity diagram mengelola materi pada Gambar 3.23 menggambarkan semua alur proses dalam mengelola materi yang dilakukan oleh tutor.



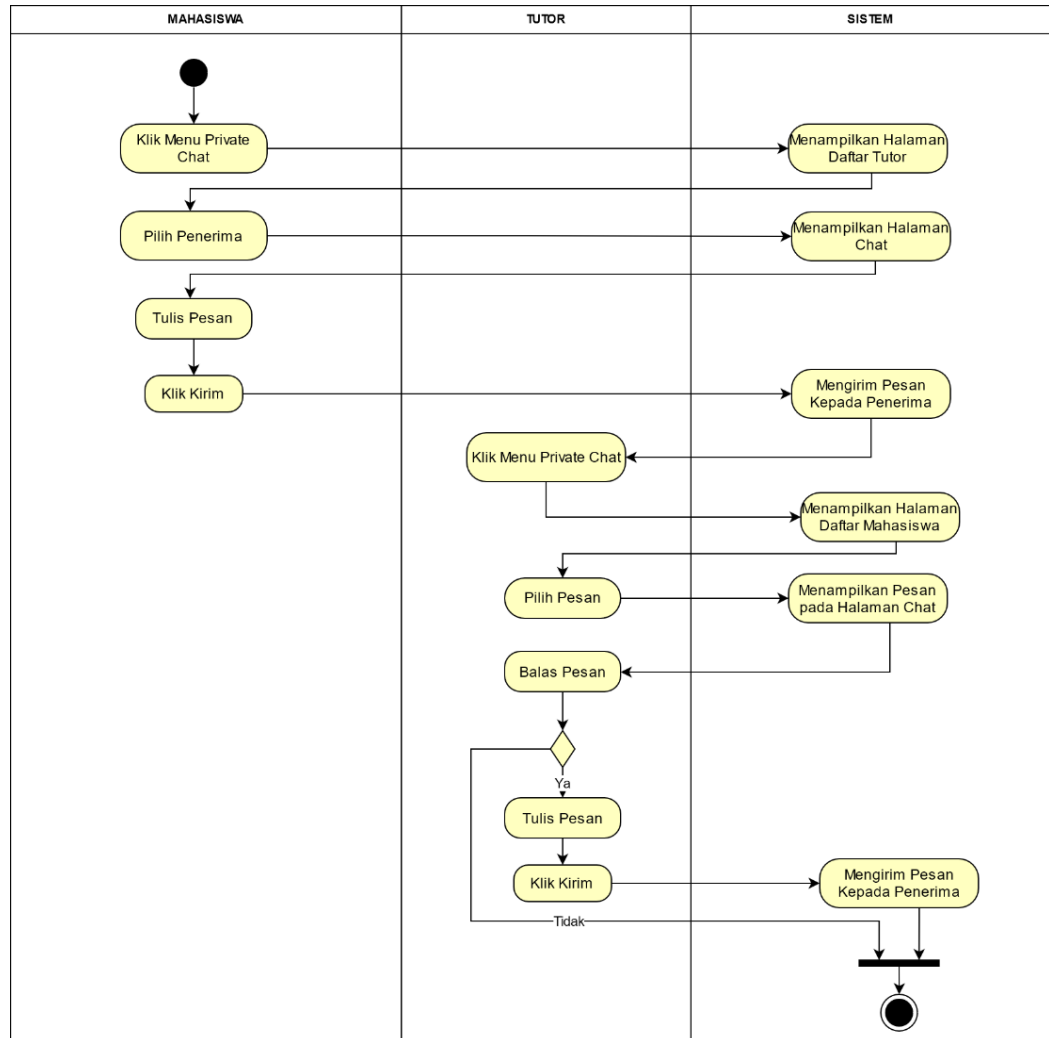
Gambar 3. 23 Activity Diagram Mengelola Materi

Berdasarkan Gambar 3.23 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengelola materi yaitu:

- 1) Tutor memilih menu materi.
- 2) Sistem akan menampilkan halaman materi. Di dalam halaman materi terdapat empat kondisi yaitu admin dapat melihat, menambah, mengubah, dan menghapus data materi. Berikut ini merupakan penjelasan masing-masing kondisi yaitu:
 - a. Jika tutor memilih untuk melihat detail materi maka tutor memilih data materi yang akan dilihat kemudian sistem akan menampilkan detail dari data materi tersebut.
 - b. Jika tutor memilih untuk menambah data materi maka tutor memilih *button* tambah data dan sistem akan menampilkan *form* penambahan data. Selanjutnya tutor memasukkan data materi setelah itu klik *button* tambah. Kemudian sistem akan menyimpan data tersebut ke dalam *database*.
 - c. Jika tutor memilih untuk mengubah data materi maka tutor memilih data materi mana yang akan diubah setelah itu klik ikon pensil data. Sistem akan menampilkan *form* pembaruan data materi. Kemudian admin memperbarui data materi pada *form* tersebut lalu klik *edit*. Sistem akan menyimpan pembaruan data tersebut dalam *database*.
 - d. Jika tutor memilih untuk menghapus data materi maka tutor memilih data materi mana yang akan dihapus. Ketika klik ikon hapus lalu sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut, kemudian sistem akan menghapus data dari *database*.

18. *Activity Diagram* Melakukan *Private Chat*

Activity diagram melakukan *private chat* pada Gambar 3.24 menggambarkan semua alur proses dalam melakukan *private chat* yang dilakukan oleh tutor dan mahasiswa.



Gambar 3. 24 Activity Diagram Melakukan Private Chat

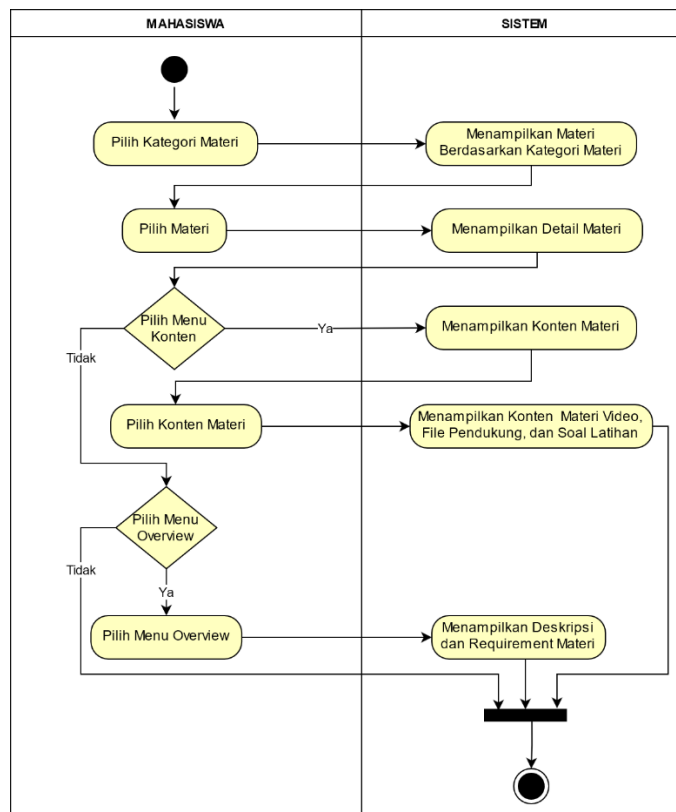
Berdasarkan Gambar 3.24 dapat dijelaskan bahwa alur proses untuk aktivitas dalam melakukan *private chat* yaitu:

- 1) Mahasiswa memilih menu *private chat*.
- 2) Sistem akan menampilkan halaman daftar tutor yang akan menjadi penerima pesan.
- 3) Mahasiswa memilih tutor dan sistem akan menampilkan halaman *chat*.
- 4) Mahasiswa menulis pesan dan klik kirim. Setelah itu sistem akan memproses pengiriman pesan agar sampai kepada penerima.
- 5) Ketika pesan sukses untuk dikirim, tutor akan menerima pesan baru. Tutor memilih menu *private chat* dan sistem akan menampilkan halaman daftar *chat* dari mahasiswa.
- 6) Tutor memilih pesan dari mahasiswa dan sistem akan menampilkan halaman *chat*.

- 7) Jika tutor ingin membalas pesan tersebut maka tutor menuliskan pesan balasan di halaman *chat* yang sudah ditampilkan kemudian klik kirim dan sistem akan memproses kembali pengiriman pesan. Jika tutor tidak ingin membalas maka aktivitas dalam melakukan *private chat* berakhir.

19. Activity Diagram Melihat Materi

Activity diagram melihat materi pada Gambar 3.25 menggambarkan semua alur proses dalam melihat materi yang dilakukan oleh mahasiswa.



Gambar 3. 25 Activity Diagram Melihat Materi

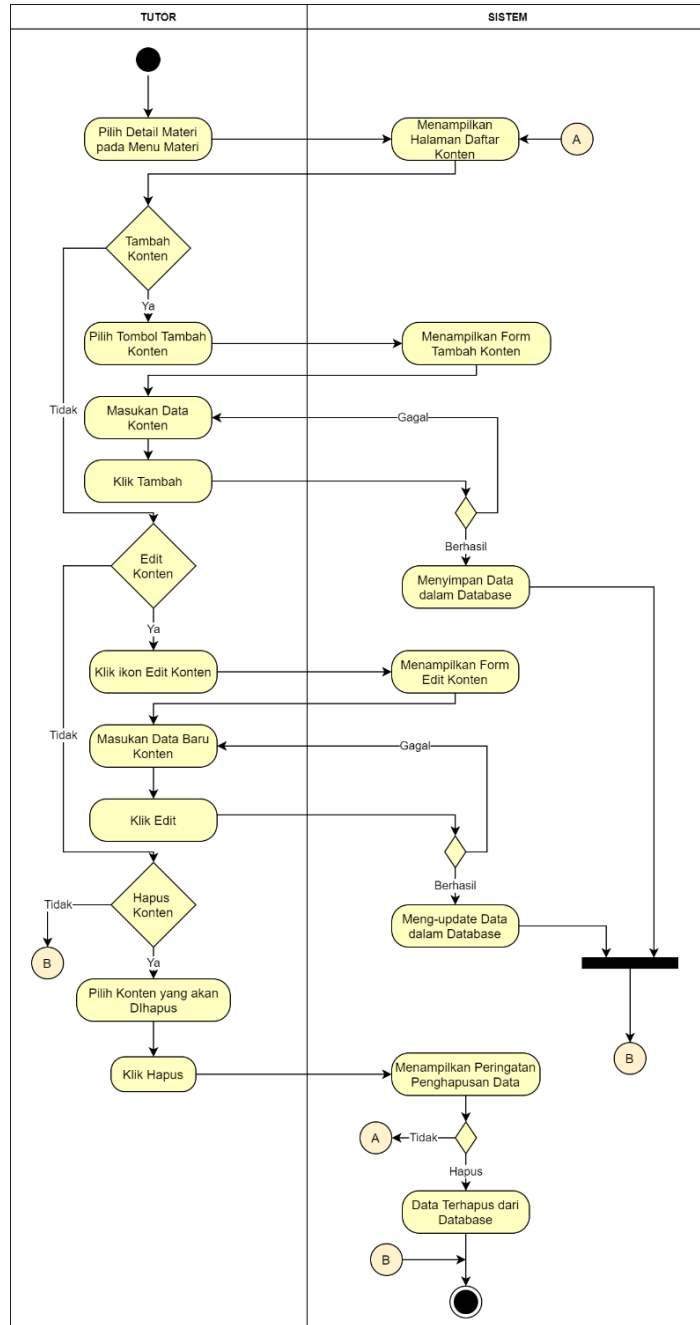
Berdasarkan Gambar 3.25 dapat dijelaskan bahwa alur proses untuk aktivitas dalam melihat materi yaitu:

- 1) Mahasiswa memilih kategori materi.
- 2) Sistem menampilkan halaman daftar materi sesuai dengan kategori yang dipilih.
- 3) Mahasiswa memilih materi dan sistem akan menampilkan detail materi tersebut.
- 4) Di dalam detail materi terdapat dua menu. Jika mahasiswa memilih menu konten maka sistem akan menampilkan daftar konten dari materi yang berisi materi video,

file pendukung materi, dan pengumpulan tugas. Jika mahasiswa memilih menu *overview* maka sistem akan menampilkan deskripsi dan informasi materi.

20. Activity Diagram Mengelola Konten

Activity diagram mengelola konten pada Gambar 3.26 menggambarkan semua alur proses dalam pengelolaan konten yang dilakukan oleh tutor.



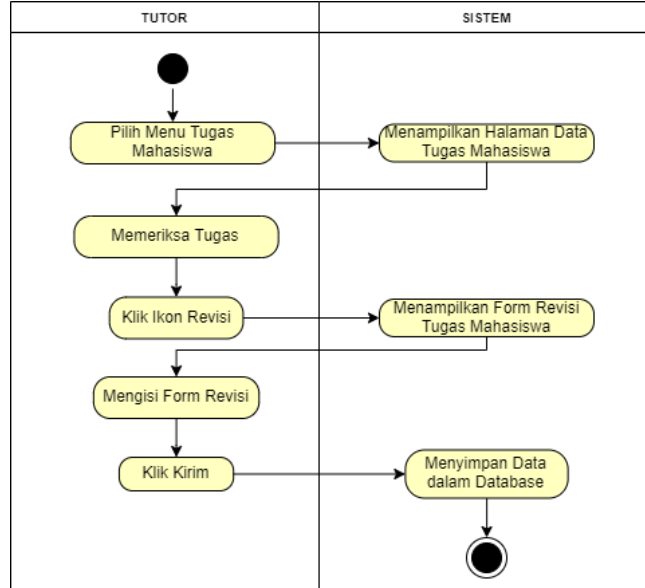
Gambar 3. 26 Activity Diagram Mengelola Konten

Berdasarkan Gambar 3.26 dapat dijelaskan bahwa alur proses untuk aktivitas dalam mengelola konten yaitu:

- 1) Tutor memilih menu materi dan klik ikon detail materi.
- 2) Sistem akan menampilkan halaman detail materi yang berisi daftar konten. Di dalam halaman daftar konten terdapat empat kondisi yaitu admin dapat melihat, menambah, mengubah, dan menghapus data konten. Berikut ini merupakan penjelasan masing-masing kondisi yaitu:
 - a. Jika tutor memilih untuk menambah konten maka admin memilih *button* tambah data dan sistem akan menampilkan *form* penambahan data. Selanjutnya tutor memasukkan data konten setelah itu klik *button* tambah. Kemudian sistem akan menyimpan data tersebut ke dalam *database*.
 - b. Jika tutor memilih untuk mengubah data konten maka tutor memilih data mana yang akan diubah setelah itu klik ikon pensil. Sistem akan menampilkan *form* pembaruan konten. Kemudian tutor mengubah data konten pada *form* tersebut lalu klik simpan. Sistem akan menyimpan pembaruan data tersebut dalam *database*.
 - c. Jika tutor memilih untuk menghapus konten maka tutor memilih data mana yang akan dihapus. Ketika klik ikon hapus lalu sistem akan menampilkan pesan peringatan sebagai tanda konfirmasi bahwa admin benar-benar akan menghapus data tersebut, kemudian sistem akan menghapus data dari *database*.

