# RANCANG BANGUN APLIKASI

**UNTUK PENYEDIA KURSUS MENGEMUDI**

**BERBASIS WEB** **DENGAN INCREMENTAL MODEL**

**SKRIPSI**

****

**Oleh :**

**YODANIS ERLANDI SUTANTIO**

**13.2020.1.00905**

**PROGRAM STRATA-1**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI ADHI TAMA SURABAYA**

**2024**

**RANCANG BANGUN APLIKASI UNTUK PENYEDIA KURSUS MENGEMUDI BERBASIS WEB DENGAN *INCREMENTAL MODEL***

Name : Yodanis Erlandi Sutantio

Student ID : 13.2020.1.00905

Advisor : Rachman Arief, S.Kom., M.Kom.

# ABSTRAK

Seiring meningkatnya jumlah pemilik kendaraan bermotor roda empat, pihak berwenang, dalam hal ini, Dinas Perhubungan dan Satuan Kepolisian Republik Indonesia menghimbau para pengemudi baru untuk mengikuti Kursus Mengemudi sebelum mulai mengendarai mobil. Jumlah pengelola kursus mengemudi di Surabaya saat ini sudah menyentuh angka ratusan, dengan semakin banyaknya pengemudi-pengemudi baru dan persaingan antar penyedia jasa kursus mengemudi seperti saat ini, dibutuhkan sebuah Sistem Informasi untuk mengatasi potensi masalah tersebut. Dengan melakukan Rekayasa Perangkat Lunak, diharapkan penelitian ini akan menghasilkan aplikasi yang dapat membantu semua pihak yang terlibat dalam proses kursus mengemudi, baik bagi Siswa / Pelajar, Instruktur Kursus, dan Pemilik / Admin Kursus. Pengembangan Perangkat Lunak menggunakan *Incremental Model* memberikan tim pengembang kemampuan untuk berfokus terhadap kebutuhan setiap pengguna yang sudah disebutkan sebelumnya. Karakteristik iteratif yang dimiliki *incremental model* diyakini akan menghasilkan aplikasi berkualitas dan mampu menyelesaikan masalah yang dihadapi oleh pihak-pihak dalam rangkaian proses kursus mengemudi.

**Kata Kunci :** Kursus Mengemudi, Sistem Informasi, Rekayasa Perangkat Lunak, *Incremental Model*

**WEB-BASED APP DEVELOPMENT FOR DRIVING SCHOOL WITH INCREMENTAL MODEL**

*Name* : Yodanis Erlandi Sutantio

*Student ID* : 13.2020.1.00905

*Advisor* : Rachman Arief, S.Kom., M.Kom.

# ABSTRACT

As the number of four-wheeled motorized vehicle owners increases, the government authorities, in this case, the Office of Transportation and the Indonesian Police Force, urge young drivers to take a driving course before starting to drive a car. The number of driving schools in Surabaya has now reached hundreds, with the increasing number of new drivers and competition between driving schools as it is today, an Information System is needed to overcome these potential problems. By implementing Software Engineering, it is hoped that this research will produce an app that can help all parties involved in the driving course process, for students, instructors, and driving school owners / admins. Incremental Model gives the software development team the ability to focus on the previously mentioned needs of each user. It is believed that the iterative characteristics of the incremental model will produce high-quality app and be able to solve problems faced by all parties in the driving course process.

**Keywords :** Driving School, Information System, Software Engineering, Incremental Model

# DAFTAR ISI

Hahaha

# DAFTAR GAMBAR

Hahaha

# DAFTAR TABEL

Hahaha

# BAB I

**PENDAHULUAN**

## Latar Belakang

Saat ini, banyak penyedia jasa yang dapat membantu kita memahami teori dan mengajarkan kita kemampuan untuk berkendara dengan baik melalui kursus mengemudi. Sering sekali kita menemui pengemudi yang sedang belajar di jalan raya yang biasanya mereka didampingi dengan mentor/instruktur untuk memandu dan memberikan arahan. Sehingga, ketika mereka berhasil lulus dari kursus tersebut, mereka diharapkan dapat memahami aturan-aturan di jalan raya, seperti cara mengemudi melewati persimpangan atau merubah lajur berkendara dengan benar.

Sebab, melalui data yang dibagikan oleh Korps Lalu Lintas Kepolisian Republik Indonesia, pada Januari tahun 2022 hingga September 2022 terjadi setidaknya 94.617 kasus kecelakaan lalu lintas di seluruh wilayah Indonesia (Sumber : dephub.go.id). Lebih lanjut lagi Korlantas Polri menjelaskan faktor-faktor yang memiliki andil pada 94.617 kasus kecelakaan tersebut, faktor-faktor tersebut diantaranya 61% kasus kecelakaan diakibatkan oleh faktor kesalahan manusia seperti kurang terampilnya pengemudi dalam mengendalikan kendaraannya, serta kelalaian-kelalaian yang lainnya seperti mengemudi dalam kondisi mengantuk, ugal-ugalan, dan lain lain. Gambar 1.1 dibawah ini adalah persentase faktor penyebab terbesar kecelakaan lalu lintas.



Gambar 1. Faktor Penyebab Terbesar Kecelakaan Lalu Lintas

(Sumber: Dephub.go.id)

Kementerian Perhubungan bersama dengan Korlantas Polri selanjutnya bekerja sama dengan masyarakat untuk menekan angka kecelakaan yang semakin tahun semakin naik dengan diadakannya sekolah atau kursus mengemudi yang sering kita temui. Masyarakat yang ingin mendirikan kursus mengemudi selanjutnya wajib mengikuti pelatihan atau sertifikasi sebagai instruktur kursus, sehingga, tidak sembarang orang dapat mengaku sebagai instruktur kursus tanpa melalui proses tersebut. Selain syarat diatas, tentunya ada syarat-syarat yang harus dipenuhi lainnya untuk sebuah lembaga atau sekolah kursus mengemudi baru dapat dianggap sah secara hukum.

Dari data yang berhasil kami himpun, di Kota Surabaya sendiri sudah terdapat lebih dari 50 lembaga kursus mengemudi, jumlah ini belum termasuk penyedia kursus perorangan dan lembaga-lembaga kursus rintisan baru. Juga tidak menutup kemungkinan jumlah tersebut akan semakin bertambah di tahun-tahun berikutnya, sehingga, timbul kebutuhan akan sebuah aplikasi yang menjadi *platform* untuk mewadahi para penyedia jasa kursus mengemudi. Karena sudah terwadahi, para penyedia jasa kursus mengemudi dapat dengan bebas melakukan kegiatan pemasaran untuk layanan yang mereka tawarkan di dalam aplikasi. Sedangkan, bagi calon siswa *platform* ini akan menjadi gerbang utama untuk menemukan informasi lengkap tentang berbagai pilihan kursus mengemudi di Surabaya.

Sistem informasi adalah salah satu disiplin ilmu yang mempelajari tentang Rekayasa Perangkat Lunak yang merupakan sebuah disiplin untuk menerapkan prinsip-prinsip *design, development, testing,* dan *maintenance* perangkat lunak. Sebagaimana penelitian yang dilakukan oleh (Made et al., 2022) dan (Adhiva Kurnia, 2023) mengangkat permasalahan yang sama tentang proses bisnis yang belum dilakukan secara efektif dimana pelajar kursus tidak menerima informasi terbaru tentang kursus, metode registrasi yang masih konvensional membutuhkan waktu yang lama, dan kendala terhadap jadwal kursus. Terbukti dengan mengintegrasikan layanan kursus mengemudi dan teknologi informasi, dua sistem tersebut berhasil mengatasi semua masalah yang disebutkan.

Singkatnya, kami ingin masyarakat untuk memahami bahaya dan risiko yang ditemui apabila mereka mengemudikan kendaraan tanpa mengikuti suatu pelatihan untuk mengasah pemahaman dan kemampuannya terlebih dahulu tentang topik tersebut. Kami ingin menjadi wadah bagi penyedia jasa kursus mengemudi untuk memberikan edukasi ke masyarakat luas bahwa mengikuti kursus mengemudi tidak memerlukan biaya yang tinggi dan meskipun biaya yang dibutuhkan masih dirasa cukup tinggi, setidaknya, mereka mengerti betapa pentingnya untuk mengikuti kursus mengemudi sebelum berkendara langsung di jalan raya. Dengan Rekayasa Perangkat Lunak, kami ingin mengurangi beban dari penyedia kursus mengemudi konvensional yang masih kebingungan dalam melakukan kegiatan promosi untuk layanan mereka, dan kebutuhan untuk harus mencetak formulir pendaftaran, sertifikat, bukti pembayaran dan yang lainnya sehingga semuanya dapat terintegrasi di satu aplikasi, dimana hasil akhir yang ingin kami capai adalah peningkatan layanan penyedia kursus mengemudi yang lebih baik dari sebelumnya. Dengan demikian, kami tertarik untuk menulis skripsi dengan judul : “**RANCANG BANGUN APLIKASI UNTUK PENYEDIA JASA KURSUS MENGEMUDI BERBASIS WEB DENGAN *INCREMENTAL MODEL***”

## Rumusan Masalah

Dari Latar Belakang diatas, dapat disimpulkan bahwa terdapat 2 pokok permasalahan yang mendasari kami untuk membuat skripsi ini:

1. Bagaimana merancang aplikasi yang dapat membantu, memudahkan peserta dan penyedia kursus dalam melakukan rangkaian proses kursus?
2. Bagaimana mengembangkan aplikasi untuk penyedia jasa kursus mengemudi dengan menggunakan *Incremental Model* ?

## Tujuan Penelitian

Tujuan dari penelitian ini yaitu untuk menjawab rumusan masalah diatas dengan:

1. Merancang aplikasi yang memudahkan peserta dan penyedia kursus mengemudi dalam melakukan rangkaian proses kursus seperti proses administrasi, penjadwalan kursus, dan lain sebagainya.
2. Dengan mengimplementasikan *Incremental Model*, perencanaan, estimasi, dan alokasi sumber daya untuk melakukan pengembangan perangkat lunak menjadi lebih sederhana.

## Batasan Masalah

Dari Latar Belakang dan Rumusan Masalah diatas, berikut adalah batasan-batasan dalam pengembangan aplikasi:

1. Aplikasi merupakan sebuah *platform* dengan konsep *marketplace* bagi penyedia jasa kursus mengemudi.
2. *Platform* ini terbuka bagi seluruh penyedia jasa kursus mengemudi yang resmi beroperasi dan berdomisili di dalam wilayah Kota Surabaya.
3. Dengan menggunakan teknologi berbasis web, *platform* ini memberikan pengguna kemampuan untuk mengaksesnya di mana saja dan melalui berbagai perangkat seperti *smartphone*, *tablet*, laptop, dan perangkat-perangkat lain yang terhubung dengan internet.
4. Fokus utama aplikasi yang kami rancang adalah mengolah dan melacak informasi tentang kursus mengemudi, dengan demikian, proses-proses teknis (contoh: pembayaran kursus, pendampingan saat praktek kursus) akan tetap dijalankan diluar aplikasi.

## Sistematika Penulisan

Dalam dokumen proposal ini, kami nantinya akan menuliskan setidaknya 3 bab yang membahas proses awal penelitian, bab-bab tersebut diantaranya :

**BAB I PENDAHULUAN**

Pada bab ini kami menjelaskan apa yang melandasi kami melakukan penelitian ini, seperti pada umumnya, isi dari bab ini adalah latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Di bab ini kami akan memberikan pembaca pemahaman dasar tentang istilah-istilah atau teknik-teknik yang nantinya kami sering sebutkan dan sering kami gunakan dalam menyelesaikan penelitian kami.

**BAB III METODOLOGI PENELITIAN**

Bab ini bertujuan untuk menjelaskan tentang tahapan-tahapan penelitian serta pengembangan perangkat lunak dengan model *incremental*, kendala yang dihadapi, sebagian kecil dari alur sistem, dan persiapan-persiapan sebelum melanjutkan proses *development* / pemrograman.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini, hasil dan temuan-temuan dari penelitian akan dijabarkan. Hasil penelitian yang dimaksud adalah aplikasi KEMUDI dan temuan-temuan yang terjadi saat proses pengujian kelayakan aplikasi.

**BAB V KESIMPULAN**

Yang terakhir, bab ini akan membahas kesimpulan terkait penelitian yang dilakukan, dan saran-saran yang diajukan penulis untuk penelitian-penelitian serupa di masa depan.

# BAB II

**LANDASAN TEORI**



## Kursus Mengemudi

Lembaga Kursus adalah salah satu penyelenggara pendidikan diluar sekolah resmi (non-formal) untuk mengembangkan kemampuan dan keterampilan diri (Mahdy et al., 2021). Kursus Mengemudi secara spesifik dapat diartikan suatu pendidikan untuk mengembangkan kemampuan dan keterampilan diri dalam mengemudikan kendaraan khususnya mobil. KBBI sendiri mendeskripsikan kursus sebagai pelajaran tentang suatu pengetahuan atau keterampilan, yang diberikan dalam waktu singkat. Atau bisa juga diartikan sebagai lembaga di luar sekolah yang memberikan pelajaran serta pengetahuan atau keterampilan yang diberikan dalam waktu singkat.

## Sistem Informasi Berbasis Web

Sistem Informasi menurut Richard Vidgen adalah sebuah kumpulan komponen-komponen yang berinteraksi, komponen yang dimaksud adalah manusia, prosedur-prosedur, dan teknologi-teknologi yang ada, dimana komponen-komponen tersebut secara bersamaan mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengontrolan, pembuatan keputusan dan pengelolaan organisasi/perusahaan. Sistem informasi sendiri berisi informasi tentang organisasi/perusahaan terkait, contohnya, kondisi tentang operasional internal mereka, dan tentang lingkungan di dalam perusahaan tersebut, sebagai contoh informasi tentang para pelanggan, para supplier, dan kompetitor-kompetitor yang ada. Tanpa adanya sistem informasi, sebuah organisasi sulit untuk bertahan. Namun, bukan berarti bahwa sistem informasi harus menggunakan teknologi-teknologi informasi yang berbentuk komputer-komputer atau jaringan internet dan komunikasi, karena banyak sekali bentuk sistem informasi. Organisasi atau perusahaan sangat bergantung pada sistem informasi, meskipun aspek-aspek formal dari sistem informasi ini masih menggunakan sistem pengarsipan berupa kertas pada era sebelum adanya teknologi informasi itu sendiri.