21. *Activity Diagram* Memberi Revisi Tugas

Activity diagram mengelola konten pada Gambar 3.27 menggambarkan semua alur proses dalam memberikan revisi pada tugas mahasiswa yang dilakukan oleh tutor.



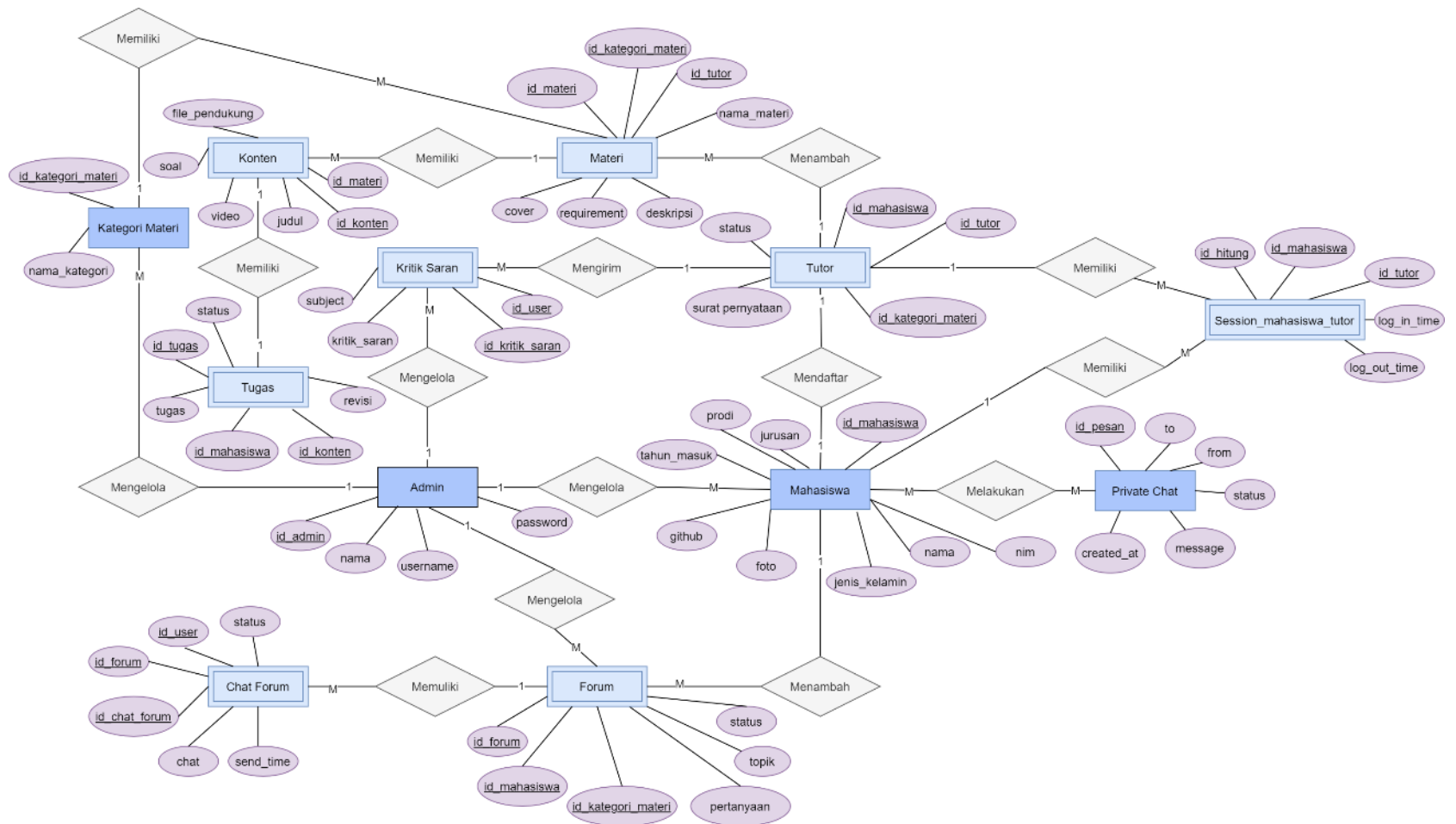
Gambar 3. 27 Activity Diagram Memberi Revisi Tugas

Berdasarkan Gambar 3.27 dapat dijelaskan bahwa alur proses untuk aktivitas dalam memberi revisi tugas yaitu:

- 1) Tutor memilih menu tugas mahasiswa dan sistem menampilkan halaman yang memuat data tugas mahasiswa.
- 2) Tutor memeriksa tugas mahasiswa.
- 3) Tutor klik ikon revisi dan sistem menampilkan halaman petunjuk dan *form* pemberian revisi.
- 4) Tutor mengisi *form* revisi dan klik kirim. Kemudian sistem akan menyimpan revisi di dalam *database*.

3.3.4 Perancangan Basis Data

Basis data merupakan kumpulan data terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Perancangan skema *database* pada sistem ini didasarkan pada *use case diagram* yang telah dibuat. Berdasarkan fitur yang telah disebutkan, maka dapat dianalisa tabel apa saja yang dibutuhkan sebagai penyimpanan data sistem. Selain merancang tabel dan entitas, juga dilakukan klasifikasi terhadap hubungan antar tabel dan entitas. Perancangan basis data ditunjukkan pada Gambar 3.28 di bawah ini.



Gambar 3. 28 Entity Relationship Diagram (ERD)

Berdasarkan Gambar 3.28 entitas yang dimiliki yaitu admin, mahasiswa, kategori materi, tutor, materi, konten, tugas, forum, chat forum, private chat, session mahasiswa dan tutor, serta kritik dan saran. Semua entitas saling berhubungan satu sama lain seperti admin dapat mengelola entitas mahasiswa, kategori materi, kritik dan saran serta forum. Entitas kategori materi memiliki banyak materi yang dibuat oleh tutor dimana di setiap materi dapat memuat banyak konten dan di setiap konten terdapat tugas yang harus dikerjakan oleh mahasiswa. Entitas mahasiswa juga dapat menambah forum dan melakukan private chat. Serta entitas tutor dan mahasiswa masing-masing terhubung dengan entitas session_mahasiswa_tutor.

3.3.5 Perancangan Antarmuka

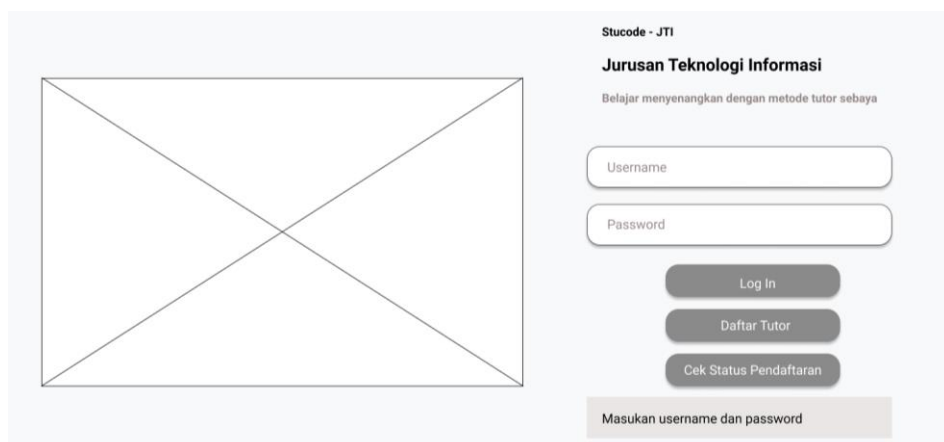
Tampilan antarmuka merupakan sebuah visual yang menjadi penghubung antara sistem agar dapat berinteraksi dengan pengguna. Berikut ini merupakan rancangan antarmuka sistem yang terdiri dari rancangan antarmuka pada halaman admin, tutor, mahasiswa, dan calon tutor.

3.3.5.1 Halaman Mahasiswa

Halaman mahasiswa merupakan halaman yang akan ditampilkan untuk mahasiswa pada sistem. Di bawah ini merupakan rancangan antarmuka pada halaman mahasiswa.

a. Halaman Login

Rancangan halaman login akan ditunjukkan pada Gambar 3.29 di bawah ini.



The image shows a login page design for 'Stucode - JTI' (Jurusan Teknologi Informasi). On the left, there is a large rectangular area with a diagonal cross, indicating a placeholder for a logo or image. On the right, the page header includes the institution name and a tagline. Below the header, there are two input fields for 'Username' and 'Password'. Underneath these fields are three buttons: 'Log In', 'Daftar Tutor', and 'Cek Status Pendaftaran'. At the bottom right, there is a light gray box containing the text 'Masukan username dan password'.

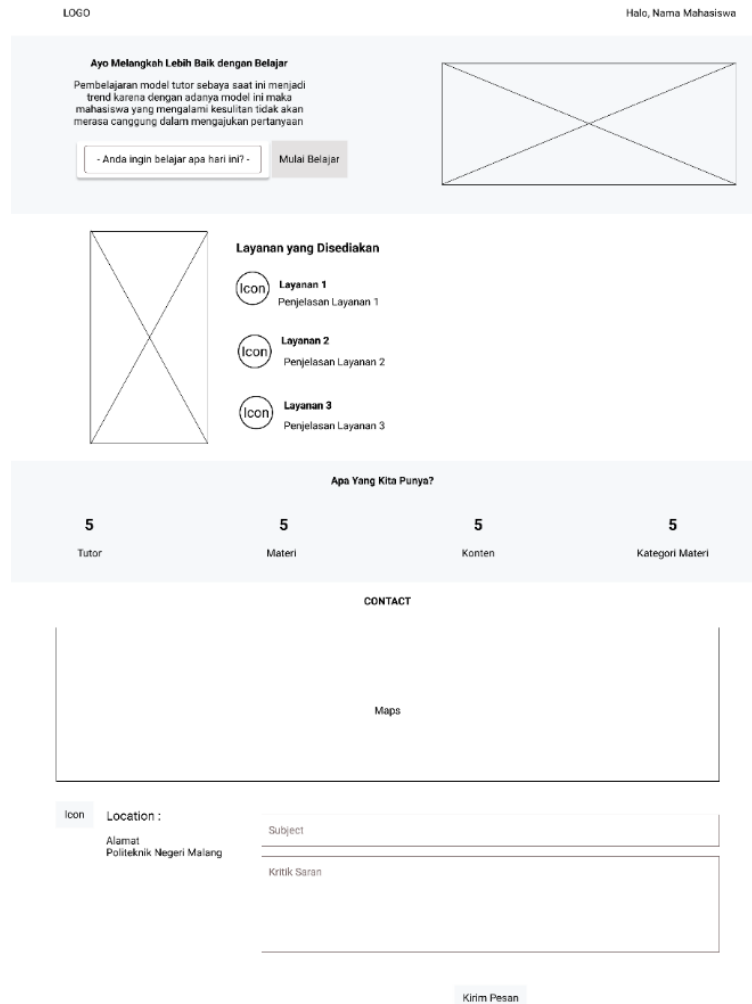
Gambar 3. 29 Perancangan Halaman Login

Pada Gambar 3.29 terdapat dua kolom isian untuk menuliskan *username* dan *password*, serta tiga tombol diantaranya tombol *login* untuk melakukan *login*, tombol *daftar tutor*

yang akan mengarah pada halaman pendaftaran tutor, dan tombol cek status pendaftaran akan mengarah pada halaman cek status pendaftaran calon tutor. Halaman *login* ini digunakan oleh mahasiswa, tutor, maupun admin untuk masuk ke dalam sistem.

b. Halaman Utama Mahasiswa

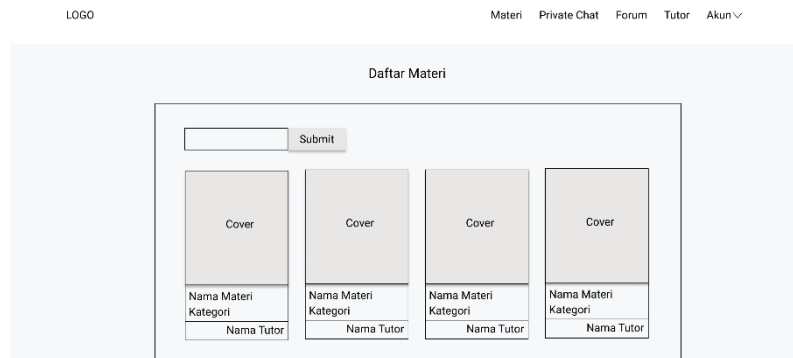
Gambar 3.30 merupakan halaman tampilan utama dari sisi pengguna setelah *login*. Halaman ini berisi nama dari mahasiswa yang sedang *login* pada bagian pojok atas kanan, *form select* untuk memilih kategori materi yang akan dipelajari, dan tombol mulai belajar dimana jika tombol ini diklik maka akan mengarah ke halaman daftar materi sesuai dengan kategori yang dipilih. Rancangan halaman utama mahasiswa akan ditunjukkan pada Gambar 3.30 di bawah ini.



Gambar 3. 30 Perancangan Halaman Utama Mahasiswa

c. Halaman Daftar Materi

Rancangan halaman daftar materi pada mahasiswa akan ditunjukkan pada Gambar 3.31 di bawah ini.

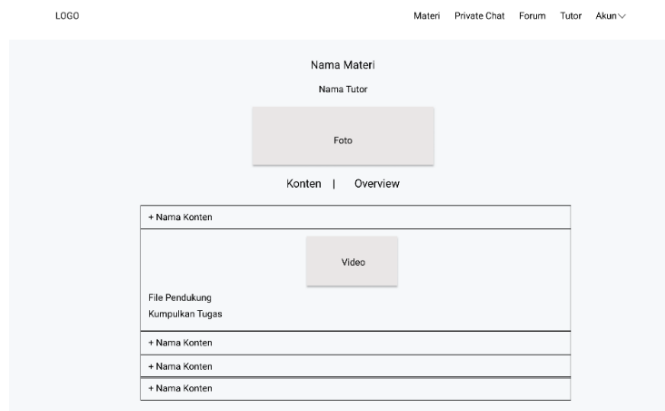


Gambar 3. 31 Perancangan Halaman Daftar Materi

Pada Gambar 3.31 merupakan halaman daftar materi yang menampilkan semua materi sesuai dengan kategori yang dipilih pada halaman utama. Hal yang ditampilkan yaitu gambar *cover* materi dan nama tutor dari materi tersebut. Selain itu terdapat tombol pencarian untuk melakukan pencarian sesuai dengan nama materi yang akan dicari.

d. Halaman Daftar Konten

Rancangan halaman daftar konten pada mahasiswa akan ditunjukkan pada Gambar 3.32 di bawah ini.