Menurut (Manullang, A. H., et al., 2021), *website* pada dasarnya adalah kumpulan halaman yang saling terhubung melalui jaringan internet yang biasanya berisi bermacam-macam media, media yang dimaksud diantaranya adalah teks, gambar, audio, video, dan lain lain. Secara analogi, *website* dapat diibaratkan sebagai sebuah perpustakaan digital tanpa batas halaman. Setiap halaman dalam *website* berpotensi berisi informasi yang beragam, dan pengguna dapat dengan mudah berpindah antar halaman hanya dengan mengklik tautan.



### *Hyper Text Markup Language* (HTML)

HTML adalah salah satu istilah pemrograman yang paling dikenal oleh masyarakat umum, namun, banyak perdebatan yang mengatakan bahwa sebenarnya HTML bukanlah sebuah bahasa pemrograman. Ada juga beberapa orang yang beranggapan bahwa karena dalam menulis HTML diperlukan setidaknya pemahaman dasar tentang pemrograman, maka HTML dianggap sebagai bahasa pemrograman. Mengacu dari jurnal yang ditulis oleh (Sari et al., 2022) HTML merupakan salah satu bahasa pemrograman yang digunakan untuk membuat website. HTML biasa ditulis untuk membantu perancangan struktur dasar halaman website atau bisa juga dianggap sebagai pondasi awal untuk menyusun kerangka halaman website secara lebih terstruktur sebelum masuk ke tahap desain dan fungsionalitas.

### *Cascading Style Sheet* (CSS)

Pada era modern saat ini, hampir tidak bisa kita temui rangkaian kode HTML tanpa dilengkapi CSS. Menurut (Sari et al., 2022) CSS adalah bahasa pemrograman yang ditujukan untuk memberikan modifikasi tampilan elemen-elemen web seperti *font, outline, background,* menyesuaikan tampilan website dengan ukuran layar, dan sebagainya. Jika HTML digunakan untuk menempatkan konten-konten apa saja yang ingin ditampilkan pada sebuah halaman web, CSS digunakan untuk memberikan pemahaman kepada mesin untuk melakukan modifikasi terhadap tampilan elemen dan penataan tata letak lebih lanjut.

Karena sejatinya, HTML tidak dirancang untuk menentukan aspek visual pada sebuah desain website. Sebab, fokus utama dari HTML adalah membagi struktur sebuah halaman website. Oleh karena itu, dikenalkan skrip “pendamping” untuk memperindah tag-tag HTML yaitu CSS. Selain itu, tujuan penggunaan CSS adalah untuk memberikan kesan konsisten di seluruh website.

### Tailwind CSS *Framework* (CSS)

(Somi, M., 2023) dalam jurnalnya menjelaskan bahwa *Framework* *Tailwind* CSS adalah *framework* CSS yang mengutamakan penggunaan kelas utilitas yang paling populer dan bertujuan untuk membangun tampilan antarmuka khusus dengan cepat dan mudah. Maksudnya, berbeda dengan *Bootstrap* yang tergolong *framework* UI *kits*, *tailwind* tidak menyediakan komponen-komponen siap pakai. *Tailwind* tidak mempunyai tema bawaan. Dengan *tailwind*, kita memberikan *style* dengan mengetikkan kelas-kelas yang sudah ditentukan sebelumnya ke kodingan HTML yang kita kerjakan.

Lebih lanjut lagi, (Somi, M., 2023) menjelaskan bahwa dengan menggunakan *tailwind*, memberikan kita kemampuan untuk mempercepat proses pemrograman tanpa kita harus menulis kode CSS di *file* lain melainkan, menulisnya secara bersamaan di kode markup HTML. Selain itu, *tailwind* memberikan kita kemampuan untuk melakukan kustomisasi secara penuh sesuai keinginan kita. Efek samping dari menggunakan *tailwind* adalah kode HTML yang dihasilkan akan jauh lebih panjang

### Javascript

Halaman website yang dihasilkan dari hanya menggunakan bahasa HTML & CSS cenderung statis dan kurang menarik. Untuk membuat tampilan yang lebih dinamis diciptakan sebuah bahasa pemrograman baru demi mengatasi kekurangan ini, yakni *Javascript*. Sebagai referensi, (Noviantoro et al., 2022) menjelaskan bahwa *Javascript* adalah salah satu bahasa pemrograman yang digunakan untuk dapat berjalan di web browser. Pada awal pengembangan bahasa pemrograman ini sempat disebut dengan nama *Mocha*, kemudian berubah penamaannya menjadi *Live-Script*, dimana ketika masa rilis, diubah lagi menjadi *Javascript*. Lebih jauh lagi dijelaskan bahwa *Javascript* adalah *script* program berbasis *client* yang dieksekusi oleh browser sehingga membuat halaman web melakukan tugas-tugas tambahan yang tidak bisa dilakukan hanya dengan memanfaatkan HTML biasa. Selain alasan yang kami sebut diatas, beberapa interaksi yang ingin kami munculkan pada aplikasi ini tidak dapat diselesaikan hanya dengan menggunakan HTML dan CSS.

### jQuery

(Sahrudin A. et al., 2023) menyebutkan pada jurnalnya menjelaskan jQuery sebagai berikut. Pada tahun 2006, John Resig memperkenalkan jQuery, sebuah *library* Javascript yang revolusioner. Sebelum hadirnya jQuery, para pengembang perangkat lunak harus menulis kode Javascript yang kompatibel dengan berbagai macam *browser*. jQuery hadir sebagai terobosan, memberikan para pengembang perangkat lunak kemampuan untuk menulis kode Javascript yang ringkas, mudah dipahami, serta dapat dieksekusi di berbagai *browser*.

### MySQL

MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang populer dan banyak digunakan di berbagai platform (Hermiati R. et al., 2021). Dikenal dengan keunggulannya dalam mendukung skrip PHP, MySQL menawarkan kemudahan integrasi dengan bahasa pemrograman web yang populer ini. Salah satu keunggulan utama MySQL adalah sintaks kuerinya yang sederhana dan mudah dipahami. Bahasa *Structured Query Language* (SQL) yang digunakan MySQL dirancang dengan intuitif, memungkinkan pengguna untuk mengelola data dengan efektif dan efisien. Selain itu, MySQL terkenal dengan kecepatannya yang luar biasa dalam memproses dan mengakses data. Kecepatan ini menjadikannya pilihan ideal untuk aplikasi web yang membutuhkan performa tinggi dan skalabilitas yang baik.

### *Hypertext Preprocessor* (PHP)

PHP atau *Hypertext Preprocessor* sebagaimana yang dijelaskan (Adrianto S., 2021) adalah sebuah bahasa pemrograman untuk membuat web yang bersifat *server-side* *scripting*, PHP memungkinkan untuk membuat halaman web bersifat dinamis. Selain itu, PHP membutuhkan *Database Management System* (DBMS) untuk dijalankan secara bersamaan. DBMS yang paling populer di kalangan pelajar pemrograman salah satunya adalah MySQL, namun, PHP juga mendukung DBMS lain seperti Oracle, Microsoft Access, Interbase, D-Base, PostgreSQL, dan DBMS yang lainnya.

Satu hal yang mungkin menjadi pertanyaan adalah apabila PHP dikenal dengan *Hypertext Preprocessor*, lalu apa kepanjangan PHP? Menurut tulisan yang diterbitkan melalui web sekawanstudio.com yang ditulis oleh (Miranda R.A., 2023) Pada tahun 1994, ketika Rasmus Lerdorf pertama kali menemukan *hypertext preprocessor*, beliau menggunakannya untuk memantau jumlah pengunjung atau yang sering kita sebut dengan *traffic website* dari halaman web pribadi nya atau dalam bahasa inggris dikenal dengan *Personal Home Page*. Alasan tersebutlah yang menyebabkan bahasa pemrograman ini dijuluki sebagai PHP.

### *Framework* PHP Laravel

Menurut penjelasan dari (Subecs, 2021) Laravel merupakan *framework* PHP yang paling sering digunakan untuk *programmer* pemula dan berpengalaman. Laravel dianggap mampu mengurangi durasi pengembangan sistem perangkat lunak serta mempersiapkan pasar dengan metode PHP berorientasi objek yang lebih modern. *Syntax-syntax* ekspresif dan *function-function* modern yang dimiliki Laravel disukai oleh para *programmer* yang ingin mengembangkan web atau aplikasi yang lebih kompleks. Dengan menggunakan *framework* ini diyakini dapat mempermudah proses pengembangan karena Laravel menggunakan sistem paket modular dimana modul-modul yang disediakan saling terkait satu sama lain, dimana kita bisa mengembangkan sistem perangkat lunak yang lebih luas lagi. *Framework* ini memberikan kita jalan pintas yang memungkinkan *programmer* berkonsentrasi terhadap masalah-masalah yang lebih penting.

## Rekayasa Perangkat Lunak

Secara bahasa, *Software Engineering* atau yang sering disebut dengan Rekayasa Perangkat Lunak dalam bahasa indonesia (RPL) tersusun dari 2 kata, *Software* dan *Engineering*. *Software* atau perangkat lunak bukan hanya sebuah program seperti yang diasumsikan banyak orang. Program adalah kode komputer yang dapat dieksekusi, yang tujuannya untuk melakukan satu atau lebih komputasi tertentu. Sedangkan *software* adalah kumpulan dari kode pemrograman komputer yang dapat dieksekusi, terorganisir dan terdokumentasi. Lalu, *Engineering* sendiri adalah semua hal yang meliputi pengembangan produk (baik fisik maupun digital) dengan menerapkan dan memanfaatkan prinsip-prinsip dan metodologi ilmiah yang tersusun dengan baik.

Sehingga, *Software Engineering* merupakan salah satu bagian dari *Engineering* yang berurusan dengan rekayasa perangkat lunak yang disusun secara baik dan menerapkan prinsip-prinsip, prosedur dan metodologi ilmiah. Hasil akhir dari RPL tentunya adalah perangkat lunak yang efisien, stabil dan bebas dari *bug*. Namun, apabila kita mengacu kepada pengertian menurut organisasi IEEE, RPL adalah (1) Penerapan dari sebuah pendekatan yang sistematik, disiplin, dan terukur untuk mengembangkan, mengoperasikan dan memelihara sebuah perangkat lunak; dalam hal ini penerapan rekayasa perangkat lunak. (2) Studi tentang pendekatan-pendekatan dari pernyataan diatas.

Selain IEEE, Fritz Bauer, ilmuwan komputer asal Jerman, mendefinisikan RPL sebagai “…pembentukan dan pemanfaatan prinsip teknik suara agar tercipta perangkat lunak yang ekonomis, yang stabil dan bekerja secara efisien di mesin sesungguhnya.”. Mengutip dari modul pembelajaran Rekayasa Perangkat Lunak – Pendekatan Terstruktur & Berorientasi Objek karya (Bahar et al., n.d.) Rekayasa perangkat lunak (*software engineering*) merupakan suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak (*software*), mulai dari tahap awal kajian spesifikasi / kebutuhan sistem sampai pemeliharaan sistem setelah digunakan (Sommerville, 2016). Pada definisi ini, ada dua istilah kunci: yang Pertama ‘Disiplin rekayasa’, yang berarti bahwa teknisi RPL membuat suatu alat bekerja. Mereka menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung.



### *Incremental Model*

Model pengembangan perangkat lunak *incremental model* dilakukan dengan membagi projek perangkat lunak menjadi bagian-bagian yang lebih kecil atau dalam istilah *incremental model* disebut dengan *iterative build.* Gambar 2.1 dibawah ini adalah siklus atau skema pengembangan perangkat lunak untuk *incremental model*.



Gambar 2. Siklus Pengembangan Perangkat Lunak *Incremental Model*

(Sumber: Sachan D., 2024)

Fitur utama dari model ini adalah proses iteratifnya, di mana setiap iterasi yang dilakukan selanjutnya dikembangkan, diuji, dan diintegrasikan satu per satu. Dengan membagi projek ke bagian-bagian kecil, tim pengembang dapat memprioritaskan fitur dan mengadaptasi perubahan-perubahan kecil dengan lebih efisien. Untuk lebih jelasnya kami akan menjelaskan setiap tahapan-tahapan di dalam *incremental model* :

1. ***Requirements*** merupakan tahap awal yang krusial dalam pengembangan perangkat lunak. Pada tahap ini, tim pengembang berfokus pada pengumpulan dan analisis kebutuhan perangkat lunak secara menyeluruh. Kebutuhan yang didapatkan berfungsi sebagai fondasi yang kokoh untuk memandu fase-fase pengembangan selanjutnya. Hasil akhir dari Requirements Gathering adalah dokumen yang berisi uraian rinci mengenai kebutuhan perangkat lunak. Dokumen ini harus ditulis dengan jelas, terstruktur, dan mudah dipahami oleh semua pihak yang terlibat dalam proyek.
2. ***Design & Development*** merupakan fase krusial dalam pengembangan perangkat lunak. Pada tahap ini, tim proyek akan melakukan pemrosesan hasil analisa kebutuhan di tahapan sebelumnya menjadi rancangan yang berfungsi sebagai acuan teknis untuk membangun perangkat lunak yang sesuai dengan kebutuhan yang telah ditetapkan sebelumnya. Desain yang dimaksud adalah desain arsitektur atau struktur dari perangkat lunak, desain interaksi perangkat lunak, dan terakhir desain tampilan perangkat lunak.Selanjutnya, tim pengembang akan berfokus pada pembangunan masing-masing fungsionalitas secara bertahap, sesuai model pengembangan yang dipilih, dalam hal ini model pengembangan perangkat lunak yang digunakan adalah *incremental model*. Sehingga proses implementasi dilakukan secara iteratif.
3. ***Testing*** merupakan salah satu fase yang wajib ada dalam pengembangan perangkat lunak untuk memastikan kualitas dan fungsionalitas yang optimal. Pengujian yang efektif dilakukan secara berkelanjutan sepanjang siklus pengembangan perangkat lunak, tidak hanya pada tahap akhir proyek. Pengujian yang dilakukan dibagi menjadi tiga, pengujian fungsional berfokus pada setiap iterasi berfungsi sesuai dengan spesifikasi. Selanjutnya, pengujian regresi yang berfokus untuk memastikan bahwa fungsionalitas yang telah dibangun sebelumnya tidak terganggu oleh penambahan fungsional baru. Dan terakhir, pengujian integrasi yang ditujukan untuk memastikan bahwa setiap fungsionalitas baru dapat bekerja sama secara harmonis dengan fungsionalitas yang dibangun pada iterasi sebelumnya.
4. ***Implementation*** proses ini dilakukan untuk memastikan bahwa semua komponen yang dibangun pada perangkat lunak sudah diimplementasikan dan dapat dioperasikan secara bersamaan tanpa gangguanatau kesalahan. Sehingga, pada fase ini pula, tim pengembang harus memastikan bahwa fitur-fitur yang dibangun di setiap iterasi dapat saling bekerja dengan baik dengan fitur-fitur dari iterasi lain.

Dengan mengimplementasikan *incremental model* tim pengembang dapat melakukan perbaikan dan penyempurnaan secara berkelanjutan. Selain itu, aplikasi yang awalnya dirasa kompleks, karena dikembangkan secara iteratif, hal ini memberikan tim pengembangan pemahaman yang lebih dalam dan memudahkan pengelolaan terkait sistem secara intrinsik.

## *Business Process Model and Notation* (BPMN)

*Business Process Model and Notation* adalah standar khusus untuk digunakan sebagai benchmark untuk pemodelan proses bisnis yang menghasilkan notasi grafis untuk memvisualisasikan proses bisnis (Firdaus et al., 2022). *Business Process Model and Notation* (BPMN) merupakan standar global yang diakui untuk pemodelan proses bisnis. Dirancang untuk memvisualisasikan dan menganalisis proses bisnis secara jelas dan mudah dipahami, BPMN telah menjadi alat penting bagi para pemangku kepentingan dalam berbagai organisasi. BPMN menyediakan notasi grafis yang terstruktur dan konsisten untuk merepresentasikan langkah-langkah, keputusan, dan aliran data dalam suatu proses bisnis. Notasi ini memungkinkan para pelaku bisnis, analis, dan pengembang perangkat lunak untuk memahami proses bisnis dengan cara yang sama, sehingga meningkatkan komunikasi dan kolaborasi di antara mereka. Notasi-notasi BPMN akan dijelaskan melalui tabel 2.1 dibawah ini.

Tabel 2. Notasi-notasi BPMN

(Sumber : OMG, 2011)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 1 |  | ***Start Event*** | Sesuai dengan nama notasinya, notasi ini mengindikasikan dimana sebuah rangkaian proses dimulai. |
| 2 |  | ***Intermediate Event*** | Notasi ini digunakan ketika terjadi sebuah proses diantara rangkaian proses. |
| 3 |  | ***End Event*** | Sebagaimana nama dari notasi disamping, notasi ini mengindikasikan akhir dari rangkaian proses. |
| 4 |  | ***Task*** | Sebuah tugas adalah aktivitas atomik yang dimasukkan di rangkaian proses bisnis. Sebuah tugas digunakan ketika sebuah pekerjaan dalam proses bisnis sudah tidak bisa lagi dipecah menjadi tugas yang lebih kecil lagi. |
| 5 |  | ***Choreograp-hy Task*** | Notasi ini merepresentasikan sebuah tugas yang melibatkan dua partisipan untuk menyelesaikannya. |
| 6 |  | ***Sub-process*** | Sebuah sub-proses adalah gabungan aktivitas yang dimasukkan pada sebuah proses. |

Tabel 2. 1 Notasi-notasi BPMN (Lanjutan-1)

(Sumber : OMG, 2011)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 7 |  | ***Gateway*** | Sebuah gateway tidak sama dengan notasi kondisional pada flowchart, notasi ini digunakan untuk mengontrol pemecahan dan pertemuan pada sebuah rangkaian proses bisnis. |
| 8 |  | ***Sequence Flow*** | Notasi ini mengindikasikan sebuah alur yang tidak dimulai dari *Intermediate Event* yang terhubung ke sebuah tugas / sub-proses. |
| 9 |  | ***Message Flow*** | Notasi ini digunakan untuk menunjukkan alur informasi antara dua partisipan yang bersiap untuk mengirim dan menerima informasi tersebut. |
| 10 |  | ***Association*** | Notasi ini digunakan untuk menghubungkan informasi dan artifak terkait teks anotasi dan artifak lain yang terasosiasikan dengan elemen grafik lainnya. |

## *Unified Modeling Language* (UML)

Menurut buku dengan judul “*The Unified Modeling Language Reference Manual”* UML adalah tujuan umum dari bahasa pemodelan visual yang digunakan untuk menspesifikasi, memvisualisasikan, menyusun, dan mendokumentasi hal-hal yang terkait sistem perangkat lunak. UML mencatat semua keputusan dan pemahaman mengenai sistem-sistem yang wajib dibangun nantinya. Selain itu, UML digunakan untuk memahami, merancang, mengeksplorasi, mengkonfigurasi, memelihara, dan mengontrol informasi terkait sistem yang dikembangkan. Namun, UML bukan merupakan bahasa pemrograman, melainkan sebuah *tools* yang dapat menginspirasi pembuatan program yang selanjutnya bisa dikembangkan menggunakan bahasa pemrograman lainnya.

Tujuan dari UML sendiri diantaranya adalah UML dibuat sebagai tujuan utama untuk bahasa pemodelan dimana semua pelaku pengembangan perangkat lunak bisa menggunakannya. Bertujuan untuk mencakup konsep-konsep metode ternama yang nantinya mereka gunakan sebagai bahasa pemodelan. UML tidak dimaksudkan untuk dijadikan metode pengembangan yang lengkap, sebab UML tidak memiliki langkah-langkah mendetail tentang proses pengembangan perangkat lunak. Sekali lagi ditegaskan oleh (Jacobson et al., 2021) UML mencakup konsep-konsep yang dianggap penting untuk mendukung sebuah proses iteratif yang modern berdasarkan penerapan dengan arsitektur yang kuat untuk menyelesaikan kebutuhan berdasarkan masalah-masalah dan kasus-kasus yang dialami pengguna.

Yang terakhir, inti dari tujuan UML adalah pemodelan yang sesederhana mungkin selama masih mampu memenuhi syarat-syarat pemodelan sistem praktis secara penuh yang nantinya akan dibuat. Teknik-teknik pemodelan dari UML yang kita kenal diantaranya adalah *Use Case Diagram*, seperti yang kami jelaskan sebelumnya, kemudian terdapat *Activity Diagram*, *Class Diagram*, *Sequence Diagram*,kemudian banyak pemodelan-pemodelan lain yang kurang populer seperti *Statechart Diagram*, *Collaboration Diagram*, *Component Diagram*, *Deployment Diagram*, *Extensibility Construct*, dan pemodelan-pemodelan yang lain.



### *Use Case Diagram*

Menurut (Simanullang et al., 2021) pada jurnalnya dengan judul Sistem Informasi Pemesanan Menu Makanan Pada Rm Sedep Roso Rantauprapat Berbasis Web, peneliti sempat sedikit menjelaskan tentang istilah ini. *Use Case Diagram* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Sedangkan (Setiawansyah et al., 2022) menjelaskan bahwa *Use Case Diagram* adalah sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem. Tabel 2.2 berikut adalah notasi-notasi yang ada pada *use case diagram*.

Tabel 2. Notasi *Use Case Diagram*

(Sumber : OMG, 2011)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 1 |  | ***Use Cases*** | Merepresentasikan fungsionalitas sebuah sistem, terkadang juga tujuan akhir dari aktor. Simbol ini selalu berada di dalam *Boundary Box.* |
| 2 |  | ***Actors*** | Aktor yang berinteraksi dengan sistem biasanya memicu *use case*. Simbol ini selalu berada di luar *Boundary Box.* |
| 3 |  | ***Association*** | Arah panah digunakan untuk menandakan sebuah hubungan antara aktor dan *use case* atau antara dua *use case*.  Panah dengan anotasi <<extend>> menandakan bahwa sebuah *use case* mungkin mengadopsi perilaku dari *use case* lain.  Panah dengan anotasi <<include>> menandakan bahwa sebuah *use case* menggunakan fungsionalitas *use case* lain. |
| 4 |  | ***Boundary Box*** | Simbol ini mengindikasikan batas lingkup sistem. |

### *Activity Diagram*

(Jacobson et al., 2021) menjelaskan *activity diagram* sebagai perwujudan khusus dari kondisi mesin yang ditujukan untuk memodel komputasi dan alur kerja sistem. Kondisi yang digambarkan pada *activity graph* mewakili kondisi ekseskusi komputasi yang dilakukan, bukan kondisi suatu objek secara spesifik. Umumnya, *activity graph* berasumsi bahwa komputasi yang terjadi tidak dipengaruhi oleh kejadian eksternal. *Activity diagram* biasanya berisi percabangan, lebih sering lagi percabangan kendali yang terbagi dua, dimana selanjutnya percabangan tersebut berjalan atau diproses bersamaan. Alur yang diproses secara bersamaan ini merepresentasikan aktivitas-aktivitas yang dapat dikerjakan oleh objek-objek atau orang-orang berbeda secara bersamaan pada sebuah organisasi. Seringkali, kejadian yang terjadi bersamaan ini muncul dari adanya agregasi, dimana objek memiliki proses yang harus dieksekusi secara bersamaan sendiri. Aktivitas yang terjadi secara bersamaan ini sebetulnya dapat dieksekusi secara bersamaan atau satu persatu. Tabel 2.3 dibawah ini menjelaskan arti dari setiap notasi untuk *activity diagram*.

Tabel 2. Notasi-notasi *Activity Diagram*

(Sumber: GeeksforGeeks, 2024)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 1 |  | ***Initial state*** | Sebuah rangkaian proses hanya membutuhkan satu kondisi awal kecuali kita menggambarkan aktivitas yang terjadi didalam aktivitas. |
| 2 |  | ***Action* atau *Activity State*** | Sebuah aktivitas merepresentasikan eksekusi dari sebuah aksi atau objek oleh objek. |
| 3 |  | ***Action Flow* atau *Control Flow*** | Alur aksi atau alur kontrol ini biasa digunakan untuk menunjukkan transisi dari satu kondisi aktivitas ke kondisi aktivitas lain |
| 4 |  | ***Decision Node* dan *Branching*** | Ketika kita akan membuat keputusan sebelum melanjutkan alur kontrol, kita bisa gunakan simpul keputusan. |

Tabel 2. 3 Notasi-notasi *Activity Diagram* (Lanjutan-1)

(Sumber: GeeksforGeeks, 2024)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 5 |  | ***Guard*** | Sebuah pelindung merujuk pada sebuah pernyataan yang ditulis disamping arah panah simpul keputusan. Pelindung membantu kita mengetahui batasan dan kondisi-kondisi yang menentukan alur sebuah proses |
| 6 |  | ***Fork*** | Simpul garpu digunakan untuk menggambarkan dua aktivitas yang terjadi secara bersamaan. Aktivitas yang berada dibawah simpul ini akan di eksekusi secara bersamaan. |
| 7 |  | ***Join*** | Simpul gabung digunakan untuk melebur dua aktivitas yang sebelumnya diproses secara bersamaan menjadi satu proses utuh kembali. |
| 8 |  | ***Merge* atau *Merge Event*** | Penggabungan aktivitas ini dapat dilakukan apabila kita ingin menggabungkan dua atau lebih aktivitas yang sebelumnya tidak diproses secara bersamaan, namun harus diselesaikan sebelum aktivitas selanjutnya dapat dieksekusi. |
| 9 |  | ***Swimlanes*** | *Swimlane* digunakan untuk mengelompokkan aktivitas-aktivitas yang dieksekusi oleh objek atau orang yang sama. |
| 10 |  | ***Time Event*** | Notasi ini merujuk pada aktivitas yang membutuhkan waktu lebih lama untuk diselesaikan. |

Tabel 2. 3 Notasi-notasi *Activity Diagram* (Lanjutan-2)

(Sumber: GeeksforGeeks, 2024)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 11 |  | ***Final State* atau *End State*** | Kondisi akhir digunakan untuk menandakan akhir dari rangkaian proses. |

### *Sequence Diagram*

Melalui buku manual UML yang diterbitkan oleh lembaga (OMG, 2011) dijelaskan bahwa *sequence diagram* digunakan untuk menampilkan sekumpulan pesan-pesan, atau dalam konteks pengembangan aplikasi pesan-pesan yang dimaksud adalah data, yang disusun dalam urutan waktu. *Sequence diagram* dapat menunjukkan sebuah skenario, atau yang lebih tepat lagi, menunjukkan jejak-jejak data pada suatu kegiatan transaksi. Dalam pengertian yang lain, sebuah *sequence diagram* dapat digunakan untuk menunjukkan urutan perilaku-perilaku pada suatu *use case*. Tabel 2.4 dibawah ini adalah notasi-notasi untuk *sequence diagram*.