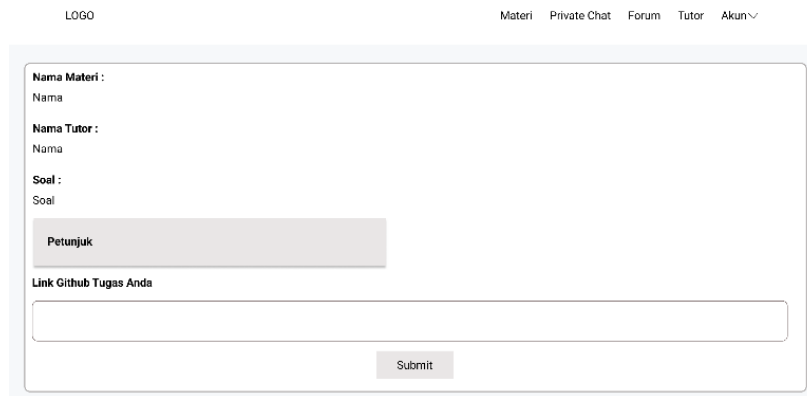


Gambar 3. 32 Perancangan Halaman Daftar Konten

Pada Gambar 3.32 tampilan seluruh daftar konten sesuai dengan materi yang dipilih. Halaman berisi judul materi, *cover* materi, deskripsi materi, daftar konten, video, *file* pendukung konten, soal latihan, dan tombol pengumpulan tugas.

e. Halaman Mengumpulkan Tugas

Rancangan halaman mengumpulkan pada mahasiswa akan ditunjukkan pada Gambar 3.33 di bawah ini.



LOGO Materi Private Chat Forum Tutor Akun

Nama Materi :
Nama

Nama Tutor :
Nama

Soal :
Soal

Petunjuk

Link Github Tugas Anda

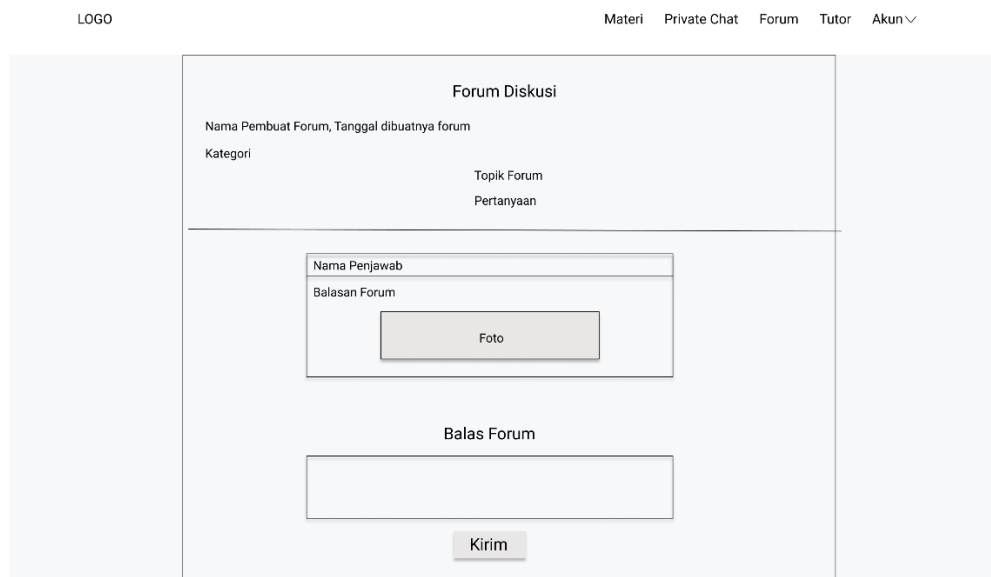
Submit

Gambar 3. 33 Perancangan Halaman Mengumpulkan Tugas

Pada Gambar 3.33 diketahui halaman mengumpulkan tugas berisi dari informasi materi yang dikerjakan, kolom isian pengumpulan *link* Github, dan tombol *submit* untuk mengkonfirmasi jawaban benar-benar dikirim.

f. Halaman Forum

Rancangan halaman forum pada mahasiswa akan ditunjukkan pada Gambar 3.34 di bawah ini.



LOGO Materi Private Chat Forum Tutor Akun

Forum Diskusi

Nama Pembuat Forum, Tanggal dibuatnya forum

Kategori

Topik Forum

Pertanyaan

Nama Penjawab

Balasan Forum

Foto

Balas Forum

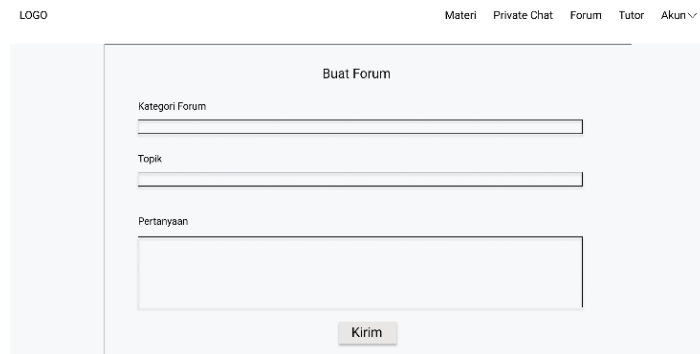
Kirim

Gambar 3. 34 Perancangan Halaman Forum

Pada Gambar 3.34 yang ditampilkan pada halaman forum antara lain topik forum yang diajukan, nama pembuat forum, tanggal pembuatan forum, jawaban, kolom isian untuk membalas diskusi, dan tombol kirim untuk mengiris jawaban diskusi forum.

g. Halaman Membuat Forum

Rancangan halaman membuat forum pada mahasiswa akan ditunjukkan pada Gambar 3.35 di bawah ini

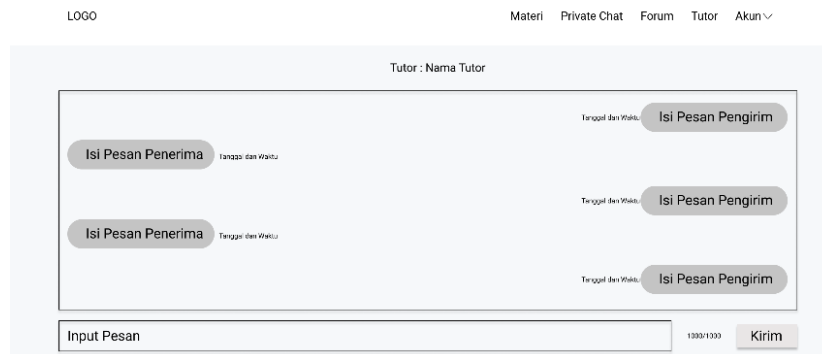


Gambar 3. 35 Perancangan Halaman Membuat Forum

Pada Gambar 3.35 diketahui bahwa pada halaman membuat forum terdiri dari *form* yang berisi kategori forum, topik yang akan ditanyakan, dan mahasiswa juga dapat menambahkan alamat URL referensi dan memasukkan gambar, serta terdapat tombol kirim untuk mengirim forum.

h. Halaman *Private Chat*

Rancangan halaman *private chat* akan ditunjukkan pada Gambar 3.36 di bawah ini



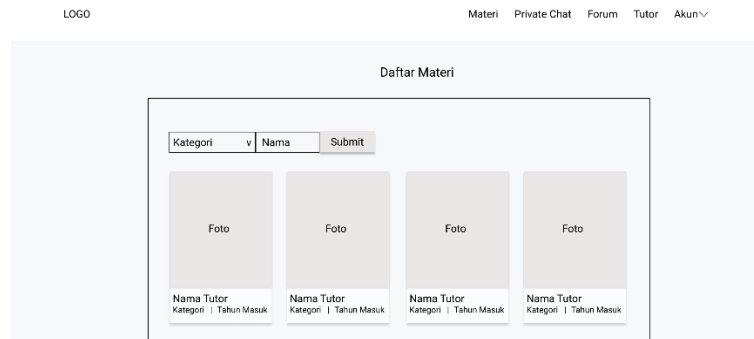
Gambar 3. 36 Perancangan Halaman *Private Chat*

Pada Gambar 3.36 tampilan halaman *private chat* terdiri dari nama penerima pesan, isi pesan, kolom isian untuk membalas, dan tombol untuk mengirim pesan. Baik pada

halaman tutor maupun mahasiswa, tampilan *private chat* sama seperti yang ditunjukkan pada Gambar 3.36.

i. Halaman Daftar Tutor

Rancangan halaman daftar tutor pada mahasiswa akan ditunjukkan pada Gambar 3.37 di bawah ini.

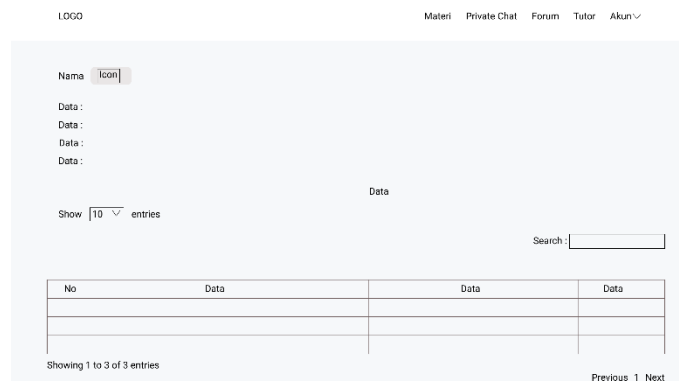


Gambar 3. 37 Perancangan Halaman Daftar Tutor

Pada Gambar 3.37 merupakan halaman daftar tutor yang menampilkan semua tutor yang telah diterima pada sistem. Hal yang ditampilkan yaitu foto tutor, nama, kategori tutor, dan tahun angkatan. Selain itu terdapat tombol pencarian untuk melakukan pencarian sesuai kategori atau nama tutor yang akan dicari.

j. Halaman Profil

Rancangan halaman profil mahasiswa dan tutor akan ditunjukkan pada Gambar 3.38 di bawah ini.



Gambar 3. 38 Perancangan Halaman Profil

Pada Gambar 3.38 merupakan rancangan tampilan detail akun mahasiswa yang berisi data diri mahasiswa, tombol *edit* profil untuk mengubah profil, dan tabel yang berisi

daftar status tugas yang telah dikerjakan. Hampir sama dengan tutor, yang membedakan adalah pada halaman detail tutor terdapat foto tutor, tabel berisi daftar materi yang dimiliki, dan tidak terdapat tombol *edit* profil.

3.3.5.2 Halaman Admin

Halaman admin merupakan halaman yang akan ditampilkan untuk admin pada sistem. Di bawah ini merupakan rancangan antarmuka pada halaman admin.

a. Halaman *Dashboard* Admin

Rancangan halaman *dashboard* admin akan ditunjukkan pada Gambar 3.39 di bawah ini.

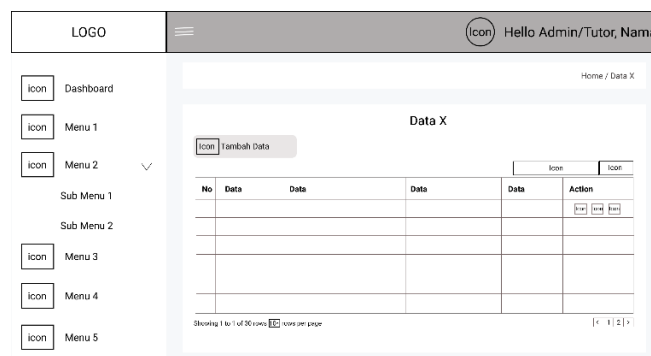


Gambar 3. 39 Perancangan Halaman *Dashboard* Admin

Pada Gambar 3.39 tampilan *dashboard* berisi informasi berupa grafik yang menunjukkan data mahasiswa dan tutor yang paling aktif serta jumlah data yang dimiliki oleh admin mulai dari data mahasiswa, tutor, kategori materi, forum, serta kritik dan saran.

b. Halaman Menampilkan Data

Rancangan halaman menampilkan data ditunjukkan pada Gambar 3.40 di bawah ini.



Gambar 3. 40 Perancangan Halaman Menampilkan Data

Pada Gambar 3.40 merupakan tampilan yang digunakan untuk menampilkan data mulai dari data mahasiswa, tutor, materi, kategori materi, tugas, calon tutor, serta kritik dan

saran. Dalam tabel yang menyajikan data tersebut juga terdapat satu kolom bernama *action* yang berisi tiga ikon yaitu ikon detail untuk melihat detail data, ikon *update* untuk mengubah data, dan ikon hapus untuk menghapus data. Halaman ini juga akan digunakan pada halaman tutor seperti untuk menampilkan materi dan verifikasi tugas mahasiswa.

c. Halaman Menambah Data

Rancangan halaman menambah data akan ditunjukkan pada Gambar 3.41 di bawah ini.

Gambar 3. 41 Perancangan Halaman Menambah Data

Pada Gambar 3.41 merupakan tampilan yang digunakan untuk menambahkan data mulai dari data mahasiswa, materi, konten, kritik dan saran, kategori materi, dan revisi. Pada halaman tersebut berisi judul penambahan data, kolom isian sesuai dengan data yang akan ditambahkan, dan tombol tambah yang digunakan sebagai proses untuk menyimpan data. Halaman ini juga akan ditampilkan di halaman tutor untuk menambah materi, konten, serta kritik dan saran.

d. Halaman Mengubah Data

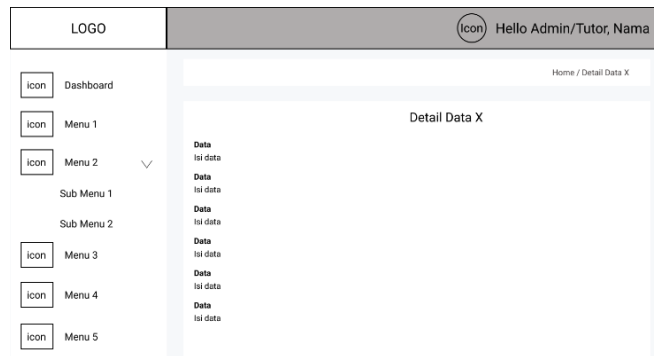
Rancangan halaman mengubah data akan ditunjukkan pada Gambar 3.42 di bawah ini.

Gambar 3. 42 Perancangan Halaman Mengubah Data

Pada Gambar 3.42 merupakan tampilan yang digunakan untuk mengubah data yang telah ada mulai dari data mahasiswa, materi, konten, kategori materi, dan *password*. Pada halaman tersebut berisi judul perubahan data, kolom isian sesuai dengan data yang akan diubah, dan tombol *edit* yang digunakan sebagai proses untuk menyimpan data. Halaman ini juga akan ditampilkan di halaman tutor untuk mengubah materi, konten, dan profil.

e. Halaman Melihat Detail Data

Rancangan halaman melihat detail data ditunjukkan pada Gambar 3.43 di bawah ini.