Tabel 2. Notasi-notasi *Sequence Diagram*

(Sumber: Letaw L., 2024)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
| 1 |  | ***Participants* atau *Objects*** | Setiap kolom pada sebuah *sequence diagram* adalah partisipan yang terlibat. Setiap partisipan umumnya adalah objek. Nama dari partisipan yang terlibat ada di dalam kotak. |
| 2 |  | ***Lifeline*** | Garis vertikal putus-putus mewakili lama peran dari sebuah partisipan. Dimulai dari atas dimana awal peran dan dibawah untuk akhir peran. Peran partisipan berakhir apabila partisipan |

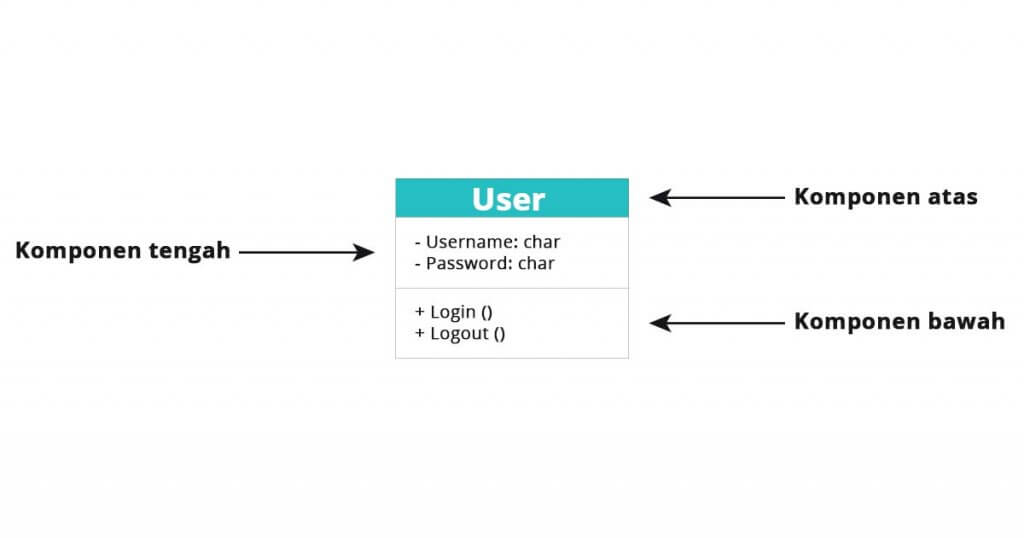
Tabel 2. 4 Notasi-notasi *Sequence Diagram*

(Sumber: Letaw L., 2024)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Notasi** | **Nama** | **Deskripsi** |
|  |  |  | dihapus. |
| 3 |  | ***Messages*** | Interaksi antar partisipan digambarkan dengan arah panah. Umumnya digunakan untuk pemanggilan *method()*. |
| 4 |  | ***Activation Bar*** | Kotak yang menumpuk *lifeline* menunjukkan bahwa partisipan sedang aktif. |
| 5 |  | ***Return*** | Arah panah dengan garis putus-putus mengindikasikan pengembalian nilai dari *method()* yang dipanggil. |
| 6 |  | ***Self-Call*** | Sebuah *method()* yang memanggil dirinya sendiri diwakilkan dengan panah yang mengarah kembali ke partisipan. |
| 7 |  | ***Deletion*** | Akhir dari peran partisipan. Digambarkan dengan (X) pada bagian bawah *lifeline* |

### *Class Diagram*

(Oktriwina, 2021) menjelaskan *class diagram* sebagai salah satu diagram struktur statis yang dimiliki UML dimana *class diagram* menggambarkan struktur sistem dengan menunjukkan sistem *class*, atribut-atribut yang dimiliki, metode-metode yang dapat dieksekusi, dan hubungan antar objek.(Jacobson et al., 2021) menjelaskan *class* itu sendiri adalah penjelasan dari konsep wewenang aplikasi atau solusi dari aplikasi. *Class* dapat dideskripsikan dari berbagai tingkatan presisi dan seberapa konkrit kita ingin mendeskripsikannya. Pada awal tahapan pengembangan perangkat lunak, *class diagram* seringkali berisi aspek-aspek logika dari masalah yang ingin diatasi. Sedangkan pada akhir tahapan pengembangan, *class diagram* dapat diisi keputusan-keputusan desain dan detail-detail implementasi di dalamnya.



Gambar 2. Notasi *Class Diagram*

(Sumber : Setiawan R., 2021)

Dari gambar 2.3 diatas, dapat kita ketahui bahwa komponen atas berisikan nama *class*. Setiap *class* pasti memiliki nama yang berbeda-beda, sebutan lain untuk nama ini adalah *simple name* (nama sederhana). Komponen tengah berisikan atribut dari *class*, komponen ini digunakan untuk menjelaskan kualitas dari suatu kelas. Atribut ini dapat menjelaskan dapat ditulis lebih detail, dengan cara memasukan tipe nilai. Sedangkan komponen bawah menyertakan operasi yang ditampilkan dalam bentuk daftar. Operasi ini dapat menggambarkan bagaimana suatu *class* dapat berinteraksi dengan data.

## *Black Box Testing*

*Black Box Testing* menguji elemen-elemen penting dalam sistem tanpa menghiraukan logika atau *source code* dari sistem yang diuji. (Adriko, S.A., 2024) *Black box testing* adalah teknik pengujian dengan merancang data pengujian yang didasarkan pada kinerja sistem yang tujuannya untuk menguji apakah sistem yang dibangun sudah beroperasi dengan benar sesuai yang diharapkan. Selanjutnya, (Setiawan, R., 2021) menjelaskan *black box testing* sebagai berikut. *Black box testing* atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, penguji tidak harus memiliki kemampuan menulis kode program. Pengujian ini dapat dilakukan oleh siapa saja. Pengujian *black box* dapat dicontohkan sebagai berikut.

Tingkat pengujian dalam *black box testing* menurut (Singh, 2023) dibagi menjadi 6 kategori, namun, demi kesederhanaan pengujian sistem, kami akan memilih 3 kategori saja dimulai dari tertinggi ke yang terendah, *Major,* fungsi dengan tingkat pengujianini sering kali merupakan masalah signifikan yang memengaruhi fungsi inti perangkat lunak, namun tidak menonaktifkannya sepenuhnya. *Minor,* biasanya dianggap gangguan yang tidak memengaruhi fungsionalitas, namun merupakan gangguan kecil yang muncul di dalam sistem. Sedangkan *Trivial*, tingkat pengujian yang lebih kecil *Minor*, biasanya hanya masalah tampilan saja dengan sedikit atau tanpa dampak pada fungsionalitas. Mereka berprioritas rendah dan umumnya hanya diperbaiki untuk meningkatkan pengalaman pengguna atau untuk mempertahankan penampilan yang baik.

Contoh 1: Masukkan *username* dan *password* dengan benar. Respon yang diharapkan adalah pengguna berhasil masuk ke aplikasi Instagram dan sistem mengarahkan pengguna ke halaman profil.

Contoh 2: Pengguna mengunggah foto berukuran besar ke aplikasi Instagram. Respon yang diharapkan adalah sistem menampilkan pesan kesalahan untuk memberi tahu pengguna bahwa ukuran foto terlalu besar.

## ISO 25010:2023

Lembaga (ISO, 2011) sendiri menjelaskan ISO 25010:2023 sebagai dokumen ini mendefinisikan model kualitas produk, yang berlaku untuk produk ICT *(Information and Communication Technology)* dan produk perangkat lunak. Model kualitas produk terdiri dari sembilan karakteristik (yang dibagi lagi menjadi sub-sub karakteristik) yang berhubungan dengan sifat kualitas produk. Karakteristik dan subkarakteristik memberikan model acuan kualitas produk yang akan ditentukan, diukur dan dievaluasi. Tabel 2.5 dibawah ini menjelaskan tentang Karakteristik dan sub-sub karakteristik dari ISO 25010:2023.

Tabel 2. Karakteristik ISO 25010:2023

(Sumber : ISO, 2023)

|  |  |  |
| --- | --- | --- |
| **No** | **Karakteristik** | **Sub Karakteristik** |
| 1 | *Functional Suitability* | *Functional Completeness* |
| *Functional Correctness* |
| *Functional Appropriateness* |
| 2 | *Performance Efficiency* | *Time Behavior* |
| *Resource Utilization* |
| *Capacity* |
| 3 | *Compatibility* | *Co-Existence* |
| *Interoperability* |
| 4 | *Interaction Capability* | *Appropriate Recognizability* |
| *Learnability* |
| *Operability* |
| *User Error Protection* |
| *User Engagement* |
| *Inclusivity* |
| *User Assistance* |
| *Self-Descriptiveness* |
| 5 | *Reliability* | *Faultlessness* |
| *Availability* |
| *Fault Tolerance* |
| *Recoverability* |
| 6 | *Security* | *Confidentiality* |
| *Integrity* |
| *Non-Repudiation* |
| *Accountability* |
| *Authenticity* |
| *Resistance* |

Tabel 2. 5 Karakteristik ISO 25010:2023 (Lanjutan-1)

(Sumber : ISO, 2023)

|  |  |  |
| --- | --- | --- |
| **No** | **Karakteristik** | **Sub Karakteristik** |
| 7 | *Maintainability* | *Modularity* |
| *Reusability* |
| *Analysability* |
| *Modifiability* |
| *Testability* |
| 8 | *Flexibility* | *Adaptability* |
| *Scalability* |
| *Installability* |
| *Replaceability* |
| 9 | *Safety* | *Operational Constraint* |
| *Risk Identification* |
| *Fail Safe* |
| *Hazard Warning* |
| *Safe Integration* |

## Skala Likert

(Adriko, S. A., 2024) menjelaskan bahwa skala Likert, sering diaplikasikan untuk mengukur penilaian, pandangan, sikap, persepsi seseorang atau kelompok tentang suatu isu atau fenomena. Dengan demikian, pengujian sistem menggunakan ISO 25010 sebelumnya, dapat diukur untuk setiap pertanyaan atau pernyataan yang diajukan kepada partisipan *testing* dengan lebih objektif. Berikut adalah bobot nilai menggunakan skala Likert yang akan dijelaskan melalui tabel 2.6.

Tabel 2. Bobot penilaian menggunakan skala Likert

|  |  |
| --- | --- |
| **Jawaban** | **Bobot** |
| Sangat Setuju | 5 |
| Setuju | 4 |
| Ragu-Ragu | 3 |
| Tidak Setuju | 2 |
| Sangat Tidak Setuju | 1 |

# BAB III

**METODE PENELITIAN**



## *Requirements*

Menganalisa kebutuhan adalah tahap yang tidak dapat terpisahkan dari proses pengembangan perangkat lunak, begitu pula penelitian kami dalam merancang aplikasi untuk penyedia jasa kursus mengemudi, hal pertama yang kita lakukan adalah melakukan wawancara dan observasi terhadap tiga penyedia jasa kursus mengemudi di Surabaya untuk mendapatkan pengetahuan tentang proses bisnis dan kendala-kendala yang selama ini dihadapi oleh mereka. Berikut adalah hasil dari proses pengumpulan dan analisa kebutuhan yang kami lakukan.



### Proses Bisnis

Seperti yang kita ketahui, berkendara dengan tertib di jalan raya merupakan tanggung jawab kita semua, namun, dalam menjaga ketertiban tersebut dibutuhkan pihak yang memiliki kewenangan untuk mengatur, mengarahkan, bahkan memberikan sanksi apabila terjadi pelanggaran di jalan raya, pihak tersebut tidak lain adalah kepolisian dan dinas perhubungan. Oleh karena itu, para penyedia jasa kursus mengemudi wajib bekerja sama dengan pihak-pihak tersebut untuk mendapatkan izin sebelum melangsungkan bisnisnya. Jika kita mengacu pada Peraturan Daerah Kota Surabaya Nomor 22 Tahun 2012 tentang Izin Penyelenggaraan Pendidikan dan Pelatihan Mengemudi Kendaraan Bermotor, para penyedia jasa kursus mengemudi, baik perorangan maupun lembaga pendidikan, yang ingin menawarkan jasanya kepada masyarakat diharapkan dapat memenuhi persyaratan yang diperlukan.

Pada saat kami melakukan penelitian ini belum ada standar atau peraturan yang mengatur atau mewajibkan para penyedia jasa kursus mengemudi untuk memiliki suatu proses tertentu. Sehingga, langkah selanjutnya adalah kami menanyakan bagaimana proses bisnis yang ada pada ketiga kursus mengemudi tersebut. Agar lebih mudah dipahami, kami bagi proses bisnis menjadi 2 alur, Pendaftaran Kursus dan Proses Kursus Mengemudi.



Gambar 3. Proses Pendaftaran pada Kursus Mengemudi ABC

Dapat dilihat pada gambar 3.1 diatas, proses pendaftaran pada Kursus ABC dimulai dengan calon siswa memilih kelas kursus kemudian menyusun jadwal, jika calon siswa merasa sudah puas proses selanjutnya adalah pembayaran, dimana nominal uang yang dibayarkan harus diperiksa terlebih dahulu oleh pihak administrasi Kursus ABC, jika sudah benar, pihak administrasi Kursus ABC memberikan tanda terima pembayaran ke calon siswa beserta kartu kehadiran kursus yang nantinya wajib dibawa setiap kursus.



Gambar 3. Proses Kursus di Kursus Mengemudi ABC

Sedangkan proses kursus di kursus mengemudi ABC sebagaimana yang dijelaskan pada gambar 3.2, dimulai dengan siswa hadir ke tempat kursus atau siswa bisa memilih untuk dijemput oleh instruktur kursus di rumahnya. Sebelum memulai praktik, siswa menunjukkan kartu kehadiran kursus ke instruktur untuk ditandatangani kehadirannya bersamaan dengan instruktur memeriksa siswa saat ini berada di pertemuan ke berapa, dimana siswa selanjutnya mempelajari materi sesuai dengan pertemuan saat ini.



Gambar 3. Proses Pendaftaran di Kursus Mengemudi Sie Bersaudara

Gambar 3.3 diatas menunjukkan alur proses pendaftaran di Kursus Sie Bersaudara, dimulai dengan siswa mengisi formulir, di dalam formulir tersebut terdapat opsi kelas dan deskripsi dari masing-masing kelas, setelah mengisi formulir, siswa melakukan pembayaran, sama dengan kursus ABC, pihak administrasi kursus Sie Bersaudara memeriksa apakah nominal yang dibayarkan sudah benar, jika sudah, selanjutnya siswa memilih jadwal kursus. Kemudian, pihak administrasi memberikan bukti pembayaran dan buku pegangan yang berisi materi yang nantinya akan diajarkan serta tertera lembar absensi di dalamnya.



Gambar 3. Proses Kursus di Kursus Mengemudi Sie Bersaudara

Gambar 3.4 diatas menunjukkan proses kursus di kursus Sie Bersaudara dimulai dengan pihak administrasi mengkonfirmasi kehadiran siswa 1-2 jam sebelum jadwal kursus, jika siswa ternyata tidak bisa hadir, pihak administrasi akan menawarkan jadwal baru ke siswa yang tidak hadir tersebut, jika siswa hadir, selanjutnya siswa akan bertemu dengan instruktur, siswa harus menunjukkan buku pegangannya untuk ditandatangani lembar absensinya. Sama dengan kursus mengemudi ABC, instruktur memeriksa di pertemuan ke berapa siswa saat ini, apabila siswa ada di pertemuan terakhir, instruktur nantinya akan mengulang materi dari awal hingga pertemuan sebelumnya, kemudian, sebelum siswa pulang, instruktur wajib memfoto siswa didepan mobil kursus untuk selanjutnya dicetak menjadi sertifikat seandainya siswa ingin mendapatkan sertifikat kursus.



Gambar 3. Proses Pendaftaran di Kursus Mengemudi "Hafiz"

Berbeda dengan dua kursus sebelumnya, gambar 3.5 menjelaskan proses pendaftaran pada kursus mengemudi “Hafiz” yang tergolong kursus mengemudi perorangan. Proses pendaftaran dimulai dengan siswa langsung memilih jadwal, kemudian siswa melakukan pembayaran, dan yang terakhir pihak kursus mencetak bukti pembayaran untuk diberikan ke siswa.



Gambar 3. Proses Kursus di Kursus Mengemudi "Hafiz"

Sedangkan gambar 3.6 diatas menjelaskan proses kursus yang saat ini dilakukan pada kursus mengemudi “Hafiz”. Hampir sama dengan proses kursus sebelumnya, dimulai dengan siswa hadir ke tempat kursus atau dijemput oleh instruktur, selanjutnya instruktur menanyakan apakah pertemuan saat ini pertemuan terakhir, jika tidak, instruktur dan siswa bisa melanjutkan materi dari pertemuan sebelumnya, jika iya, instruktur bersama siswa akan mempraktekkan materi kursus dari awal hingga pertemuan sebelumnya.

### Analisa Kebutuhan

Dalam mengembangkan perangkat lunak, analisa kebutuhan merupakan hasil riset dengan beberapa pihak dan bidang yang berbeda, seperti dari bidang bisnis diantaranya adalah situasi pasar, pengguna akhir, calon konsumen, dan peluang-peluang teknis. Tabel 3.1 dibawah ini adalah daftar kebutuhan aplikasi untuk penyedia jasa kursus mengemudi.

Tabel 3. Kebutuhan Aplikasi

|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| **Sisi Admin Kursus (Lembaga) / Pemilik (Perorangan & Lembaga)** | |
| 1 | Fungsi *Login* sebagai Admin / Pemilik |
| 2 | *Dashboard* yang berisi jadwal kursus mendatang |
| 3 | Mengubah data yang ada pada profil akun lembaga kursus |
| 4 | Menambah kelas kursus beserta deskripsi, harga, jumlah pertemuan, benefit, dan kategori kelas kursus |
| 5 | Mengubah nama, deskripsi, harga, jumlah pertemuan, benefit, dan kategori kelas kursus |
| 6 | Menonaktifkan kelas kursus |
| 7 | Menghapus kelas kursus |
| 8 | Menambah Instruktur kursus |
| 9 | Menonaktifkan data Instruktur kursus |
| 10 | Menghapus data Instruktur kursus |
| 11 | Menampilkan daftar Siswa aktif |
| 12 | Menampilkan detail *progress* kursus siswa terkait |
| 13 | Menghubungi Siswa atau Instruktur |
| 14 | Menampilkan hasil pengisian formulir pendaftaran kursus |
| 15 | Memverifikasi pembayaran kursus |

Tabel 3. 1 Kebutuhan Aplikasi (Lanjutan-1)

|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| 16 | Mengubah jadwal kursus |
| 17 | Mengunggah dokumen izin penyelenggaraan kursus mengemudi |
| 18 | Menonaktifkan Sementara Lembaga Kursus |
| 19 | Menutup / Menghapus data Lembaga Kursus |
| 20 | Fungsi *Logout* dari aplikasi |
| **Sisi Instruktur (Lembaga & Perorangan)** | |
| 1 | Fungsi *Login* sebagai Instruktur |
| 2 | *Dashboard* yang berisi jadwal kursus mendatang |
| 3 | Mengubah data yang ada pada profil akun Instruktur |
| 4 | Menampilkan daftar Siswa aktif |
| 5 | Menampilkan detail *progress* kursus siswa terkait |
| 6 | Menghubungi Siswa atau Pemilik / Admin |
| 7 | Mengunggah sertifikat instruktur kursus mengemudi |
| 8 | Fungsi *Logout* dari aplikasi |
| **Sisi *General User*** | |
| 1 | Fungsi Daftar Akun |
| 2 | Fungsi *Login* sebagai *General User* |
| 3 | *Dashboard* yang berisi rekomendasi kelas kursus, penyedia kursus dan *progress* kursus yang sedang aktif jika ada |
| 4 | Mengubah data yang ada pada profil akun *General User* |
| 5 | Mengakses detail informasi kelas kursus |
| 6 | Melakukan pengisian formulir pendaftaran kelas kursus |
| 7 | Menampilkan detail *progress* kursus siswa terkait |

Tabel 3. 1 Kebutuhan Aplikasi (Lanjutan-2)

|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| 8 | Melakukan pembayaran kelas kursus |
| 9 | Menghubungi Admin / Pemilik dan Instruktur |
| 10 | Memilih jadwal kursus |
| 11 | Mengakses panduan teknik berkendara di setiap pertemuan |
| 12 | Menyelesaikan quiz di setiap pertemuan |
| 13 | Menampilkan riwayat kursus yang pernah diikuti |
| 14 | Melakukan pencarian kelas kursus atau penyedia kursus mengemudi |
| 15 | Menjadi pemilik kursus dengan mengunggah surat izin penyelenggaraan kursus mengemudi |
| 16 | Fungsi *Logout* dari aplikasi |
| 17 | Menghapus akun |
| **Kebutuhan Teknis** | |
| 1 | Menggunakan bahasa pemrograman berbasis web seperti HTML, CSS, Javascript dan PHP lebih spesifik menggunakan *Framework* Laravel |
| 2 | Menggunakan sistem pengelolaan basis data relasional MySQL |
| 3 | Aplikasi dibangun untuk semua jenis *browser* |
| 4 | Spesifikasi perangkat keras yang disarankan adalah RAM diatas 2 GB |

## *Build* 1 : Sisi Pemilik / Admin Kursus

Proses *increment* kami mulai dengan mengembangkan aplikasi dari sisi Pemilik / Admin Kursus. Kami menganggap bahwa, keberadaan Pemilik / Admin Kursus dirasa krusial dalam proses kursus mengemudi. Proses kursus mengemudi pada aplikasi kami juga tidak dapat dimulai tanpa adanya pengguna ini. Berikut adalah rangkaian proses untuk *build* 1.

### *Design & Development*

Selama rangkaian proses perancangan atau desain dan pengembangan dengan *incremental model* yang kami lakukan, kami membagi tahap perancangan menjadi dua bagian, bagian pertama untuk perancangan struktur aplikasi, bagian ini kami gunakan untuk menjelaskan akses apa saja yang dimiliki oleh masing-masing pengguna sedangkan bagian kedua menjelaskan alur kerja sistem ketika sebuah fitur dieksekusi oleh pengguna, setelah tahap perancangan selesai, selanjutnya akan dilanjutkan dengan mengimplementasikan rancangan tersebut.

#### Use Case

Dari analisa kebutuhan diatas, kami mulai memahami kebutuhan dan kendala-kendala yang dihadapi oleh pihak penyedia jasa kursus mengemudi. Selanjutnya, untuk mempermudah tim pengembang memahami bagaimana struktur aplikasi, kami akan menggunakan bantuan *use case diagram* dan *use case scenario*. Gambar 3.7 adalah *use case diagram* untuk pengguna Pemilik / Admin Kursus*.*



Gambar 3. *Use Case Diagram* untuk Admin / Pemilik

Admin / Pemilik memiliki peranan yang cukup penting dalam mengelola operasional kursus. Dapat dilihat pada diagram diatas bahwa Pemilik Kursus kami generalisasi menjadi Admin karena akses fitur yang dimiliki pada aplikasi sama, sehingga, kedua belah pihak dapat saling bekerja sama mengelola lembaga kursus mereka.

##### Proses *Login*

Proses *login* dimulai dari pengguna berada pada halaman *login*. Proses *login* dibutuhkan untuk mencatat semua aktivitas yang dilakukan oleh pengguna. Tabel 3.2 dibawah ini menjelaskan tentang interaksi antar pengguna dengan respon sistem.

Tabel 3. *Use Case Scenario* untuk proses *Login*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_login\_pemilik\_kursus | |
| **Nama *Use Case*** | *Login* | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk memberikan autentikasi kepada pengguna sehingga, setiap operasi yang dilakukan dapat dikenali oleh sistem | |
| **Kondisi Awal** | Aktor berada di halaman *Login* | |
| **Kondisi Akhir** | Aktor berhasil memasukkan *username* atau nomor whatsapp dan *password* dengan benar | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan *form* *Login* |
| 2. Aktor mengisi kolom *username* atau nomor whatsappdan *password* |  |
|  | 3. Sistem memeriksa inputan dari Aktor dengan data yang ada pada *Database* |
|  | 4. Sistem berhasil menemukan data yang cocok dengan inputan Aktor |
|  | 5. Sistem menampilkan *modals* “Login Berhasil” dan mengarahkan Aktor ke halaman selanjutnya |

##### Proses Akses *Dashboard* Pemilik / Admin Kursus

Pemilik / Admin dapat melihat jadwal kursus mendatang. Jumlah hari dibatasi sampai 7 hari demi menghindari beban halaman.. Tabel 3.3 dibawah akan menjelaskan respon sistem akan interaksi yang dilakukan oleh Aktor.

Tabel 3. *Use Case Scenario* untuk *Dashboard* Pemilik / Admin Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ dashboard\_pemilik\_kursus | |
| **Nama *Use Case*** | *Dashboard* Pemilik / Admin Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menampilkan jadwal kursus di beranda pihak kursus | |
| **Kondisi Awal** | Aktor berada di halaman Beranda | |
| **Kondisi Akhir** | Sistem menampilkan semua jadwal kursus mendatang | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem mengumpulkan data siswa yang kursus dengan Aktor |
|  | 2. Sistem secara *default* menampilkan jadwal untuk hari ini, dan memberikan opsi menampilkan jadwal untuk 7 hari kedepan |
|  | 3. Sistem menyediakan tombol untuk mengarahkan Aktor melihat detail progress kursus siswa yang dipilih |

##### Proses Mengubah Data Profil

Pemilik / Admin bisa mengubah data yang terkait dengan profil mereka, diantaranya adalah *username, password,* foto profil, nomor whatsapp, dll. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.4 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengubah Data Profil

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_ubah\_data\_akun\_pemilik\_kursus |
| **Nama *Use Case*** | Ubah Data Profil |
| **Aktor** | Pemilik / Admin Kursus |

Tabel 3. 4 *Use Case Scenario* untuk Mengubah Data Profil

|  |  |  |
| --- | --- | --- |
| **Deskripsi** | Skenario untuk mengubah data akun Aktor | |
| **Kondisi Awal** | Aktor berada di halaman Profil | |
| **Kondisi Akhir** | Aktor berhasil mengubah data pada akun | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Ubah Profil” yang ada di halaman profil |  |
|  | 2. Sistem menampilkan *form* untuk mengganti elemen-elemen yang terkait dengan akun Aktor (*username*, *password*, Nama Aktor, Nomor Whatsapp, foto profil Aktor, dll.) |
| 3. Aktor mengubah data-data terkait dengan akun yang ingin diubah |  |
| 4. Aktor mengkonfirmasi perubahan data |  |
|  | 5. Sistem melakukan penyimpanan data baru ke *database* dan memperbarui data di halaman *profil* sesuai dengan data baru |

##### Proses Menambah Kelas Kursus

Pemilik / Admin Kursus dapat melakukan penambahan kelas kursus. Penambahan kelas dimaksudkan apabila Pemilik / Admin ingin menambah kelas kursus yang mempunyai unsur berbeda dari kelas kursus sebelumnya. Sebagai contoh, unsur yang dimaksud adalah jenis mobil, apakah kelas kursus yang baru diperuntukkan bagi calon siswa yang ingin belajar mobil *matic* / manual ? Apakah kelas kursus yang baru sifatnya kilat, yang jumlah pertemuannya kurang dari 4, dan sebagainya. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.5 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menambah Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_tambah\_kelas\_kursus | |
| **Nama *Use Case*** | Tambah Kelas Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menambah kelas kursus | |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus | |
| **Kondisi Akhir** | Sistem mengarahkan aktor ke daftar kelas kursus | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Tambah Kelas Kursus” |  |
|  | 2. Sistem menampilkan *form* penambahan kelas kursus |
| 3. Aktor mengisi semua kolom yang wajib diisi |  |
|  | 4. Aktor mengkonfirmasi penambahan kelas |
|  | 5. Sistem menambahkan kelas baru ke *database* |
|  | 6. Sistem mengarahkan Aktor ke daftar kelas kursus dan menampilkan kelas kursus baru yang ditambahkan. |