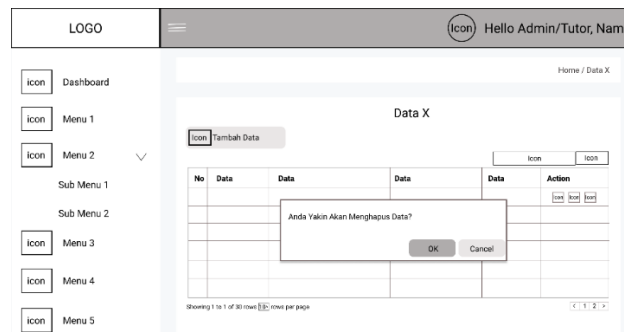


Gambar 3. 43 Perancangan Halaman Melihat Detail Data

Pada Gambar 3.43 merupakan tampilan yang digunakan untuk melihat detail dari suatu data yang dipilih. Pada gambar di atas merupakan contoh detail dari data mahasiswa sehingga data yang ditampilkan berupa semua data dari mahasiswa tersebut seperti nama, NIM, jenis kelamin, jurusan, program studi, tahun masuk, dan alamat Github.

f. Halaman Menghapus Data

Rancangan halaman menghapus data akan ditunjukkan pada Gambar 3.44 di bawah ini.



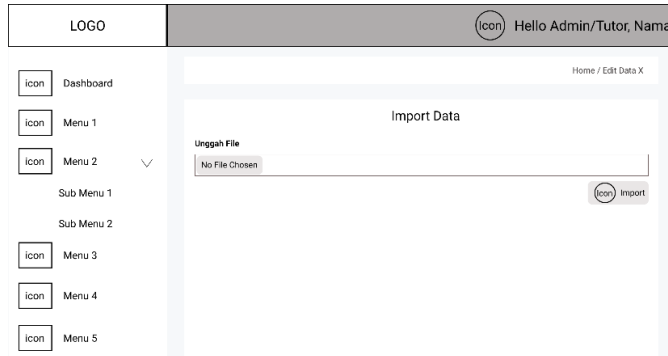
Gambar 3. 44 Perancangan Halaman Menghapus Data

Pada Gambar 3.44 merupakan halaman untuk menghapus data yang berupa *pop-up message* setelah pengguna menekan ikon hapus yang terdapat di setiap baris pada tabel

data. *Pop-up message* tersebut menampilkan konfirmasi penghapusan data. Pada halaman tutor halaman ini juga akan muncul ketika tutor melakukan hapus data pada data materi, konten, dan tugas mahasiswa.

g. Halaman *Import Data*

Rancangan halaman *import* data akan ditunjukkan pada Gambar 3.45 di bawah ini.



Gambar 3. 45 Perancangan Halaman *Import Data*

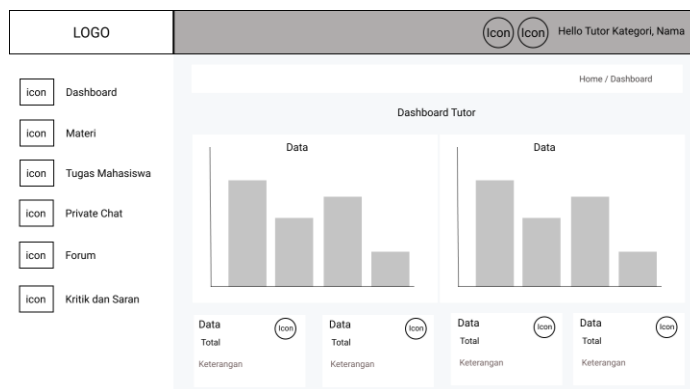
Berdasarkan Gambar 3.45 halaman *import* data berisi *form* untuk mengunggah *file* yang akan di-*import* dan tombol *import*. Halaman ini digunakan admin untuk meng-*import* data mahasiswa dari *file excel* dengan tujuan admin tidak memasukkan data mahasiswa secara satu per satu.

3.3.5.3 Halaman Tutor

Halaman tutor merupakan halaman yang akan ditampilkan untuk tutor pada sistem. Di bawah ini merupakan rancangan antarmuka pada halaman tutor.

a. Halaman *Dashboard* Tutor

Rancangan halaman *dashboard* tutor akan ditunjukkan pada Gambar 3.46 di bawah ini.

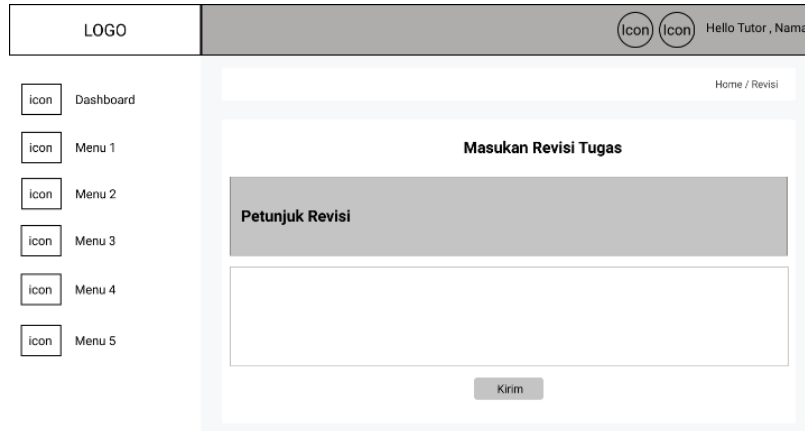


Gambar 3. 46 Perancangan Halaman *Dashboard* Tutor

Pada Gambar 3.46 tampilan *dashboard* tutor berisi informasi berupa grafik yang menunjukkan konten terfavorit yang dimiliki dan data mahasiswa yang paling aktif serta jumlah data yang dimiliki oleh tutor mulai dari materi, konten, dan forum.

b. Halaman Memberi Revisi Tugas

Rancangan halaman memberi revisi tugas ditunjukkan pada Gambar 3.47 di bawah ini.



Gambar 3. 47 Perancangan Halaman Memberi Revisi Tugas

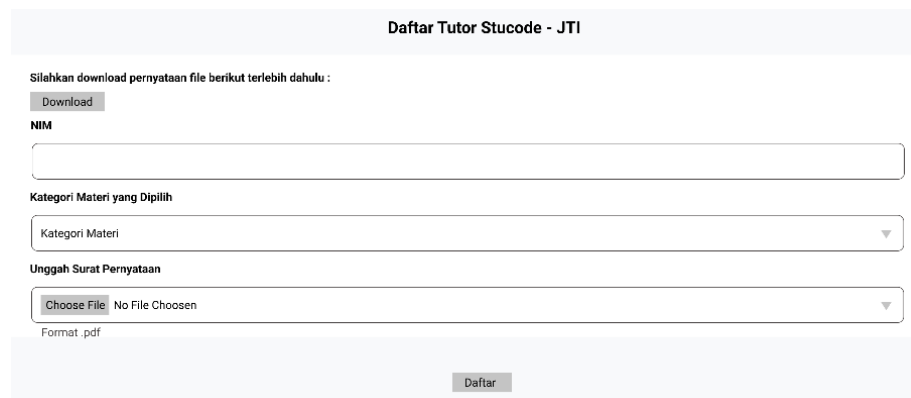
Pada Gambar 3.47 halaman ini berisi catatan berupa petunjuk pemberian revisi untuk tutor, isian untuk memberikan revisi, dan tombol kirim untuk mengirim revisi.

3.3.5.4 Halaman Calon Tutor

Halaman calon tutor merupakan halaman yang akan ditampilkan untuk calon tutor pada sistem. Di bawah ini merupakan rancangan antarmuka pada halaman calon tutor.

a. Halaman Mendaftar Tutor

Rancangan halaman mendaftar tutor akan ditunjukkan pada Gambar 3.48 di bawah ini.

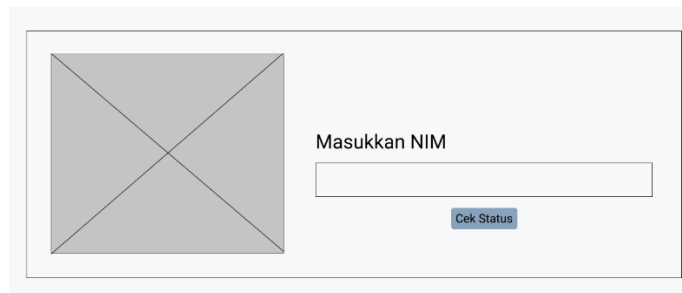


Gambar 3. 48 Perancangan Halaman Mendaftar Tutor

Pada Gambar 3.48 terdapat tombol untuk mengunduh surat pernyataan ketersediaan dalam menjadi tutor, kolom isian NIM, kategori yang dipilih, dan unggahan berkas yang berisi surat pernyataan yang telah diisi, serta tombol daftar untuk mendaftar.

b. Halaman Memeriksa Status Pendaftaran

Rancangan halaman memeriksa status pendaftaran pada calon tutor akan ditunjukkan pada Gambar 3.49 di bawah ini.

The image shows a web interface for checking registration status. On the left, there is a grey square placeholder with a large 'X' over it. To the right of this placeholder is a form area. At the top of the form area, the text 'Masukkan NIM' is displayed above a white rectangular input field. Below the input field is a blue button with the text 'Cek Status' in white.

Gambar 3. 49 Perancangan Halaman Memeriksa Status Pendaftaran

Pada Gambar 3.49 terdapat kolom isian untuk memasukkan NIM yang telah didaftarkan dan tombol cari untuk melakukan pencarian.

3.4 Pengujian Sistem

Pada subbab ini akan dijelaskan mengenai metode pengujian pada sistem yang akan dibangun. Pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibangun sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. Pengujian sistem akan dilakukan dengan teknik pengujian *equivalence partitioning* dan *usability*.

3.4.1 Metode Pengujian *Equivalence Partitioning*

Pengujian akan dilakukan kepada Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang. Pengujian pada sistem menggunakan teknik *Equivalence Partitioning* (Partisi Ekuivalensi) dengan metode *Black Box*. Tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah menampilkan keadaan valid terhadap masukan pengguna sehingga didapatkan sistem yang berkualitas. Pengujian ini akan dilakukan oleh salah satu Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang yang akan menguji semua fitur di setiap aktor.

3.4.1.1 Tahapan Pengujian *Equivalence Partitioning*

Dalam pengujian sistem yang akan dilakukan memiliki beberapa tahapan yang dilakukan. Tahap pengujian bertujuan untuk menentukan keadaan valid atau invalid terhadap kondisi input pada perangkat lunak. Dalam tahap pengujian *equivalence partitioning* hal-hal yang dilakukan yaitu:

1. Menentukan *use case* yang akan diujikan sesuai dengan rancangan *use case* yang telah ada.
2. Menentukan kriteria apa saja yang akan diujikan berdasarkan pilihan *use case*.
3. Mendefinisikan partisi dari masing-masing kriteria yang telah ditentukan. Kriteria akan dipecah menjadi beberapa partisi sesuai dengan karakteristik kriteria tersebut.
4. Membuat data uji berdasarkan partisi yang telah dibuat pada langkah sebelumnya yang digunakan sebagai bahan penentuan kasus uji.
5. Membuat kasus uji berdasarkan data uji yang telah dirancang.
6. Melakukan pengujian sistem berdasarkan daftar kasus uji yang telah dibuat.
7. Melakukan evaluasi dari hasil pengujian yang telah dilakukan. Tahap evaluasi akan menghasilkan sebuah umpan balik bagi pengembang berupa informasi apakah sistem yang dibangun sudah sesuai atau belum serta dapat dijelaskan kekurangan dan kelebihan dari sistem sehingga pengembang dapat memperbaiki sistem untuk menjadi lebih baik lagi.

3.4.1.2 Penentuan *Use Case*

Tahapan pertama yang harus dilakukan sebelum melakukan pengujian dengan teknik *equivalence partitioning* yaitu melakukan penentuan *use case* akan diujikan. Karena dalam aplikasi yang akan dibangun memiliki empat pengguna yaitu admin, mahasiswa, tutor, dan calon tutor maka akan dilakukan pembagian *use case* untuk masing-masing pengguna.

3.4.1.2.1 Penentuan *Use Case* Admin

Penentuan *use case* pada admin digunakan untuk mengetahui *use case* yang akan diuji pada sisi admin. Penentuan *use case* admin akan dapat dilihat pada Tabel 3.24 di bawah ini.

Tabel 3. 24 Penentuan *Use Case* Admin

Kode Use Case	Nama Use Case
UC-A 01	Mengelola Forum
UC-A 02	Mengelola Data Mahasiswa
UC-A 03	Mengelola Data Tutor
UC-A 04	Mengelola Data Kategori Materi
UC-A 05	Mengelola Kritik dan Saran
UC-A 06	Mengubah <i>Password</i>
UC-A 07	Memverifikasi Calon Tutor

3.4.1.2.2 Penentuan *Use Case* Calon Tutor

Penentuan *use case* pada calon tutor digunakan untuk mengetahui *use case* yang akan diuji pada sisi calon tutor. Penentuan *use case* calon tutor akan dapat dilihat pada Tabel 3.25 di bawah ini.

Tabel 3. 25 Penentuan *Use Case* Calon Tutor

Kode Use Case	Nama Use Case
UC-CT 01	Mendaftar Tutor
UC-CT 02	Memeriksa Status Pendaftaran

3.4.1.2.3 Penentuan *Use Case* Tutor

Penentuan *use case* pada tutor digunakan untuk mengetahui *use case* yang akan diuji pada sisi tutor. Penentuan *use case* tutor akan dapat dilihat pada Tabel 3.26 di bawah ini.

Tabel 3. 26 Penentuan *Use Case* Tutor

Kode Use Case	Nama Use Case
UC-T 01	Mengelola Materi
UC-T 02	Mengelola Konten
UC-T 03	Memverifikasi Tugas
UC-T 04	Memberi Revisi Tugas

UC-T 05	Melakukan <i>Private Chat</i>
UC-T 06	Menjawab Forum
UC-T 07	Mengelola Profil
UC-T 08	Mengirim Kritik dan Saran

3.4.1.2.4 Penentuan *Use Case* Mahasiswa

Penentuan *use case* mahasiswa digunakan untuk mengetahui *use case* yang akan diuji pada sisi mahasiswa. Penentuan *use case* mahasiswa ditunjukkan Tabel 3.27 di bawah ini.

Tabel 3. 27 Penentuan *Use Case* Mahasiswa

Kode <i>Use Case</i>	Nama <i>Use Case</i>
UC-M 01	Mengumpulkan Tugas
UC-M 02	Mengelola Profil
UC-M 03	Menjawab Forum
UC-M 04	Bertanya pada Forum
UC-M 05	Mengirim Kritik dan Saran
UC-M 06	Melakukan <i>Private Chat</i>
UC-M 07	Melihat Materi dan Konten
UC-M 08	Melihat Tutor

3.4.1.3 Penentuan Kriteria Uji

Setelah ditentukan *use case* yang akan diuji, tahap selanjutnya yaitu menentukan kriteria pengujian berdasarkan *use case* yang dibuat. Penentuan kriteria akan disusun sesuai dengan penentuan *use case* yang dibuat.

3.4.1.3.1 Penentuan Kriteria Uji Admin

Pada Tabel 3.28 di bawah ini akan dijelaskan detail kriteria uji yang di ujikan dari setiap *use case* admin.