##### Proses Mengubah Detail Kelas Kursus

Pemilik / Admin Kursus dapat mengubah detail kelas kursus. Perubahan kelas yang dimaksudkan apabila Pemilik / Admin ingin mengubah detail kelas kursus seperti nama, menyesuaikan harga dll. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.6 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengubah Detail Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ubah\_kelas\_kursus | |
| **Nama *Use Case*** | *Edit* Kelas Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk mengubah detail kelas kursus | |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus | |
| **Kondisi Akhir** | Sistem mengubah detail kelas baru | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih tombol “Edit” di kelas yang dipilih |  |
|  | 2. Sistem menampilkan *form* mengubah kelas kursus dengan data saat ini sudah terisi didalam setiap kolom |
| 3. Aktor mengubah data yang diinginkan |  |
| 4. Aktor mengkonfirmasi perubahan data |  |
|  | 5. Sistem menyimpan data baru di *database* |
|  | 6. Sistem mengarahkan Aktor ke daftar kelas kursus dan menampilkan kelas kursus sesuai dengan data baru |

##### Proses Menonaktifkan Kelas Kursus

Pemilik / Admin Kursus dapat melakukan penonaktifan kelas kursus yang sudah tersimpan di *database* secara sementara. Dengan demikian, kelas yang ditawarkan oleh lembaga kursus yang bersangkutan adalah kelas-kelas terpilih dan diyakini dapat bersaing dengan penyedia kursus yang lain. Untuk detail dari alur penonaktifan kelas kursus akan dijelaskan menggunakan tabel 3.7 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menonaktifkan Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_nonaktifkan\_kelas\_kursus | |
| **Nama *Use Case*** | Nonaktifkan Kelas Kursus | |
| **Aktor** | Pemilik / Admin | |
| **Deskripsi** | Skenario untuk menonaktifkan kelas kursus | |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus | |
| **Kondisi Akhir** | Sistem menghapus atau menonaktifkan kelas kursus yang dipilih | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan data kelas yang dimiliki oleh Pemilik / Admin Kursus |
| 2. Aktor memilih kelas yang akan dinonaktifkan dari daftar kelas, dengan menekan tombol *switch* |  |
|  | 3. Sistem menonaktifkan kelas kursus dan mengubah tampilan *switch* |

##### Proses Menghapus Kelas Kursus

Pemilik / Admin Kursus dapat melakukan penghapusan kelas kursus yang sudah tersimpan di *database*. Dengan demikian, kelas yang ditawarkan oleh lembaga kursus yang bersangkutan adalah kelas-kelas terpilih dan diyakini dapat bersaing dengan penyedia kursus yang lain. Untuk detail dari alur penghapusan dan penonaktifan kelas kursus akan dijelaskan menggunakan tabel 3.8 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menghapus Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_hapus\_kelas\_kursus | |
| **Nama *Use Case*** | Hapus Kelas Kursus | |
| **Aktor** | Pemilik / Admin | |
| **Deskripsi** | Skenario untuk menghapus kelas kursus | |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus | |
| **Kondisi Akhir** | Sistem menghapus atau menonaktifkan kelas kursus yang dipilih | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan data kelas yang dimiliki oleh Pemilik / Admin Kursus |
| 2. Aktor memilih tombol “Hapus” di kelas yang dipilih |  |
|  | 3. Sistem menampilkan *pop-up* untuk mengkonfirmasi aksi yang dilakukan |
| 4. Aktor menekan tombol “Ya” |  |
|  | 5. Sistem memastikan bahwa jumlah kelas yang ditawarkan lebih dari 1 |
|  | 6. Sistem melakukan penghapusan atau penonaktifan kelas sesuai yang dipilih Aktor |

##### Proses Menambah Instruktur Kursus

Pemilik / Admin Kursus dapat melakukan penambahan personil Instruktur. Penambahan Instruktur dapat dilakukan selama instruktur terkait sudah mempunyai sertifikat Instruktur. Untuk detail alur menambah instruktur kursus akan dijelaskan melalui tabel 3.9 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menambah Instruktur Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_tambah\_instruktur\_kursus | |
| **Nama *Use Case*** | Tambah Instruktur Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menambah Instruktur Kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil Pemilik / Admin Kursus | |
| **Kondisi Akhir** | Sistem menampilkan daftar Instruktur Kursus baru | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Daftar Instruktur” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman “Daftar Instruktur” menampilkan data semua instruktur yang terkait dengan Aktor |
| 3. Aktor menekan tombol “Tambah Instruktur” |  |
|  | 4. Sistem menampilkan *form* untuk mengisi data terkait Instruktur |
| 5. Aktor mengisi semua kolom yang wajib diisi |  |
| 6. Aktor mengkonfirmasi penambahan Instruktur |  |
|  | 7. Sistem secara otomatis membuat akun Instruktur baru sesuai dengan data yang dimasukkan |
|  | 8. Sistem menyimpan hasil inputan yang diisi pada *form* sebelumnya. |

Tabel 3. 9 *Use Case Scenario* untuk Menambah Instruktur Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 9. Sistem mengarahkan Aktor kembali ke halaman “Daftar Instruktur” dan menampilkan data Instruktur Kursus baru |

##### Proses Menonaktifkan Instruktur Kursus

Pemilik / Admin Kursus dapat menonaktifkan Instruktur Kursus apabila Instruktur Kursus mengajukan cuti atau sedang berhalangan selama beberapa kurun waktu tertentu. Untuk detail dari alur penonaktifan kelas kursus akan dijelaskan menggunakan tabel 3.10 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menonaktifkan Instruktur Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_nonaktifkan\_instruktur\_kursus | |
| **Nama *Use Case*** | Nonaktifkan Instruktur | |
| **Aktor** | Pemilik / Admin | |
| **Deskripsi** | Skenario untuk menonaktifkan Instruktur Kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil Pemilik / Admin Kursus | |
| **Kondisi Akhir** | Sistem menampilkan daftar Instruktur Kursus | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Daftar Instruktur” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman “Daftar Instruktur” menampilkan data semua instruktur yang terkait dengan Aktor |

Tabel 3. 10 *Use Case Scenario* untuk Menonaktifkan Instruktur Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
| 3. Aktor memilih tombol “Hapus” di instruktur yang dipilih |  |
|  | 4. Sistem menonaktifkan instruktur kursus dan mengubah tampilan *switch* |

##### Proses Menghapus Instruktur Kursus

Pemilik / Admin menghapus data Instruktur Kursus apabila sudah tidak diperlukan lagi. Untuk detail dari alur penghapusan dan penonaktifan kelas kursus akan dijelaskan menggunakan tabel 3.11 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menghapus Instruktur Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_hapus\_instruktur\_kursus | |
| **Nama *Use Case*** | Hapus Instruktur | |
| **Aktor** | Pemilik / Admin | |
| **Deskripsi** | Skenario untuk menghapus Instruktur Kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil Pemilik / Admin Kursus | |
| **Kondisi Akhir** | Sistem menampilkan daftar Instruktur Kursus | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Daftar Instruktur” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman “Daftar Instruktur” menampilkan data semua instruktur yang terkait dengan Aktor |

Tabel 3. 11 *Use Case Scenario* untuk Menghapus Instruktur Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
| 3. Aktor memilih tombol “Hapus” di instruktur kursus yang dipilih |  |
|  | 4. Sistem menampilkan *pop-up* untuk mengkonfirmasi aksi yang dilakukan |
| 5. Aktor menekan tombol “Ya” |  |
|  | 6. Sistem memastikan bahwa jumlah instruktur yang ditawarkan lebih dari 1 |
|  | 7. Sistem melakukan penghapusan instruktur kursus sesuai yang dipilih Aktor |

##### Proses Menampilkan Daftar Siswa Aktif

Untuk menampilkan daftar siswa aktif, aktor dapat mengaksesnya melalui halaman kursus, kemudian ke tab daftar siswa. Siswa aktif yang dimaksud adalah siswa-siswa yang masih memiliki pertemuan kursus sampai hari ini. Untuk mengetahui detail alur menampilkan daftar siswa aktif akan dijelaskan menggunakan tabel 3.12 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menampilkan Daftar Siswa Aktif

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_tampilkan\_daftar\_siswa\_aktif\_pemilik\_kursus |
| **Nama *Use Case*** | Menampilkan Daftar Siswa Aktif |
| **Aktor** | Pemilik / Admin Kursus |
| **Deskripsi** | Skenario untuk menampilkan daftar siswa aktif yang dibawahi Aktor |
| **Kondisi Awal** | Aktor berada di halaman kursus |

Tabel 3. 12 *Use Case Scenario* untuk Menampilkan Daftar Siswa Aktif (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Kondisi Akhir** | Sistem menampilkan semua siswa yang terkait dengan Aktor | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tab “Daftar Siswa Aktif” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman “Daftar Siswa Aktif” |
|  | 3. Sistem mengambil semua data siswa yang masih memiliki pertemuan kursus sampai hari ini |

##### Proses Menampilkan Detail *Progress* Kursus Siswa

Pemilik / Admin dapat melihat detail kursus dari siswa yang dibawahi. Halaman detail *progress* ini menunjukkan nama instruktur dan di pertemuan ke berapakah siswa saat ini, serta apakah siswa sudah membaca teori yang diberikan, apakah *quiz* yang diberikan sudah dikerjakan, dan capaian-capaian lain. Tabel 3.13 akan menjelaskan alur dari proses pihak kursus menampilkan detail *progress* siswa.

Tabel 3. *Use Case Scenario* untuk Menampilkan Detail *Progress* Kursus Siswa

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_akses\_detail\_progress\_kursus\_pemilik\_kursus |
| **Nama *Use Case*** | Menampilkan Detail *Progress* Siswa |
| **Aktor** | Pemilik / Admin Kursus |
| **Deskripsi** | Skenario untuk menampilkan detail kursus aktif |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus |
| **Kondisi Akhir** | Aktor mengetahui informasi lebih lanjut tentang kursus siswa |

Tabel 3. 13 *Use Case Scenario* untuk Menampilkan Detail *Progress* Kursus Siswa (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih salah satu siswa dari daftar jadwal yang ditampilkan di *dashboard* atau Daftar Siswa Aktif |  |
|  | 2. Sistem menampilkan data kursus (Nama Kelas Kursus, Instruktur yang bertugas, dll.) terkait dengan siswa yang dipilih Aktor |
|  | 3. Sistem menampilkan daftar capaian yang sudah dilakukan oleh siswa |

##### Proses Menghubungi Siswa atau Instruktur Kursus

Pemilik / Admin Kursus dapat menghubungi Siswa atau Instruktur Kursus melalui aplikasi yang selanjutnya akan diteruskan melalui Whatsapp. Nomor setiap Aktor didapatkan dari data pada akun profil masing-masing Aktor. Selanjutnya akan dijelaskan lebih jauh menggunakan tabel 3.14 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menghubungi Siswa atau Instruktur Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_komunikasi\_pemilik\_kursus |
| **Nama *Use Case*** | Hubungi Siswa dan Instruktur Kursus |
| **Aktor** | Pemilik / Admin Kursus |
| **Deskripsi** | Skenario untuk melakukan komunikasi daring antar Aktor |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus Siswa |
| **Kondisi Akhir** | Aktor berhasil sampai di tampilan percakapan Whatsapp dengan Aktor yang dimaksud |

Tabel 3. 14 *Use Case Scenario* untuk Menghubungi Siswa atau Instruktur Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Hubungi Pihak Lain” |  |
|  | 2. Sistem menampilkan dua opsi, untuk menghubungi Siswa atau Instruktur Kursus |
| 3. Aktor menekan opsi Aktor tujuan |  |
|  | 4. Sistem mengambil data Nomor Whatsapp dari *database* |
|  | 5. Sistem membuat link “wa.me/Nomor\_Telepon\_Aktor\_Tujuan” |
| 6. Aktor menekan tombol “Lanjut ke Chat” |  |

##### Proses Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus

Pemilik / Admin kursus dapat mengetahui hasil pengisian Formulir Pendaftaran Kursus yang dilakukan oleh Siswa pada halaman Detail Progress Kursus. Hak akses ini diberikan sebagai salah satu cara untuk melakukan pengarsipan secara lokal apabila diperlukan. Tabel 3.15 berikut adalah penjelasan alur pengguna dan respon sistem.

Tabel 3. *Use Case Scenario* untuk Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_tampilkan \_formulir\_kursus |
| **Nama *Use Case*** | Tampilkan Formulir Kursus |
| **Aktor** | Pemilik / Admin Kursus |

Tabel 3. 15 *Use Case Scenario* untuk Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Deskripsi** | Skenario untuk menampilkan dan mengunduh hasil pengisian formulir pendaftaran kursus | |
| **Kondisi Awal** | Aktor berada di halaman detail progress kursus | |
| **Kondisi Akhir** | Aktor berhasil mendapatkan data terkait dengan siswa | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Formulir Pendaftaran” |  |
|  | 2. Sistem menampilkan *form* pendaftaran kursus yang sudah diisi oleh siswa |

##### Proses Memverifikasi Pembayaran Kursus

Selain mengakses hasil pengisian formulir pendaftaran kursus, Pemilik / Admin selanjutnya dapat memverifikasi pembayaran yang dilakukan siswa dengan menampilkan bukti pembayaran yang diunggah dan mengubah status pembayaran menjadi “Lunas”. Untuk lebih lengkapnya, akan dijelaskan melalui tabel 3.16 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Memverifikasi Pembayaran Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_verifikasi\_pembayaran\_kursus |
| **Nama *Use Case*** | Memverifikasi Pembayaran Kursus |
| **Aktor** | Pemilik / Admin Kursus |
| **Deskripsi** | Skenario untuk memverifikasi pembayaran kursus |
| **Kondisi Awal** | Aktor berada di halaman detail progress kursus |
| **Kondisi Akhir** | Status Pembayaran berubah |

Tabel 3. 16 *Use Case Scenario* untuk Memverifikasi Pembayaran Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Pembayaran” |  |
|  | 2. Sistem menampilkan foto bukti pembayaran yang diunggah Siswa, tombol “Verifikasi Pembayaran” dan “Tolak Pembayaran” |
| 3. Aktor menekan tombol “Verifikasi Pembayaran” |  |
|  | 4. Sistem mengubah status pembayaran menjadi “Lunas” |

##### Proses Perubahan Jadwal Kursus

Setiap Aktor dapat menginisiasi perubahan jadwal kursus kepada Aktor lain. Dengan catatan, pengajuan perubahan tidak dilakukan kurang dari 24 jam dari jadwal kursus pertemuan selanjutnya. Untuk lebih jelasnya, tabel 3.17 akan menjabarkan alurnya dengan detail.