Tabel 3. 28 Penentuan Kriteria Uji Admin

Kode Kriteria	Kode <i>Use Case</i>	Kriteria
K-A 01	UC-A 01	Menghapus Forum

K-A 02	UC-A 02	Menambah Data Mahasiswa
K-A 03	UC-A 02	Mengubah Data Mahasiswa
K-A 04	UC-A 02	Menghapus Data Mahasiswa
K-A 05	UC-A 03	Menghapus Data Tutor
K-A 06	UC-A 04	Menambah Data Kategori Materi
K-A 07	UC-A 04	Mengubah Data Kategori Materi
K-A 08	UC-A 04	Menghapus Data Kategori Materi
K-A 09	UC-A 05	Menghapus Kritik dan Saran
K-A 10	UC-A 06	Mengisi <i>Form</i> Perubahan <i>Password</i>
K-A 11	UC-A 07	Memverifikasi Calon Tutor

3.4.1.3.2 Penentuan Kriteria Uji Calon Tutor

Pada Tabel 3.29 di bawah ini akan dijelaskan detail kriteria uji yang diujikan dari setiap *use case* calon tutor.

Tabel 3. 29 Penentuan Kriteria Uji Calon Tutor

Kode Kriteria	Kode Use Case	Kriteria
K-CT 01	UC-CT 01	Mengisi <i>Form</i> Pendaftaran Tutor
K-CT 02	UC-CT 02	Mengisi <i>Form</i> Pemeriksaan Status

3.4.1.3.3 Penentuan Kriteria Uji Tutor

Pada Tabel 3.30 di bawah ini akan dijelaskan detail kriteria uji yang diujikan dari setiap *use case* tutor.

Tabel 3. 30 Penentuan Kriteria Uji Tutor

Kode Kriteria	Kode Use Case	Kriteria
K-T 01	UC-T 01	Menambah Data Materi
K-T 02	UC-T 01	Mengubah Data Materi
K-T 03	UC-T 01	Menghapus Data Materi
K-T 04	UC-T 02	Menambah Konten
K-T 05	UC-T 02	Mengubah Konten

K-T 06	UC-T 02	Menghapus Konten
K-T 07	UC-T 03	Memverifikasi Tugas
K-T 08	UC-T 04	Mengisi <i>Form</i> Pemberian Revisi Tugas
K-T 09	UC-T 05	Mengisi <i>Form</i> Kirim Pesan
K-T 10	UC-T 06	Mengisi <i>Form</i> Balas Forum
K-T 11	UC-T 07	Mengubah Profil
K-T 12	UC-T 08	Mengisi <i>Form</i> Pengiriman Kritik dan Saran

3.4.1.3.4 Penentuan Kriteria Uji Mahasiswa

Pada Tabel 3.31 di bawah ini akan dijelaskan detail kriteria uji yang diujikan dari setiap *use case* mahasiswa.

Tabel 3. 31 Penentuan Kriteria Uji Mahasiswa

Kode Kriteria	Kode <i>Use Case</i>	Kriteria
K-M 01	UC-M 01	Mengisi <i>Link</i> Pengumpulan Tugas
K-M 02	UC-M 02	Mengubah Profil
K-M 03	UC-M 03	Mengisi <i>Form</i> Balas Forum
K-M 04	UC-M 04	Mengisi <i>Form</i> Pembuatan Forum
K-M 05	UC-M 05	Mengisi <i>Form</i> Pengiriman Kritik dan Saran
K-M 06	UC-M 06	Mengisi <i>Form</i> Kirim Pesan
K-M 07	UC-M 07	Mengisi <i>Form</i> Pencarian Materi
K-M 08	UC-M 08	Mengisi <i>Form</i> Pencarian Tutor

3.4.1.4 Pendefinisian Partisi Uji

Pendefinisian partisi uji didasarkan pada kriteria yang telah dibuat seperti pada tahap sebelumnya di mana dalam tahap ini kriteria akan dipecah menjadi beberapa partisi lagi. Hasil dari pendefinisian partisi uji akan menghasilkan nilai valid dan invalid.

3.4.1.4.1 Pendefinisian Partisi Uji Admin

Pada Tabel 3.32 di bawah ini akan dijelaskan pendefinisian partisi uji dari setiap kriteria uji pada admin.

Tabel 3. 32 Pendefinisian Partisi Uji Admin

Kode Partisi	Kode Kriteria	Partisi Uji	Tipe Uji
KP-A 01	K-A 01	Forum dihapus	Valid
KP-A 02	K-A 01	Forum tidak dihapus	Invalid
KP-A 03	K-A 02	Input data NIM bukan dengan angka	Invalid
KP-A 04	K-A 02	Input data NIM dengan angka dalam rentang (1-10) karakter	Valid
KP-A 05	K-A 02	Input data NIM dengan angka melebihi 10 karakter	Invalid
KP-A 06	K-A 02	Input semua <i>field</i> pada <i>form</i>	Valid
KP-A 07	K-A 02	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> kecuali <i>field</i> “Alamat Github”	Invalid
KP-A 08	K-A 02	<i>Import</i> data mahasiswa dengan <i>file</i> yang memiliki ekstensi yang sesuai dan berukuran < 2Mb	Valid
KP-A 09	K-A 02	<i>Import</i> data mahasiswa dengan <i>file</i> yang memiliki ekstensi tidak sesuai dengan format (xls, xlsx)	Invalid
KP-A 10	K-A 02	<i>Import</i> data mahasiswa dengan <i>file</i> yang memiliki ekstensi yang sesuai dan berukuran > 2Mb	Invalid
KP-A 11	K-A 03	Input data NIM bukan dengan angka	Invalid
KP-A 12	K-A 03	Input data NIM dengan angka dalam rentang (1-10) karakter	Valid
KP-A 13	K-A 03	Input data NIM dengan angka melebihi 10 karakter	Invalid
KP-A 14	K-A 03	Input data di semua <i>field</i> pada <i>form</i>	Valid
KP-A 15	K-A 03	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> kecuali <i>field</i> “Alamat Github”	Invalid
KP-A 16	K-A 04	Data mahasiswa dihapus	Valid
KP-A 17	K-A 04	Data mahasiswa tidak dihapus	Invalid
KP-A 18	K-A 05	Data tutor dihapus	Valid
KP-A 19	K-A 05	Data tutor tidak dihapus	Invalid

KP-A 20	K-A 06	Input data nama kategori	Valid
KP-A 21	K-A 06	<i>Field</i> nama kategori dibiarkan kosong	Invalid
KP-A 22	K-A 07	Input data nama kategori	Valid
KP-A 23	K-A 07	<i>Field</i> nama kategori dibiarkan kosong	Invalid
KP-A 24	K-A 08	Data kategori materi dihapus	Valid
KP-A 25	K-A 08	Data kategori materi tidak dihapus	Invalid
KP-A 26	K-A 09	Data kritik dan saran dihapus	Valid
KP-A 27	K-A 09	Data kritik dan saran tidak dihapus	Invalid
KP-A 28	K-A 10	Input <i>password</i> lama yang tidak sesuai	Invalid
KP-A 29	K-A 10	Input <i>password</i> baru kurang dari delapan karakter	Invalid
KP-A 30	K-A 10	Input <i>password</i> lama yang sesuai dan <i>password</i> baru lebih dari delapan karakter	Valid
KP-A 31	K-A 11	Data calon tutor diverifikasi	Valid
KP-A 32	K-A 11	Data calon tutor tidak diverifikasi	Valid

3.4.1.4.2 Pendefinisian Partisi Uji Calon Tutor

Pada Tabel 3.33 di bawah ini akan dijelaskan pendefinisian partisi uji dari setiap kriteria uji pada calon tutor.

Tabel 3. 33 Pendefinisian Partisi Uji Calon Tutor

Kode Partisi	Kode Kriteria	Partisi Uji	Tipe Uji
KP-CT 01	K-CT 01	Input NIM yang telah terdaftar sebagai mahasiswa dan mengisi semua <i>field</i> pada <i>form</i>	Valid
KP-CT 02	K-CT 01	Input NIM yang belum terdaftar sebagai mahasiswa dan mengisi semua <i>field</i> pada <i>form</i>	Invalid
KP-CT 03	K-CT 01	Input NIM yang terdaftar sebagai mahasiswa tetapi ada salah satu <i>field</i> pada <i>form</i> yang kosong	Invalid
KP-CT 04	K-CT 01	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format (selain pdf)	Invalid

KP-CT 05	K-CT 01	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimal (2 Mb)	Invalid
KP-CT 06	K-CT 02	Input NIM yang tidak terdaftar sebagai mahasiswa	Invalid
KP-CT 07	K-CT 02	Input NIM yang tidak mendaftar sebagai tutor	Invalid
KP-CT 08	K-CT 02	Input NIM yang mendaftar sebagai tutor yang belum diverifikasi	Valid
KP-CT 09	K-CT 02	Input NIM yang mendaftar sebagai tutor yang telah diverifikasi	Valid

3.4.1.4.3 Pendefinisian Partisi Uji Tutor

Pada Tabel 3.34 di bawah ini akan dijelaskan pendefinisian partisi uji dari setiap kriteria uji pada tutor.

Tabel 3. 34 Pendefinisian Partisi Uji Tutor

Kode Partisi	Kode Kriteria	Partisi Uji	Tipe Uji
KP-T 01	K-T 01	Input data di semua <i>field</i> pada <i>form</i>	Valid
KP-T 02	K-T 01	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i>	Invalid
KP-T 03	K-T 01	Mengunggah gambar dengan ekstensi tidak sesuai format (jpg/jpeg/png)	Invalid
KP-T 04	K-T 01	Mengunggah gambar dengan ukuran melebihi kapasitas maksimal (2 Mb)	Invalid
KP-T 05	K-T 02	Input data di semua <i>field</i> pada <i>form</i>	Valid
KP-T 06	K-T 02	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i>	Invalid
KP-T 07	K-T 02	Mengunggah gambar dengan ekstensi tidak sesuai format (jpg/jpeg/png)	Invalid
KP-T 08	K-T 02	Mengunggah gambar dengan ukuran melebihi kapasitas maksimal (2 Mb)	Invalid
KP-T 09	K-T 03	Data materi dihapus	Valid
KP-T 10	K-T 03	Data materi tidak dihapus	Invalid

KP-T 11	K-T 04	Input data di semua <i>field</i> pada <i>form</i> dengan benar	Valid
KP-T 12	K-T 04	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> (kecuali <i>field</i> “ <i>file</i> pendukung” karena bersifat opsional)	Invalid
KP-T 13	K-T 04	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format (pdf)	Invalid
KP-T 14	K-T 04	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimum (2 Mb)	Invalid
KP-T 15	K-T 05	Input data di semua <i>field</i> pada <i>form</i> dengan benar	Valid
KP-T 16	K-T 05	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> (kecuali <i>field</i> “ <i>file</i> pendukung” karena bersifat opsional)	Invalid
KP-T 17	K-T 05	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format (pdf)	Invalid
KP-T 18	K-T 05	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimum (2 Mb)	Invalid
KP-T 19	K-T 06	Konten dihapus	Valid
KP-T 20	K-T 06	Konten tidak dihapus	Invalid
KP-T 21	K-T 07	Tugas diverifikasi	Valid
KP-T 22	K-T 08	Input data revisi pada <i>form</i>	Valid
KP-T 23	K-T 08	<i>Field</i> revisi tidak diisi	Invalid
KP-T 24	K-T 09	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid
KP-T 25	K-T 09	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid
KP-T 26	K-T 09	<i>Field</i> pesan tidak diisi	Invalid
KP-T 27	K-T 10	Input data di <i>field</i> balas forum	Valid
KP-T 28	K-T 10	<i>Field</i> balas forum tidak diisi	Invalid
KP-T 29	K-T 11	Input data di semua <i>field</i>	Valid
KP-T 30	K-T 11	Mengunggah gambar dengan ekstensi tidak sesuai dengan format (jpg/png/jpeg)	Invalid

KP-T 31	K-T 11	Mengunggah gambar dengan ukuran melebihi kapasitas maksimum (2 Mb)	Invalid
KP-T 32	K-T 12	Input data di semua <i>field</i> pada <i>form</i>	Valid
KP-T 33	K-T 12	Menyisakan satu atau lebih <i>field</i> tidak terisi	Invalid

3.4.1.4.4 Pendefinisian Partisi Uji Mahasiswa

Pada Tabel 3.35 di bawah ini akan dijelaskan pendefinisian partisi uji dari setiap kriteria uji pada mahasiswa.

Tabel 3. 35 Pendefinisian Partisi Uji Mahasiswa

Kode Partisi	Kode Kriteria	Partisi Uji	Tipe Uji
KP-M 01	K-M 01	Input <i>link</i> pada <i>field</i> pengumpulan tugas	Valid
KP-M 02	K-M 01	<i>Field</i> pengumpulan tugas tidak diisi	Invalid
KP-M 03	K-M 02	Input <i>link</i> alamat Github pada <i>field</i> “Alamat Github”	Valid
KP-M 04	K-M 02	<i>Field</i> alamat Github tidak diisi	Valid
KP-M 05	K-M 03	Input data di <i>field</i> balas forum	Valid
KP-M 06	K-M 03	<i>Field</i> balas forum tidak diisi	Invalid
KP-M 07	K-M 04	Input data di semua <i>field</i> tanya forum	Valid
KP-M 08	K-M 04	Menyisakan satu atau lebih <i>field</i> tidak diisi	Invalid
KP-M 09	K-M 05	Input data di semua <i>field</i> pada <i>form</i>	Valid
KP-M 10	K-M 05	Menyisakan satu atau lebih <i>field</i> tidak terisi	Invalid
KP-M 11	K-M 06	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid
KP-M 12	K-M 06	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid
KP-M 13	K-M 06	<i>Field</i> pesan tidak diisi	Invalid
KP-M 14	K-M 07	Input kata kunci pencarian materi yang tersedia	Valid
KP-M 15	K-M 07	Input kata kunci pencarian materi yang tidak tersedia	Valid
KP-M 16	K-M 07	<i>Field</i> pada <i>form</i> pencarian materi tidak diisi	Invalid
KP-M 17	K-M 08	Memilih kategori tutor dan mengisi nama tutor yang dicari	Valid

KP-M 18	K-M 08	Memilih kategori tutor tapi tidak menuliskan nama tutor yang dicari	Valid
KP-M 19	K-M 08	Menuliskan nama tutor yang dicari tetapi tidak memilih kategori tutor	Valid
KP-M 20	K-M 08	Pilihan kategori tutor dan kolom isian nama tutor tidak diisi	Invalid

3.4.1.5 Data Uji

Tahapan data uji merupakan langkah lanjutan setelah dibuat partisi uji. Pada tahap ini akan dibuat data uji dalam bentuk tabel berisi bahan penentuan kasus yang diujikan.

3.4.1.5.1 Data Uji Admin

Pada Tabel 3.36 di bawah ini akan dijelaskan data uji dari setiap partisi uji pada tahap sebelumnya pada admin.

Tabel 3. 36 Data Uji Admin

Kode Data Uji	Kode Partisi	Field	Data Uji	Tipe Uji
D-A 01	KP-A 01	Forum dihapus	Dihapus	Valid
D-A 02	KP-A 02	Forum tidak dihapus	Tidak dihapus	Invalid
D-A 03	KP-A 03	Input data NIM bukan dengan angka	Memasukkan NIM dengan “abcdef”	Invalid
D-A 04	KP-A 04	Input data NIM dengan angka dalam rentang (1-10) karakter	Memasukkan NIM dengan “1831710062”	Valid
D-A 05	KP-A 05	Input data NIM dengan angka melebihi 10 karakter	Memasukkan NIM dengan “12345678910”	Invalid
D-A 06	KP-A 06	Input semua <i>field</i> pada <i>form</i>	Mengisi <i>field</i> NIM dengan “1831710062”, nama “Redika Meima Mahdalena”, program	Valid

			studi memilih “Manajemen Informatika”, dan tahun masuk “2018”	
D-A 07	KP-A 07	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> kecuali <i>field</i> “Alamat Github”	Mengisi <i>field</i> NIM dengan “1841710001”, nama “Syakilla Maharani Augusta”, dan program studi memilih “Teknik Informatika”	Invalid
D-A 08	KP-A 08	Melakukan <i>import</i> data mahasiswa dengan ekstensi yang sesuai (xls, xlsx) dan berukuran < 2 Mb	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data Mahasiswa.xlsx” dengan ukuran <i>file</i> “32 Kb”	Valid
D-A 09	KP-A 09	Melakukan <i>import</i> data mahasiswa dengan ekstensi yang tidak sesuai dengan format	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data Mahasiswa.pdf”	Invalid
D-A 10	KP-A 10	Melakukan <i>import</i> data dengan ekstensi excel dan berukuran > 2 Mb	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data Mahasiswa.xls” dengan ukuran file 5 Mb.	Invalid
D-A 11	KP-A 11	Input data NIM bukan dengan angka	Mengubah data NIM dengan bukan angka “abcdferd”	Invalid
D-A 12	KP-A 12	Input data NIM dengan angka dalam rentang (1-10) karakter	Mengubah data NIM dengan rentang 1-10 karakter “1831710001”	Valid
D-A 13	KP-A 13	Input data NIM dengan angka melebihi 10 karakter	Mengisi data NIM melebihi 10 karakter “12345678900”	Invalid

D-A 14	KP-A 14	Input data di semua <i>field</i> pada <i>form</i>	Mengubah <i>field</i> nama dengan “Redika” dan membiarkan <i>field</i> lainnya terisi data sebelumnya	Valid
D-A 15	KP-A 15	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> kecuali <i>field</i> “Alamat Github”	Menghapus data nama dan membiarkan <i>field</i> lainnya terisi data sebelumnya	Invalid
D-A 16	KP-A 16	Data mahasiswa dihapus	Dihapus	Valid
D-A 17	KP-A 17	Data mahasiswa tidak dihapus	Tidak dihapus	Invalid
D-A 18	KP-A 18	Data tutor dihapus	Dihapus	Valid
D-A 19	KP-A 19	Data tutor tidak dihapus	Tidak dihapus	Invalid
D-A 20	KP-A 20	Input data nama kategori	Mengisi <i>field</i> nama kategori dengan “Pemrograman <i>Mobile</i> ”	Valid
D-A 21	KP-A 21	<i>Field</i> nama kategori dibiarkan kosong	<i>Field</i> nama kategori tidak diisi	Invalid
D-A 22	KP-A 22	Input data nama kategori	Mengubah nama kategori dengan memperbarui <i>field</i> menjadi “Pemrograman <i>Mobile 2</i> ”	Valid
D-A 23	KP-A 23	<i>Field</i> nama kategori dibiarkan kosong	<i>Field</i> nama kategori materi tidak diisi	Invalid
D-A 24	KP-A 24	Data kategori materi dihapus	Dihapus	Valid

D-A 25	KP-A 25	Data kategori materi tidak dihapus	Tidak dihapus	Invalid
D-A 26	KP-A 26	Data kritik dan saran dihapus	Dihapus	Valid
D-A 27	KP-A 27	Data kritik dan saran tidak dihapus	Tidak dihapus	Invalid
D-A 28	KP-A 28	Input <i>password</i> lama yang tidak sesuai	Memasukkan data pada <i>field password</i> lama yang tidak sesuai dengan data yang ada “admin999”	Invalid
D-A 29	KP-A 29	Input <i>password</i> baru kurang dari delapan karakter	Memasukkan <i>password</i> baru pada <i>field password</i> baru < 8 karakter yaitu “admin”	Invalid
D-A 30	KP-A 30	Input <i>password</i> lama yang sesuai dan <i>password</i> baru lebih dari delapan karakter	Memasukkan data pada <i>field password</i> lama yang sesuai yaitu “admin1234” dan <i>password</i> baru > 8 karakter yaitu “admin12345678”	Valid
D-A 31	KP-A 31	Data calon tutor diverifikasi	Diverifikasi	Valid
D-A 32	KP-A 32	Data calon tutor tidak diverifikasi	Tidak diverifikasi	Valid

3.4.1.5.2 Data Uji Calon Tutor

Pada Tabel 3.37 di bawah ini akan dijelaskan data uji dari setiap partisi uji pada tahap sebelumnya pada tutor.

Tabel 3. 37 Data Uji Calon Tutor

Kode Data Uji	Kode Partisi	Field	Data Uji	Tipe Uji
D-CT 01	KP-CT 01	Input NIM yang telah terdaftar sebagai mahasiswa dan mengisi semua <i>field</i> pada <i>form</i>	Memasukkan NIM dengan “1831710049”, memilih kategori materi “Dasar Pemrograman”, dan mengunggah <i>file</i> surat pernyataan kesanggupan “1831710049_Mey Ndita Nuraini.pdf” dengan ukuran < 1 Mb	Valid
D-CT 02	KP-CT 02	Input NIM yang belum terdaftar sebagai mahasiswa dan mengisi semua <i>field</i> pada <i>form</i>	Memasukkan NIM dengan “1234567890”, memilih kategori materi “Dasar Pemrograman”, dan mengunggah <i>file</i> surat pernyataan kesanggupan “1234567890_Mey Ndita Nuraini.pdf” dengan ukuran < 2 Mb	Invalid
D-CT 03	KP-CT 03	Input NIM yang terdaftar sebagai mahasiswa tetapi ada salah satu <i>field</i> pada <i>form</i> yang kosong	Memasukkan NIM dengan “1831710049”, memilih kategori materi “Dasar Pemrograman”	Invalid

D-CT 04	KP-CT 04	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format	Mengunggah <i>file</i> surat pernyataan kesanggupan dengan format docx (1831710049_Mey Ndita Nuraini.docx)	Valid
D-CT 05	KP-CT 05	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimal (2 Mb)	Mengunggah <i>file</i> surat pernyataan kesanggupan dengan ukuran 8 Mb	Invalid
D-CT 06	KP-CT 06	Input NIM yang tidak terdaftar sebagai mahasiswa	Mengisi NIM dengan "1234567890"	Valid
D-CT 07	KP-CT 07	Input NIM yang tidak mendaftar sebagai tutor	Mengisi NIM dengan "1831710139"	Invalid
D-CT 08	KP-CT 08	Input NIM yang mendaftar sebagai tutor yang belum diverifikasi	Mengisi NIM dengan "1831710049"	Valid
D-CT 09	KP-CT 09	Input NIM yang mendaftar sebagai tutor yang telah diverifikasi	Mengisi NIM dengan "183171055"	Valid

3.4.1.5.3 Data Uji Tutor

Pada Tabel 3.38 di bawah ini akan dijelaskan data uji dari setiap partisi uji pada tahap sebelumnya pada tutor.

Tabel 3. 38 Data Uji Tutor

Kode Data Uji	Kode Partisi	Field	Data Uji	Tipe Uji
D-T 01	KP-T 01	Input data di semua <i>field</i> pada <i>form</i>	Memasukkan data pada <i>field</i> nama materi dengan "Sintaks Pemilihan",	Valid

			deskripsi “Sintaks pemilihan merupakan sebuah <i>statement</i> yang digunakan untuk mengatur tindakan yang harus dilakukan oleh komputer pada kondisi-kondisi tertentu”, <i>requirement</i> dengan “PC/Laptop, JDK, dan Sublime Text”, serta mengunggah gambar berekstensi .jpg yang berukuran < 2Mb	
D-T 02	KP-T 02	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i>	Memasukkan data pada <i>field</i> nama materi dengan “Sintaks Pemilihan”, deskripsi “Sintaks pemilihan merupakan sebuah <i>statement</i> yang digunakan untuk mengatur tindakan yang harus dilakukan oleh komputer pada kondisi-kondisi tertentu”, <i>requirement</i> dengan “PC/Laptop, JDK, dan Sublime Text”	Invalid
D-T 03	KP-T 03	Mengunggah gambar dengan ekstensi tidak sesuai format (jpg/jpeg/png)	Mengunggah gambar dengan ekstensi .gif	Invalid
D-T 04	KP-T 04	Mengunggah gambar dengan ukuran melebihi kapasitas maksimal (2 Mb)	Mengunggah gambar berekstensi .jpg dengan ukuran <i>file</i> 2.4 Mb	Invalid
D-T 05	KP-T 05	Input data di semua <i>field</i> pada <i>form</i>	Mengubah data pada <i>field</i> nama materi dengan “Sintaks Pemilihan 2” dan data lainnya tetap	Valid

D-T 06	KP-T 06	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i>	Menghapus data pada <i>field</i> deskripsi dan membiarkan <i>field</i> lainnya tetap terisi data sebelumnya	Invalid
D-T 07	KP-T 07	Mengunggah gambar dengan ekstensi tidak sesuai format (jpg/jpeg/png)	Mengunggah ulang gambar dengan ekstensi .gif	Invalid
D-T 08	KP-T 08	Mengunggah gambar dengan ukuran melebihi kapasitas maksimal (2 Mb)	Mengunggah ulang gambar dengan ekstensi .jpg dan ukuran 2.44 Mb	Invalid
D-T 09	KP-T 09	Data materi dihapus	Dihapus	Valid
D-T 10	KP-T 10	Data materi tidak dihapus	Tidak dihapus	Invalid
D-T 11	KP-T 11	Input data di semua <i>field</i> pada <i>form</i> dengan benar	Memasukkan data pada <i>field</i> nama konten “ <i>If-Else</i> ”, link video https://youtu.be/wIZEDYIO7cM , soal latihan “coba praktikkan soal yang ada di video”, dan mengunggah <i>file</i> pendukung dengan format pdf dan berukuran < 2 Mb	Valid
D-T 12	KP-T 12	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> (kecuali <i>field</i> “ <i>file</i> pendukung” karena bersifat opsional)	Mengisi <i>field</i> nama konten dengan “Sintaks Pemilihan 2” dan membiarkan <i>field</i> lain kosong	Invalid

D-T 13	KP-T 13	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format (pdf)	Mengunggah <i>file</i> tidak sesuai dengan format “Modul_If-Else.docx”	Invalid
D-T 14	KP-T 14	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimum (2 Mb)	Mengunggah <i>file</i> yang berekstensi .pdf dengan ukuran > 2 Mb	Invalid
D-T 15	KP-T 15	Input data di semua <i>field</i> pada <i>form</i> dengan benar	Mengubah <i>field</i> nama konten dengan “Sintaks Pemilihan <i>If-Else</i> ” dan membiarkan <i>field</i> lain tetap terisi data sebelumnya	Valid
D-T 16	KP-T 16	Menyisakan satu atau lebih <i>field</i> kosong pada <i>form</i> (kecuali <i>field</i> “ <i>file</i> pendukung” karena bersifat opsional)	Menghapus data soal latihan dan membiarkan <i>field</i> lain tetap terisi data sebelumnya	Invalid
D-T 17	KP-T 17	Mengunggah <i>file</i> dengan ekstensi tidak sesuai format (pdf)	Mengunggah <i>file</i> tidak sesuai dengan format “Modul_If-Else.docx”	Invalid
D-T 18	KP-T 18	Mengunggah <i>file</i> dengan ukuran melebihi kapasitas maksimum (2 Mb)	Mengunggah <i>file</i> yang berekstensi .pdf dengan ukuran > 2 Mb	Invalid
D-T 19	KP-T 19	Konten dihapus	Dihapus	Valid
D-T 20	KP-T 20	Konten tidak dihapus	Tidak dihapus	Invalid

D-T 21	KP-T 21	Tugas diverifikasi	Diverifikasi	Valid
D-T 22	KP-T 22	Input data revisi pada <i>form</i>	Mengisi <i>form</i> revisi tugas dengan memasukkan teks “Pada <i>line</i> 7 terdapat kesalahan kurangnya tanda titik koma”, alamat URL yang mengarah pada <i>line</i> yang dituju “https://github.com/shevaputriw/Tugas_if_else/blob/a485f3ea8c7e41bd4f493a994530e72750bc514/Umur.java#L9”, dan gambar	Valid
D-T 23	KP-T 23	<i>Field</i> revisi tidak diisi	<i>Field</i> pemberian revisi kosong	Invalid
D-T 24	KP-T 24	Mengisi <i>field</i> pesan dengan karakter < 2000	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid
D-T 25	KP-T 25	Mengisi <i>field</i> pesan dengan karakter > 2000	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid
D-T 26	KP-T 26	<i>Field</i> pesan tidak diisi	<i>Field</i> pesan dibiarkan kosong	Invalid
D-T 27	KP-T 27	Input data di <i>field</i> balas forum	Mengisi <i>field</i> pada balas forum dengan “Menggunakan CSS dengan pada <i>style</i> dengan menuliskan warna pada <i>color</i> ”	Valid
D-T 28	KP-T 28	<i>Field</i> balas forum tidak diisi	<i>Field</i> balas forum tidak diisi	Invalid
D-T 29	KP-T 29	Input data di semua <i>field</i>	Mengisi <i>field</i> alamat Github dengan “https://github.com/meyndita” dan	Valid

			mengunggah <i>file</i> gambar dengan ekstensi .jpg dan berukuran < 2Mb	
D-T 30	KP-T 30	Mengunggah gambar dengan ekstensi tidak sesuai dengan format (jpg/png/jpeg)	Mengunggah <i>file</i> gambar dengan ekstensi .gif	Invalid
D-T 31	KP-T 31	Mengunggah gambar dengan ukuran melebihi kapasitas maksimum (2 Mb)	Mengunggah <i>file</i> gambar dengan ekstensi .jpg dan ukuran <i>file</i> 2.4 Mb	Invalid
D-T 32	KP-T 32	Input data di semua <i>field</i> pada <i>form</i>	Mengisi <i>field</i> subjek dengan “Performa” dan <i>field</i> kritik saran dengan “Performa bagus”	Valid
D-T 33	KP-T 33	Menyisakan satu atau lebih <i>field</i> tidak terisi	Mengisi <i>field</i> subjek dengan “Performa” dan membiarkan <i>field</i> lain kosong	Invalid

3.4.1.5.4 Data Uji Mahasiswa

Pada Tabel 3.39 di bawah ini akan dijelaskan data uji dari setiap partisi uji pada tahap sebelumnya pada mahasiswa.