Tabel 3. *Use Case Scenario* untuk Mengubah Jadwal Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_mengubah\_jadwal\_kursus\_pemilik\_kursus |
| **Nama *Use Case*** | Mengubah Jadwal Kursus |
| **Aktor** | Pemilik / Admin Kursus |
| **Deskripsi** | Skenario untuk mengubahjadwal kursus |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus Siswa |
| **Kondisi Akhir** | Aktor berada di halaman detail *progress* kursus Siswa |

Tabel 3. 17 *Use Case Scenario* untuk Mengubah Jadwal Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor pertemuan yang ingin diubah jadwalnya dan memilih tombol “Ubah Jadwal” |  |
|  | 2. Sistem memastikan bahwa pertemuan selanjutnya > 24 jam |
|  | 3. Sistem menampilkan *form* untuk mengubah jadwal dengan jadwal saat ini sudah terisi |
| 4. Aktor mengubah Tanggal dan Jam Pertemuan yang dipilih dan mengkonfirmasi perubahan |  |
|  | 5. Sistem menyimpan jadwal kursus baru ke *database* dan mengarahkan Aktor kembali ke halaman detail *progress* kursus |

##### Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

Pemilik / Admin kursus wajib memiliki dokumen izin penyelenggaraan kursus mengemudi yang sah dan berlaku selama menjadi penyedia jasa kursus di aplikasi nantinya. Seandainya masa berlaku dokumen lama sudah habis atau dokumen sudah tidak berlaku, Pemilik / Admin kursus dapat mengunggah dokumen terbaru sekali lagi. Tabel 3.18 dibawah ini akan menjelaskan interaksi antara aktor dan respon sistem selama proses pengunggahan berlangsung.

Tabel 3. *Use Case Scenario* untuk Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_unggah\_izin\_penyelenggaraan\_kursus\_pemilik\_kursus |
| **Nama *Use Case*** | Unggah Dokumen Izin Penyelenggaraan Kursus |
| **Aktor** | Pemilik / Admin Kursus |

Tabel 3. 18 *Use Case Scenario* untuk Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Deskripsi** | Skenario untuk mengunggah dokumen izin penyelenggaraan kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil | |
| **Kondisi Akhir** | Dokumen izin penyelenggaraan kursus yang baru tersimpan di *database* | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Izin Penyelenggaraan” |  |
|  | 2. Sistem menampilkan *preview* dokumen yang tersimpan saat ini tanggal mulai dan tanggal akhir masa berlaku dokumen tersebut beserta status dokumen saat ini |
| 3. Aktor menekan tombol “Unggah Izin Baru” |  |
|  | 4. Sistem menampilkan *form* untuk Aktor dapat mengunggah file baru dan satu kolom tambahan untuk tanggal awal & berakhir masa berlaku |
| 5. Aktor memilih *file* yang akan diunggah dan memilih tanggal awal & berakhir masa berlaku dokumen tersebut |  |
|  | 6. Sistem menyimpan inputan yang dimasukkan Aktor ke *database* |

##### Proses Menonaktifkan Sementara Lembaga Kursus

Jika terjadi suatu kondisi dimana Pemilik / Admin merasa harus menonaktifkan lembaga kursus secara sementara, Pemilik / Admin dapat melakukannya melalui aplikasi. Tabel 3.19 dibawah akan menjelaskan alur proses menonaktifkan lembaga kursus secara sementara.

Tabel 3. *Use Case Scenario* untuk Menonaktifkan Sementara Lembaga Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_nonaktifkan\_lembaga\_kursus | |
| **Nama *Use Case*** | Menonaktifkan Lembaga Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menonaktifkan sementara Lembaga Kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil | |
| **Kondisi Akhir** | Aktor berada di halaman profil | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Nonaktifkan / Tutup Kursus” |  |
|  | 2. Sistem menampilkan *pop-up* untuk mengkonfirmasi apakah Aktor yakin ingin menutup kursus atau hanya ingin menonaktifkan sementara |
| 3. Aktor memilih opsi nonaktifkan sementara |  |
|  | 4. Sistem kembali mengkonfirmasi pilihan Aktor dengan memunculkan *pop-up* baru dengan pilihan “Ya” atau “Batal” |
| 5. Aktor memilih “Ya” |  |
|  | 6. Sistem menonaktifkan lembaga kursus dan mengarahkan kembali ke halaman profil |

##### Proses Menutup Lembaga Kursus

Jika terjadi suatu kondisi dimana Pemilik / Admin merasa harus menutup lembaga kursus, Pemilik / Admin dapat melakukannya melalui aplikasi. Tabel 3.20 dibawah akan menjelaskan alur proses penutupan lembaga kursus.

Tabel 3. *Use Case Scenario* untuk Menutup Lembaga Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_tutup\_lembaga\_kursus | |
| **Nama *Use Case*** | Menutup Lembaga Kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menutup Lembaga Kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil | |
| **Kondisi Akhir** | Aktor berada di halaman beranda *General User* | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Nonaktifkan / Tutup Kursus” |  |
|  | 2. Sistem menampilkan *pop-up* untuk mengkonfirmasi apakah Aktor yakin ingin menutup kursus atau hanya ingin menonaktifkan sementara |
| 3. Aktor memilih opsi tutup kursus |  |
|  | 4. Sistem menghapus data lembaga kursus Aktor, mengubah peran Aktor menjadi “*user*” dan mengarahkan kembali ke halaman beranda *General User* |

##### Proses Logout

Proses *logout* dimulai dari pengguna menekan tombol “Logout” di *navbar* atau di bagian *footer* aplikasi. Tabel 3.21 dibawah akan menjelaskan alur proses *logout*.

Tabel 3. *Use Case Scenario* untuk proses *Logout*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_logout\_pemilik\_kursus | |
| **Nama *Use Case*** | *Logout* | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk melakukan proses *logout* | |
| **Kondisi Awal** | Aktor menekan tombol dengan label “Logout” | |
| **Kondisi Akhir** | Aktor berada di halaman awal aplikasi | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan pesan *pop-up* “Anda yakin ingin keluar dari aplikasi?” dan menampilkan dua tombol, “Ya” dan “Tidak” |
| 2. Aktor menekan tombol “Ya” |  |
|  | 3. Sistem menghentikan *session* dari akun Aktor |
|  | 4. Sistem mengarahkan Aktor ke halaman awal aplikasi |

#### Activity Diagram

Setelah pemaparan *use case scenario* diatas, yang menjelaskan interaksi antar Aktor yang terlibat ­­­­­dan bagaimana respon sistem terhadap interaksi tersebut. Selanjutnya, kami akan menjelaskan alur masing-masing aktifitas diatas dengan diagram aktifitas dibawah ini.

##### *Activity Diagram* untuk Proses *Login*

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.8 berikut adalah diagram aktifitas untuk proses *Login*.



Gambar 3. *Activity Diagram* untuk Proses *Login*

##### *Activity Diagram* untuk Proses Akses *Dashboard* Pemilik / Admin Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.9 dibawah ini adalah diagram aktifitas akses *dashboard* pemilik / admin kursus.



Gambar 3. *Activity Diagram* untuk Proses Akses *Dashboard* Pemilik / Admin Kursus

##### *Activity Diagram* untuk Proses Mengubah Data Profil

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.10 dibawah ini adalah diagram aktifitas mengubah data pada akun profil.



Gambar 3. *Activity Diagram* untuk Proses Mengubah Data Profil

##### *Activity Diagram* untuk Proses Menambah Kelas Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.11 dibawah ini adalah diagram aktifitas menambah kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Menambah Kelas Kursus

##### *Activity Diagram* untuk Proses Mengubah Detail Kelas Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.12 dibawah ini adalah diagram aktifitas mengubah detail kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Mengubah Detail Kelas Kursus

##### *Activity Diagram* untuk Proses Menonaktifkan Kelas Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.13 dibawah ini adalah daigram aktifitas menonaktifkan kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Menonaktifkan Kelas Kursus

##### *Activity Diagram* untuk Proses Menghapus Kelas Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.14 dibawah ini adalah daigram aktifitas menonaktifkan kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Menghapus Kelas Kursus

##### *Activity Diagram* untuk Proses Menambah Instruktur Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.15 dibawah ini adalah diagram aktifitas menambah Instruktur Kursus.



Gambar 3. *Activity Diagram* untuk Proses Menambah Instruktur Kursus

##### *Activity Diagram* untuk Proses Menonaktifkan Instruktur Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.16 dibawah ini adalah diagram aktifitas menonaktifkan Instruktur Kursus.



Gambar 3. *Activity Diagram* untuk Proses Menonaktifkan Instruktur Kursus

##### *Activity Diagram* untuk Proses Menghapus Instruktur Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.17 dibawah ini adalah diagram aktifitas menghapus Instruktur Kursus.



Gambar 3. *Activity Diagram* untuk Proses Menghapus Instruktur Kursus

##### *Activity Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.18 dibawah ini adalah diagram aktifitas menampilkan daftar siswa aktif.



Gambar 3. *Activity Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

##### *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.19 dibawah ini adalah diagram aktifitas menampilkan detail *progress* kursus Siswa.



Gambar 3. *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

##### *Activity Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.20 dibawah ini adalah diagram aktifitas menghubungi siswa atau instruktur kursus.



Gambar 3. *Activity Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

##### *Activity Diagram* untuk Proses Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.21 dibawah ini adalah diagram aktifitas menampilkan hasil pengisian formulir pendaftaran kursus.



Gambar 3. *Activity Diagram* untuk Proses Mengakses Hasil Pengisian Formulir Pendaftaran Kursus

##### *Activity Diagram* untuk Proses Memverifikasi Pembayaran Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.22 dibawah ini adalah penjelasan dari alur proses memverifikasi pembayaran kursus.



Gambar 3. *Activity Diagram* untuk Proses Mengakses Bukti Pembayaran Kursus

##### *Activity Diagram* untuk Proses Mengubah Jadwal Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.23 dibawah ini adalah diagram aktifitas mengubah jadwal kursus.



Gambar 3. *Activity Diagram* untuk Proses Pengajuan Perubahan Jadwal Kursus

##### *Activity Diagram* untuk Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.24 dibawah ini adalah diagram aktifitas mengunggah dokumen izin penyelenggaraan kursus mengemudi.



Gambar 3. *Activity Diagram* untuk Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

##### *Activity Diagram* untuk Proses Menonaktifkan Sementara Lembaga Kursus

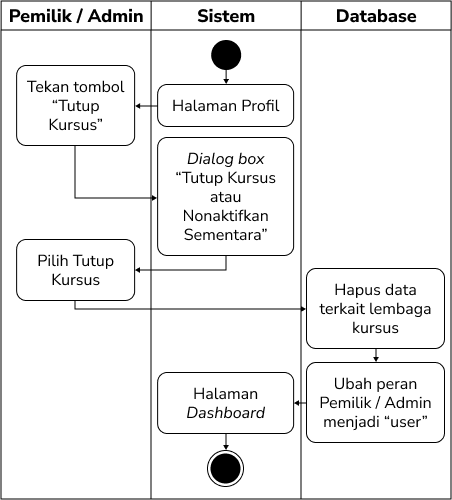
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.25 dibawah ini adalah diagram aktifitas menonaktifkan sementara lembaga kursus.



Gambar 3. *Activity Diagram* untuk Proses Menonaktifkan Sementara Lembaga Kursus

##### *Activity Diagram* untuk Proses Menutup Lembaga Kursus

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.26 dibawah ini adalah diagram aktifitas menutup lembaga kursus.



Gambar 3. *Activity Diagram* untuk Proses Menutup Lembaga Kursus

##### Activity Diagram untuk Proses Logout

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.27 dibawah ini adalah diagram aktifitas *logout*.



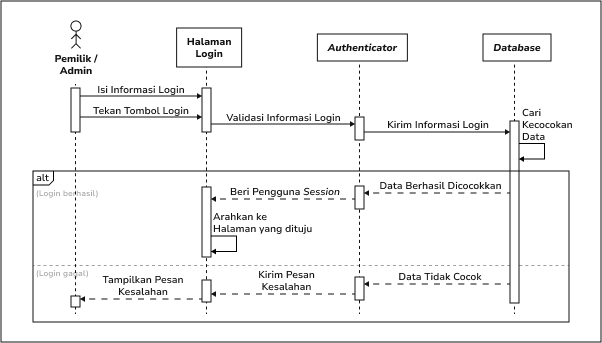
Gambar 3. *Activity Diagram* untuk Proses *Logout*

#### Sequence Diagram

Setelah pemaparan *use case scenario* dan *activity diagram* diatas, dibawah ini kami akan memaparkan jejak-jejak data ketika suatu proses berjalan menggunakan *sequence diagram*.