Tabel 3. 39 Data Uji Mahasiswa

Kode Data Uji	Kode Partisi	Field	Data Uji	Tipe Uji
D-M 01	KP-M 01	Input <i>link</i> pada <i>field</i> pengumpulan tugas	Mengisi <i>field</i> alamat Github dengan <i>link</i> alamat Github dari tugas yang dikerjakan “https://github.com/triardyansyah/Tugas_if_else/blob/master/Umur.java”	Valid

D-M 02	KP-M 02	<i>Field</i> pengumpulan tugas tidak diisi	<i>Field</i> alamat Github pada pengumpulan tugas tidak diisi	Invalid
D-M 03	KP-M 03	Input <i>link</i> alamat Github pada <i>field</i> “Alamat Github”	Mengisi <i>field</i> alamat Github mahasiswa dengan “https://github.com/triar/”	Valid
D-M 04	KP-M 04	<i>Field</i> alamat Github tidak diisi	<i>Field</i> alamat Github tidak diisi	Valid
D-M 05	KP-M 05	Input data di <i>field</i> balas forum	Mengisi <i>field</i> pada balas forum dengan “Menggunakan CSS dengan pada <i>style</i> dengan menuliskan warna pada <i>color</i> ”	Valid
D-M 06	KP-M 06	<i>Field</i> balas forum tidak diisi	<i>Field</i> balas forum tidak diisi	Invalid
D-M 07	KP-M 07	Input data di semua <i>field</i> tanya forum	Memilih kategori forum, dan mengisi <i>field</i> pertanyaan forum dengan teks, <i>link</i> ataupun gambar	Valid
D-M 08	KP-M 08	Menyisakan satu atau lebih <i>field</i> tidak diisi	Memilih kategori forum dan <i>field</i> lainnya dibiarkan kosong	Invalid
D-M 09	KP-M 09	Input data di semua <i>field</i> pada <i>form</i>	Mengisi <i>field</i> subjek dengan “Tampilan” dan <i>field</i> kritik saran dengan “Tampilan menarik”	Valid
D-M 10	KP-M 10	Menyisakan satu atau lebih <i>field</i> tidak terisi	Mengisi <i>field</i> subjek dengan “Tampilan” dan membiarkan <i>field</i> lain kosong	Invalid
D-M 11	KP-M 11	Mengisi <i>field</i> pesan dengan karakter < 2000	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid

D-M 12	KP-M 12	Mengisi <i>field</i> pesan dengan karakter > 2000	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid
D-M 13	KP-M 13	<i>Field</i> pesan tidak diisi	<i>Field</i> pesan dibiarkan kosong	Invalid
D-M 14	KP-M 14	Input kata kunci pencarian materi yang tersedia	Memasukkan kata kunci "Sintaks"	Valid
D-M 15	KP-M 15	Input kata kunci pencarian materi yang tidak tersedia	Memasukkan kata kunci "JavaScript"	Valid
D-M 16	KP-M 16	<i>Field</i> pada <i>form</i> pencarian materi tidak diisi	<i>Field</i> pada <i>form</i> pencarian tidak diisi	Invalid
D-M 17	KP-M 17	Memilih kategori tutor dan mengisi nama tutor yang dicari	Memilih kategori "Pemrograman Web" dan mengisi nama tutor "Nadia"	Valid
D-M 18	KP-M 18	Memilih kategori tutor tapi tidak menuliskan nama tutor yang dicari	Memilih kategori "Pemrograman Web" dan <i>field</i> nama tutor tidak diisi	Valid
D-M 19	KP-M 19	Menuliskan nama tutor yang dicari tetapi tidak memilih kategori tutor	Menuliskan nama tutor dan tidak memilih kategori tutor	Valid
D-M 20	KP-M 20	Pilihan kategori tutor dan kolom isian nama tutor tidak diisi	Semua <i>field</i> pada <i>form</i> pencarian data tutor tidak diisi	Invalid

3.3.1.6 Kasus Uji

Tahap ini merupakan tahap kelima dari teknik pengujian *equivalence partitioning*. Pada tahap ini akan ditentukan kasus uji yang akan diujikan berdasarkan data uji sebelumnya. Kasus uji akan menjadi acuan saat melakukan tahap pengujian.

3.4.1.6.1 Kasus Uji Admin

Pada Tabel 3.40 di bawah ini akan dijelaskan kasus uji dari setiap data uji pada tahap sebelumnya pada admin.

Tabel 3. 40 Kasus Uji Admin

Kode Kasus Uji	Kode Data Uji	Data Uji	Tipe Uji	Output
KU-A 01	D-A 01	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “OK” maka sistem akan menampilkan pesan bahwa forum berhasil dihapus dan data akan hilang dari tabel forum
KU-A 02	D-A 02	Tidak dihapus	Invalid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “Cancel” maka sistem akan kembali ke halaman data forum
KU-A 03	D-A 03	Memasukkan NIM dengan “abcdef”	Invalid	Sistem menampilkan pesan <i>alert</i> “NIM harus angka”
KU-A 04	D-A 04	Memasukkan NIM dengan “1831710062”	Valid	Tidak terjadi pesan kesalahan dan dapat melanjutkan mengisi <i>form</i>

KU-A 05	D-A 05	Memasukkan NIM dengan “12345678910”	Invalid	Sistem menampilkan pesan <i>alert</i> kesalahan “NIM maksimal 10 karakter”
KU-A 06	D-A 06	Mengisi <i>field</i> NIM dengan “1831710062”, nama “Redika Meima Mahdalena”, program studi memilih “Manajemen Informatika”, dan tahun masuk “2018”	Valid	Data akan berhasil masuk ke dalam <i>database</i> dan sistem menampilkan pesan <i>alert</i> “Data Mahasiswa berhasil ditambahkan”. Kemudian data tampil pada tabel mahasiswa
KU-A 07	D-A 07	Mengisi <i>field</i> NIM dengan “1841710001”, nama “Syakilla Maharani Augusta”, dan program studi memilih “Teknik Informatika”	Invalid	Karena <i>field</i> tahun masuk kosong, maka sistem akan menampilkan pesan kesalahan “ <i>The tahun masuk is required</i> ”
KU-A 08	D-A 08	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data Mahasiswa.xls” dengan ukuran <i>file</i> “19 Kb”	Valid	Data dalam <i>file</i> tersebut akan berhasil masuk ke dalam <i>database</i> dan sistem akan menampilkan data tersebut dalam tabel mahasiswa
KU-A 09	D-A 09	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data Mahasiswa.pdf”	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select pdf file</i> ”
KU-A 10	D-A 10	Melakukan <i>import</i> data mahasiswa dengan <i>file</i> “Import Data	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”

		Mahasiswa.xls” dengan ukuran <i>file</i> 5 Mb.		
KU-A 11	D-A 11	Mengubah data NIM dengan bukan angka “abcderfd”	Invalid	Sistem menampilkan pesan <i>alert</i> “NIM harus angka”
KU-A 12	D-A 12	Mengubah data NIM dengan rentang 1-10 karakter “1831710001”	Valid	Tidak terjadi pesan kesalahan dan dapat melanjutkan mengisi <i>form</i>
KU-A 13	D-A 13	Mengisi data NIM lebih 10 karakter “12345678900”	Invalid	Sistem menampilkan pesan <i>alert</i> kesalahan “NIM maksimal 10 karakter”
KU-A 14	D-A 14	Mengubah <i>field</i> nama dengan “Redika” dan membiarkan <i>field</i> lainnya terisi data sebelumnya	Valid	Data akan berhasil masuk ke dalam <i>database</i> dan sistem menampilkan pesan <i>alert</i> “Data Mahasiswa berhasil <i>diedit</i> ”. Dan akan tampil pada tabel mahasiswa dengan data terbaru
KU-A 15	D-A 15	Menghapus data nama dan membiarkan <i>field</i> lainnya terisi data sebelumnya	Invalid	Karena <i>field</i> nama kosong, maka sistem akan menampilkan pesan kesalahan “ <i>The nama is required</i> ”
KU-A 16	D-A 16	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan pesan bahwa data mahasiswa berhasil dihapus dan data akan hilang dari tabel mahasiswa

KU-A 17	D-A 17	Tidak dihapus	Invalid	Sistem Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “ <i>Cancel</i> ” maka sistem akan kembali ke halaman tabel data mahasiswa
KU-A 18	D-A 18	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan pesan bahwa data tutor berhasil dihapus dan data akan hilang dari tabel tutor
KU-A 19	D-A 19	Tidak dihapus	Invalid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “ <i>Cancel</i> ” maka sistem akan kembali ke halaman tabel data tutor
KU-A 20	D-A 20	Mengisi <i>field</i> nama kategori dengan “Pemrograman <i>Mobile</i> ”	Valid	Data akan berhasil masuk ke dalam <i>database</i> dan sistem menampilkan pesan <i>alert</i> “Kategori materi berhasil ditambahkan”. Kemudian data tersebut tampil di tabel kategori materi
KU-A 21	D-A 21	<i>Field</i> nama kategori tidak diisi	Invalid	Karena <i>field</i> nama kategori materi kosong, maka sistem akan

				menampilkan pesan kesalahan “ <i>The nama kategori is required</i> ”
KU-A 22	D-A 22	Mengubah nama kategori dengan memperbarui <i>field</i> menjadi “Pemrograman <i>Mobile 2</i> ”	Valid	Data akan berhasil masuk ke dalam <i>database</i> dan sistem menampilkan pesan <i>alert</i> “Kategori materi berhasil <i>diedit</i> ”. Kemudian data terbaru akan tampil di tabel kategori materi
KU-A 23	D-A 23	<i>Field</i> nama kategori materi tidak diisi	Invalid	Karena <i>field</i> nama kategori materi kosong, maka sistem akan menampilkan pesan kesalahan “ <i>The nama kategori is required</i> ”
KU-A 24	D-A 24	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan pesan bahwa kategori materi berhasil dihapus dan data akan hilang dalam tabel kategori materi
KU-A 25	D-A 25	Tidak dihapus	Invalid	Sistem Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “ <i>Cancel</i> ” maka sistem akan kembali ke halaman tabel data kategori materi
KU-A 26	D-A 26	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan

				pesan bahwa kritik dan saran berhasil dihapus dan data akan hilang dari tabel kritik dan saran
KU-A 27	D-A 27	Tidak dihapus	Invalid	Sistem Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “Cancel” maka sistem akan kembali ke halaman tabel data kritik dan saran
KU-A 28	D-A 28	Memasukkan data pada <i>field password</i> lama yang tidak sesuai dengan data yang ada “admin999”	Invalid	Sistem akan menampilkan pesan kesalahan “Password Tidak Sesuai”
KU-A 29	D-A 29	Memasukkan <i>password</i> baru pada <i>field password</i> baru < 8 karakter yaitu “admin”	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “Password minimal memiliki 8 karakter”
KU-A 30	D-A 30	Memasukkan data pada <i>field password</i> lama yang sesuai yaitu “admin1234” dan <i>password</i> baru > 8 karakter yaitu “admin12345678”	Valid	<i>Password</i> baru akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan <i>alert</i> “Password berhasil diubah”
KU-A 31	D-A 31	Diverifikasi	Valid	Sistem akan menampilkan pesan “Tutor berhasil diterima”
KU-A 32	D-A 32	Tidak diverifikasi	Valid	Sistem akan menampilkan pesan “Tutor berhasil ditolak”

3.4.1.6.2 Kasus Uji Calon Tutor

Pada Tabel 3.41 di bawah ini akan dijelaskan kasus uji dari setiap data uji pada tahap sebelumnya pada calon tutor.

Tabel 3. 41 Kasus Uji Calon Tutor

Kode Kasus Uji	Kode Data Uji	Data Uji	Tipe Uji	Output
KU-CT 01	D-CT 01	Memasukkan NIM dengan “1831710049”, memilih kategori materi “Dasar Pemrograman”, dan mengunggah <i>file</i> surat pernyataan kesanggupan “1831710049_Mey Ndita Nuraini.pdf” dengan ukuran < 1 Mb	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan “Anda berhasil mendaftar”. Kemudian data akan tampil di halaman admin pada tabel data tutor yang belum diverifikasi
KU-CT 02	D-CT 02	Memasukkan NIM dengan “1234567890”, memilih kategori materi “Dasar Pemrograman”, dan mengunggah <i>file</i> surat pernyataan kesanggupan “1234567890_Mey Ndita Nuraini.pdf” dengan ukuran < 2 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “NIM tidak terdaftar”

KU-CT 03	D-CT 03	Memasukkan NIM dengan “1831710049”, memilih kategori materi “Dasar Pemrograman”	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please choose a file to upload</i> ”
KU-CT 04	D-CT 04	Mengunggah <i>file</i> surat pernyataan kesanggupan dengan format docx (1831710049_Mey Ndita Nuraini.docx)	Valid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select pdf file</i> ”
KU-CT 05	D-CT 05	Mengunggah <i>file</i> surat kesanggupan dengan ukuran 8 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-CT 06	D-CT 06	Mengisi NIM dengan “1234567890”	Valid	Sistem akan menampilkan pesan kesalahan bahwa “Masukkan NIM dengan benar”
KU-CT 07	D-CT 07	Mengisi NIM dengan “1831710139”	Invalid	Sistem akan menampilkan pesan bahwa “Anda gagal atau belum mendaftar sebagai tutor”
KU-CT 08	D-CT 08	Mengisi NIM dengan “1831710049”	Valid	Sistem akan menampilkan halaman yang memuat informasi bahwa proses pendaftaran masih diproses
KU-CT 09	D-CT 09	Mengisi NIM dengan “183171055”	Valid	Sistem akan menampilkan halaman yang memuat informasi bahwa Anda telah diterima sebagai tutor

3.4.1.6.3 Kasus Uji Tutor

Pada Tabel 3.42 di bawah ini akan dijelaskan kasus uji dari setiap data uji pada tahap sebelumnya pada tutor.

Tabel 3. 42 Kasus Uji Tutor

Kode Kasus Uji	Kode Data Uji	Data Uji	Tipe Uji	Output
KU-T 01	D-T 01	Memasukkan data pada <i>field</i> nama materi dengan “Sintaks Pemilihan”, deskripsi “Sintaks pemilihan merupakan sebuah <i>statement</i> yang digunakan untuk mengatur tindakan yang harus dilakukan oleh komputer pada kondisi-kondisi tertentu”, <i>requirement</i> dengan “PC/Laptop, JDK, dan Sublime Text”, serta mengunggah gambar berekstensi .jpg yang berukuran < 2Mb	Valid	Data berhasil disimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Materi berhasil ditambahkan”. Kemudian akan tampil pada tabel data materi
KU-T 02	D-T 02	Memasukkan data pada <i>field</i> nama materi dengan “Sintaks Pemilihan”, deskripsi “Sintaks pemilihan merupakan sebuah <i>statement</i> yang digunakan untuk mengatur tindakan yang harus dilakukan oleh komputer pada kondisi-kondisi tertentu”, <i>requirement</i> dengan	Invalid	Karena <i>field cover</i> materi kosong, maka sistem akan menampilkan pesan kesalahan “ <i>Please choose file to upload</i> ”

		“PC/Laptop, JDK, dan Sublime Text”		
KU-T 03	D-T 03	Mengunggah gambar dengan ekstensi .gif	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select jpg/png/jpeg file</i> ”
KU-T 04	D-T 04	Mengunggah gambar berekstensi .jpg dengan ukuran file 2.4 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-T 05	D-T 05	Mengubah data pada <i>field</i> nama materi dengan “Sintaks Pemilihan 2” dan data lainnya tetap	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan “Materi berhasil <i>diedit</i> ”. Kemudian data terbaru akan tampil di tabel kategori materi
KU-T 06	D-T 06	Menghapus data pada <i>field</i> deskripsi dan membiarkan <i>field</i> lainnya tetap terisi data sebelumnya	Invalid	Karena <i>field</i> deskripsi kosong maka sistem akan menampilkan pesan kesalahan “ <i>The deskripsi is required</i> ”
KU-T 07	D-T 07	Mengunggah ulang gambar dengan ekstensi .gif	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select jpg/png/jpeg file</i> ”

KU-T 08	D-T 08	Mengunggah ulang gambar dengan ekstensi .jpg dan ukuran 2.44 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-T 09	D-T 09	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan pesan bahwa materi berhasil dihapus dan data akan terhapus dari tabel materi
KU-T 10	D-T 10	Tidak dihapus	Invalid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “ <i>Cancel</i> ” maka sistem akan kembali ke data materi
KU-T 11	D-T 11	Memasukkan data pada <i>field</i> nama konten “ <i>If-Else</i> ”, link video https://youtu.be/wIZEDYIO7c M, soal latihan “coba praktikkan soal yang ada di video”, dan mengunggah <i>file</i>	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Konten berhasil ditambahkan”. Kemudian data akan tampil pada tabel konten

		pendukung dengan format pdf dan berukuran < 2 Mb		
KU-T 12	D-T 12	Mengisi <i>field</i> nama konten dengan “Sintaks Pemilihan 2” dan membiarkan <i>field</i> lain kosong	Invalid	Karena <i>field link</i> video, soal latihan kosong maka sistem menampilkan pesan kesalahan “ <i>The soal is required, the video is required</i> ”
KU-T 13	D-T 13	Mengunggah <i>file</i> tidak sesuai dengan format “Modul_If-Else.docx”	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select pdf file</i> ”
KU-T 14	D-T 14	Mengunggah <i>file</i> yang berekstensi .pdf dengan ukuran > 2 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-T 15	D-T 15	Mengubah <i>field</i> nama konten dengan “Sintaks Pemilihan <i>If-Else</i> ” dan membiarkan <i>field</i> lain tetap terisi data sebelumnya	Valid	Data akan tersimpan di <i>database</i> dan sistem akan menampilkan pesan bahwa “Konten berhasil <i>diedit</i> ”. Kemudian data terbaru akan tampil di tabel kategori materi
KU-T 16	D-T 16	Menghapus data soal latihan dan membiarkan <i>field</i> lain tetap terisi data sebelumnya	Invalid	Karena <i>field</i> soal latihan kosong maka sistem menampilkan pesan kesalahan “ <i>The soal is required</i> ”

KU-T 17	D-T 17	Mengunggah <i>file</i> tidak sesuai dengan format “Modul_If-Else.docx”	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select pdf file</i> ”
KU-T 18	D-T 18	Mengunggah <i>file</i> yang berekstensi .pdf dengan ukuran > 2 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-T 19	D-T 19	Dihapus	Valid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika OK maka sistem akan menampilkan pesan bahwa konten berhasil dihapus dan data akan hilang dari tabel konten
KU-T 20	D-T 20	Tidak dihapus	Invalid	Sistem menampilkan pesan peringatan “Apakah Anda yakin akan menghapus data?” jika “ <i>Cancel</i> ” maka sistem akan kembali ke data konten
KU-T 21	D-T 21	Diverifikasi	Valid	Sistem akan menampilkan pesan “Tugas Berhasil Diverifikasi”

KU-T 22	D-T 22	Mengisi <i>form</i> revisi tugas dengan memasukkan teks “Pada <i>line</i> 7 terdapat kesalahan kurangnya tanda titik koma”, alamat URL yang mengarah pada <i>line</i> yang dituju “ https://github.com/shevaputriw/Tugas_if_else/blob/a485f3ea8c7e41bdd4f493a994530e72750bc514/Umur.java#L9 ”, dan gambar	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Revisi berhasil ditambahkan”. Pesan revisi akan tampil pada tabel revisi tugas
KU-T 23	D-T 23	<i>Field</i> pemberian revisi kosong	Invalid	Karena <i>field</i> revisi kosong maka sistem menampilkan pesan kesalahan “ <i>The revisi is required</i> ”
KU-T 24	D-T 24	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid	Data tersimpan pada <i>database</i> dan akan tampil di halaman <i>chatbox</i>
KU-T 25	D-T 25	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid	Sistem akan membatasi penginputan pesan
KU-T 26	D-T 26	<i>Field</i> pesan dibiarkan kosong	Invalid	Karena <i>field</i> pesan kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”
KU-T 27	D-T 27	Mengisi <i>field</i> pada balas forum dengan “Menggunakan CSS	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan

		pada <i>style</i> dengan menuliskan warna pada <i>color</i> ”		pesan bahwa “Jawaban Anda berhasil dikirim”. Kemudian jawaban tampil pada halaman forum
KU-T 28	D-T 28	<i>Field</i> balas forum tidak diisi	Invalid	Karena <i>field</i> balas forum kosong maka sistem menampilkan pesan kesalahan “ <i>The message is required</i> ”
KU-T 29	D-T 29	Mengisi <i>field</i> alamat Github dengan “https://github.com/meyndita/” dan mengunggah <i>file</i> gambar dengan ekstensi .jpg dan berukuran < 2Mb	Valid	Data berhasil disimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Profil berhasil diedit”. Data profil terbaru akan tampil pada halaman profil
KU-T 30	D-T 30	Mengunggah <i>file</i> gambar dengan ekstensi .gif	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>Please select jpg/jpeg/png file</i> ”
KU-T 31	D-T 31	Mengunggah <i>file</i> gambar dengan ekstensi .jpg dan ukuran <i>file</i> 2.4 Mb	Invalid	Sistem akan menampilkan pesan kesalahan bahwa “ <i>The file size should not exceed 2 MB</i> ”
KU-T 32	D-T 32	Mengisi <i>field</i> subjek dengan “Performa” dan <i>field</i> kritik saran dengan “Performa bagus”	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan

				pesan bahwa “Kritik dan Saran berhasil dikirim”
KU-T 33	D-T 33	Mengisi <i>field</i> subjek dengan “Performa” dan membiarkan <i>field</i> lain kosong	Invalid	Karena <i>field</i> kritik_saran kosong maka sistem menampilkan pesan kesalahan “ <i>The Kritik Saran field is required</i> ”

3.4.1.6.4 Kasus Uji Mahasiswa

Pada Tabel 3.43 di bawah ini akan dijelaskan kasus uji dari setiap data uji pada tahap sebelumnya pada mahasiswa.

Tabel 3. 43 Kasus Uji Mahasiswa

Kode Kasus Uji	Kode Data Uji	Data Uji	Tipe Uji	Output
KU-M 01	D-M 01	Mengisi <i>field</i> alamat Github dengan <i>link</i> alamat Github dari tugas yang dikerjakan “ https://github.com/triaryansyah/Tugas_if_else/blob/master/Umur.java ”	Valid	Data disimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Tugas berhasil dikirim! Jawaban akan segera diperiksa”
KU-M 02	D-M 02	<i>Field</i> alamat Github pada pengumpulan tugas tidak diisi	Invalid	Karena <i>field link</i> pengumpulan tugas kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”

KU-M 03	D-M 03	Mengisi <i>field</i> alamat Github mahasiswa dengan “https://github.com/triar”	Valid	Data akan tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Profil berhasil diedit”. Data terbaru akan tampil di halaman profil
KU-M 04	D-M 04	<i>Field</i> alamat Github tidak diisi	Valid	Karena <i>field</i> Github kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”
KU-M 05	D-M 05	Mengisi <i>field</i> pada balas forum dengan “Menggunakan CSS dengan pada <i>style</i> dengan menuliskan warna pada <i>color</i> ”	Valid	Data tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Jawaban Anda berhasil dikirim” dan tampil di halaman detail forum
KU-M 06	D-M 06	<i>Field</i> balas forum tidak diisi	Invalid	Karena <i>field</i> balas forum kosong maka sistem menampilkan pesan kesalahan “ <i>The message is required</i> ”
KU-M 07	D-M 07	Memilih kategori forum, topik, dan mengisi <i>field</i> pertanyaan forum dengan teks, <i>link</i> ataupun gambar	Valid	Data tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Forum berhasil dibuat!”

KU-M 08	D-M 08	Memilih kategori forum dan <i>field</i> lainnya dibiarkan kosong	Invalid	Karena <i>field</i> pertanyaan dan topik kosong maka sistem menampilkan pesan kesalahan “ <i>The Pertanyaan is required</i> ” dan “ <i>The Topik is required</i> ”
KU-M 09	D-M 09	Mengisi <i>field</i> subjek dengan “Tampilan” dan <i>field</i> kritik saran dengan “Tampilan menarik”	Valid	Data tersimpan dalam <i>database</i> dan sistem akan menampilkan pesan bahwa “Kritik dan Saran berhasil dikirim”
KU-M 10	D-M 10	Mengisi <i>field</i> subjek dengan “Tampilan” dan membiarkan <i>field</i> lain kosong	Invalid	Karena <i>field</i> kritik_saran kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”
KU-M 11	D-M 11	Mengisi <i>field</i> pesan dengan karakter < 2000	Valid	Data tersimpan pada <i>database</i> dan akan tampil di halaman <i>chatbox</i>
KU-M 12	D-M 12	Mengisi <i>field</i> pesan dengan karakter > 2000	Invalid	Sistem akan membatasi penginputan pesan
KU-M 13	D-M 13	Field pesan dibiarkan kosong	Invalid	Karena <i>field</i> pesan kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”
KU-M 14	D-M 14	Memasukkan kata kunci “Sintaks”	Valid	Sistem menampilkan hasil pencarian materi

				yang memiliki kata “Sintaks” di dalamnya
KU-M 15	D-M 15	Memasukkan kata kunci “JavaScript”	Valid	Sistem menampilkan pesan bahwa materi dengan kata kunci tersebut tidak tersedia
KU-M 16	D-M 16	<i>Field</i> pada <i>form</i> pencarian tidak diisi	Invalid	Karena <i>field</i> pencarian kosong maka sistem menampilkan pesan kesalahan “ <i>Please fill out this field</i> ”
KU-M 17	D-M 17	Memilih kategori “Pemrograman Web” dan mengisi nama tutor “Nadia”	Valid	Sistem akan menampilkan data tutor sesuai dengan kategori dan kata kunci yang dimasukkan
KU-M 18	D-M 18	Memilih kategori “Pemrograman Web” dan <i>field</i> nama tutor tidak diisi	Valid	Sistem akan menampilkan data tutor dengan kategori pemrograman web
KU-M 19	D-M 19	Menuliskan nama tutor “sheva” dan tidak memilih kategori tutor	Valid	Sistem akan menampilkan data tutor Sheva
KU-M 20	D-M 20	Semua <i>field</i> pada <i>form</i> pencarian data tutor tidak diisi	Invalid	Sistem akan menampilkan pesan “Salah satu kata kunci harus diisi”

3.4.2 Pengujian *Usability*

Usability adalah analisa kualitatif yang menentukan seberapa mudah pengguna menggunakan antarmuka suatu aplikasi. Suatu aplikasi disebut *usable* jika fungsi-fungsi di dalamnya dapat dijalankan secara efektif, efisien, dan memuaskan (Firmansyah, 2018).

Definisi *usability* menurut (Hadi et al., 2018) adalah sejauh mana suatu produk dapat digunakan oleh pengguna tertentu untuk mencapai target yang ditetapkan dengan efektivitas, efisiensi dan mencapai kepuasan penggunaan dalam konteks tertentu. Berdasarkan definisi tersebut (Hadi et al., 2018) juga menjelaskan bahwa *usability* diukur berdasarkan komponen:

- a. Kemudahan (*learnability*) didefinisikan seberapa cepat pengguna mahir dalam menggunakan sistem serta kemudahan dalam penggunaan menjalankan suatu fungsi serta apa yang pengguna inginkan dapat mereka dapatkan.
- b. Efisiensi (*efficiency*) didefinisikan sebagai sumber daya yang dikeluarkan guna mencapai ketepatan dan kelengkapan tujuan.
- c. Mudah diingat (*memorability*) didefinisikan bagaimana kemampuan pengguna mempertahankan pengetahuannya setelah jangka waktu tertentu, kemampuan mengingat didapatkan dari peletakkan menu yang selalu tetap.
- d. Kesalahan dan keamanan (*errors*) didefinisikan berapa banyak kesalahan-kesalahan apa saja yang dibuat pengguna, kesalahan yang dibuat pengguna mencakup ketidaksesuaian apa yang pengguna pikirkan dengan apa yang sebenarnya disajikan oleh sistem.
- e. Kepuasan (*satisfaction*) didefinisikan kebebasan dari ketidaknyamanan, dan sikap positif terhadap penggunaan produk atau ukuran subjektif sebagaimana pengguna merasa tentang penggunaan sistem.

Dalam melakukan pengujian *usability* responden berjumlah 5 (lima) orang sudah cukup untuk melakukan pengujian (Iryanto et al., 2019). Berdasarkan pernyataan tersebut penulis akan melakukan pengujian *usability* kepada Mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang untuk setiap aktor yaitu admin sebanyak 5 orang, tutor dan calon tutor 5 orang, dan mahasiswa 17 orang.

3.4.2.1 Tahapan Pengujian *Usability*

Di dalam pengujian *usability* memiliki langkah-langkah yang harus dijalankan antara lain:

1. Menentukan sistem yang akan diuji. Sistem yang akan diuji adalah Aplikasi Pembelajaran *Coding* bagi Mahasiswa Jurusan Teknologi Informasi dengan Studi Kasus di Politeknik Negeri Malang.
2. Menentukan responden. Responden akan ditujukan kepada mahasiswa aktif Jurusan Teknologi Informasi Politeknik Negeri Malang.
3. Pemberian *task* kepada responden. *Task* digunakan sebagai jalan pengguna dalam mencoba sistem secara menyeluruh untuk mendapatkan *user experience*.
4. Penyebaran kuesioner. Kuesioner dibuat dengan tiga belas pertanyaan untuk masing-masing aktor.
5. Merekap hasil kuesioner untuk mendapatkan hasil penilaian user terhadap sistem yang telah diujikan.
6. Perhitungan nilai *usability* untuk mengetahui bahwa apakah sistem yang diujikan telah berjalan secara efektif, efisien, dan memuaskan.