##### Sequence Diagram untuk Proses Login

Gambar 3.28 dibawah ini adalah alur sebuah data pada aplikasi ketika proses *login* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Login*

##### *Sequence Diagram* untuk Proses Akses *Dashboard* Pemilik / Admin Kursus

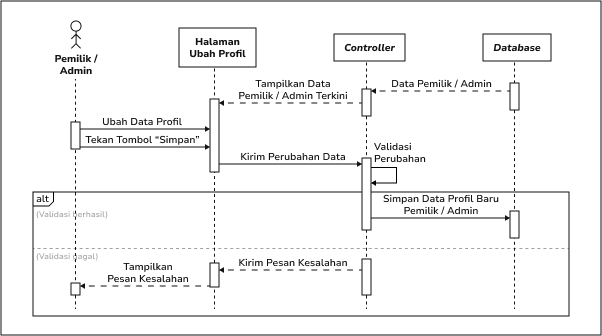
Gambar 3.29 dibawah ini adalah alur sebuah data pada aplikasi ketika proses akses *dashboard* pemilik / admindilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Akses *Dashboard* Pemilik / Admin Kursus

##### *Sequence Diagram* untuk Proses Mengubah Data Profil

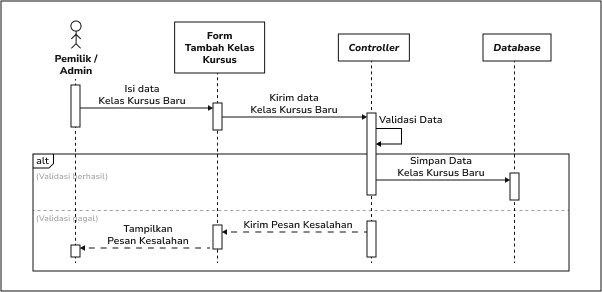
Gambar 3.30 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengubah data profildilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengubah Data Profil

##### *Sequence Diagram* untuk Proses Menambah Kelas Kursus

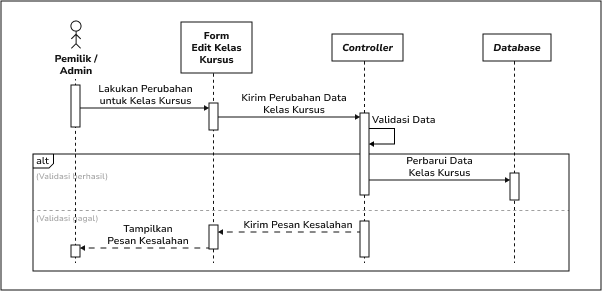
Gambar 3.31 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menambah kelas kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menambah Kelas Kursus

##### *Sequence Diagram* untuk Proses Mengubah Detail Kelas Kursus

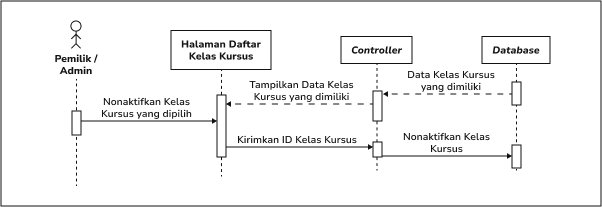
Gambar 3.32 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengubah detail kelas kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengubah Detail Kelas Kursus

##### *Sequence Diagram* untuk Proses Menonaktifkan Kelas Kursus

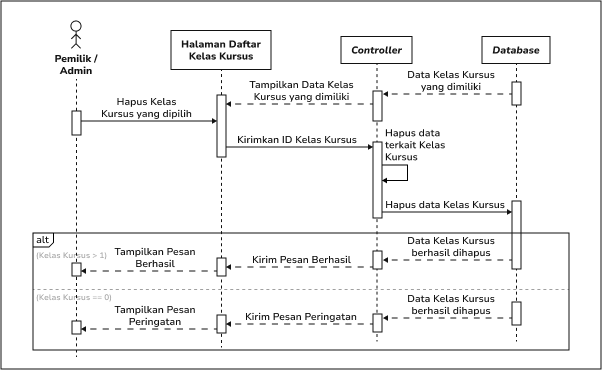
Gambar 3.33 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menonaktifkan kelas kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menonaktifkan Kelas Kursus

##### *Sequence Diagram* untuk Proses Menghapus Kelas Kursus

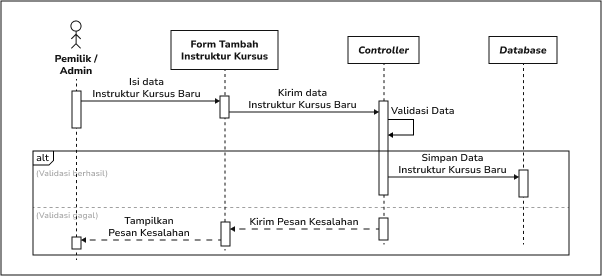
Gambar 3.34 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghapus kelas kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghapus Kelas Kursus

##### *Sequence Diagram* untuk Proses Menambah Instruktur Kursus

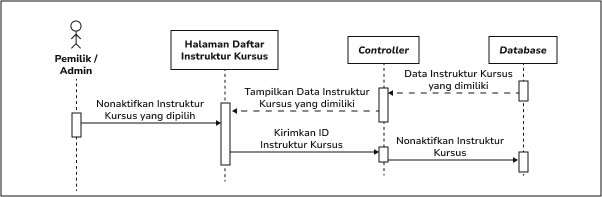
Gambar 3.35 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menambah instruktur kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menambah Instruktur Kursus

##### *Sequence Diagram* untuk Proses Menonaktifkan Instruktur Kursus

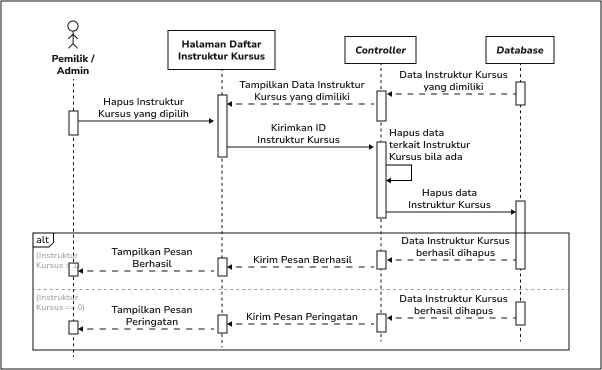
Gambar 3.36 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menonaktifkan instruktur kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menonaktifkan Instruktur Kursus

##### *Sequence Diagram* untuk Proses Menghapus Instruktur Kursus

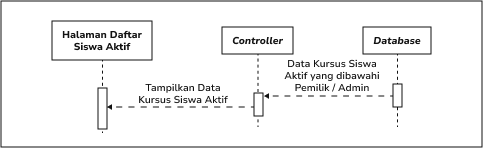
Gambar 3.37 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghapus instruktur kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghapus Instruktur Kursus

##### *Sequence Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

Gambar 3.38 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan daftar siswa aktifdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

##### *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

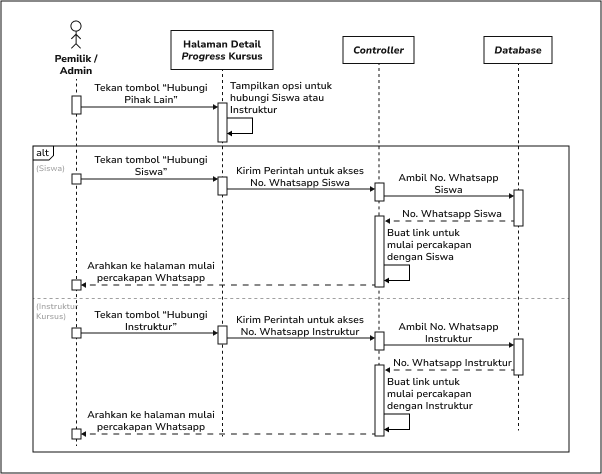
Gambar 3.39 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan detail *progress* kursus siswadilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

##### *Sequence Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

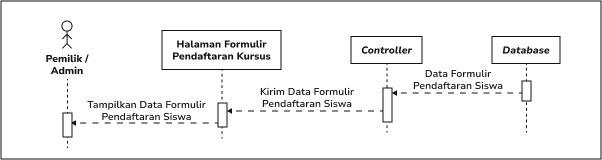
Gambar 3.40 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghubungi siswa atau instruktur kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

##### *Sequence Diagram* untuk Proses Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus

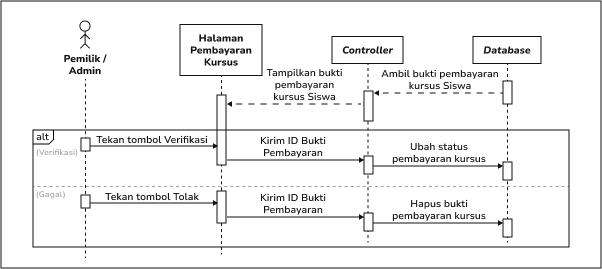
Gambar 3.41 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan hasil pengisian formulir pendaftaran kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengakses Hasil Pengisian Formulir Pendaftaran Kursus

##### *Sequence Diagram* untuk Proses Memverifikasi Pembayaran Kursus

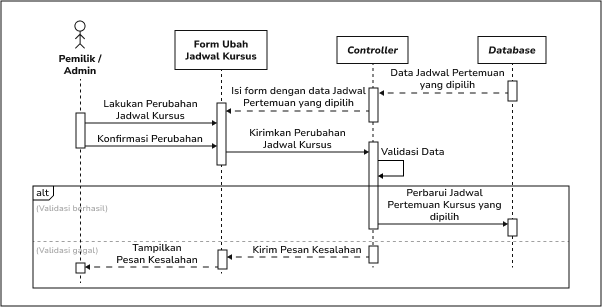
Gambar 3.42 dibawah ini adalah alur sebuah data pada aplikasi ketika proses memverifikasi pembayaran kursus dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengakses Bukti Pembayaran Kursus

##### *Sequence Diagram* untuk Proses Mengubah Jadwal Kursus

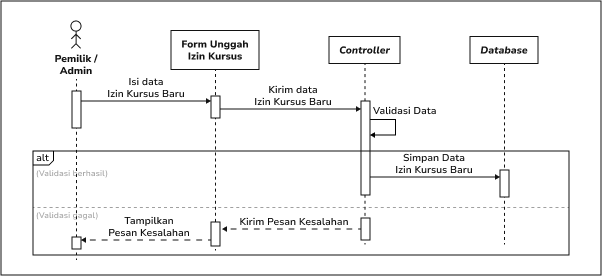
Gambar 3.43 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengubah jadwal kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengubah Jadwal Kursus

##### *Sequence Diagram* untuk Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

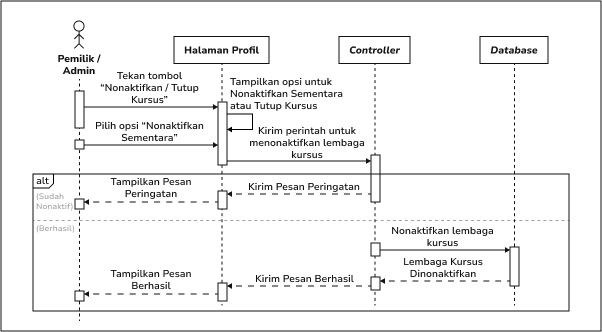
Gambar 3.44 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengunggah dokumen izin penyelenggaraan kursus mengemudidilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi

##### *Sequence Diagram* untuk Proses Menonaktifkan Sementara Lembaga Kursus

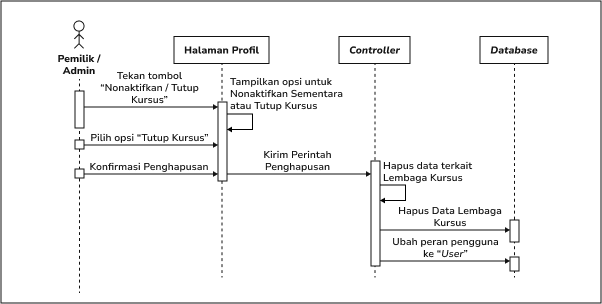
Gambar 3.45 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menonaktifkan sementara lembaga kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menonaktifkan Sementara Lembaga Kursus

##### *Sequence Diagram* untuk Proses Menutup Lembaga Kursus

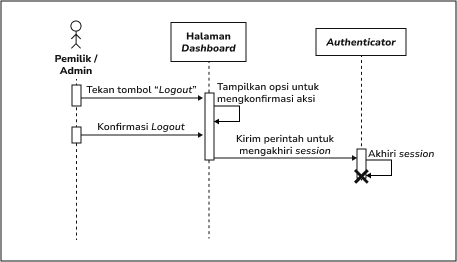
Gambar 3.46 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menutup lembaga kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menutup Lembaga Kursus

##### Sequence Diagram untuk Proses Logout

Gambar 3.47 dibawah ini adalah alur sebuah data pada aplikasi ketika proses *logout* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Logout*

Proses selanjutnya dari rangkaian proses *incremental model* adalah pengembangan aplikasi berdasarkan desain yang sudah dirancang. Pengembangan ini dilakukan untuk menerjemahkan hasil desain ke bentuk aplikasi agar dapat dilakukan interaksi, yang selanjutnya akan dilakukan proses pengujian. Sebelum proses pengembangan dilakukan, sebagai acuan dalam proses pengembangan, berikut adalah gambaran atau skema relasi antar tabel yang akan dikembangkan menggunakan bantuan *class diagram* dari gambar 3.24 dibawah ini.



Gambar 3. *Class Diagram* Aplikasi *Build* 1

### Testing

Pengujian atau *Testing* aplikasi dibagi menjadi dua fase, yaitu pengujian menggunakan *Black Box* untuk setiap *build* dan pengujian terhadap seluruh aplikasi yang nantinya menggunakan ISO 25010:2023. Tabel 3.22 dibawah akan menjelaskan alur dan skenario pengujian untuk *build* 1 aplikasi dari sisi Pemilik/Admin.

Tabel 3. *Black Box Testing* untuk *Build* 1

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses *Login* | | | |
| B1-T01 | Mengosongkan semua kolomkemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *input* | *Minor* |
| B1-T02 | Mengisi *username* / no. whatsapptetapi mengosongkan *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *password* | *Minor* |
| B1-T03 | Mengosongkan *username* / no. whatsapp tetapi mengisi *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *username* / no. whatsapp | *Minor* |
| B1-T04 | Mengisi *username* / no.whatsappyang belum terdaftar di *database* kemudian menekan tombol ‘Login’ | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B1-T05 | Pemilik/Admin menekan tombol ‘Login’ tapi *Password* salah | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B1-T06 | Mengisi *username* / no. whatsappdan *password* yang sesuai kemudian menekan tombol ‘Login’ | Sistem mengarahkan pengguna ke halaman *dashboard* | *Major* |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-1)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Mengakses *Dashboard* Pemilik / Admin | | | |
| B1-T07 | Tidak ada kursus mendatang | Sistem menampilkan pesan bahwa tidak ada kursus mendatang | *Minor* |
| B1-T08 | Pada bagian jadwal per hari, tidak ada kursus pada hari yang dipilih | Sistem menampilkan pesan bahwa tidak ada kursus pada tanggal yang dipilih | *Minor* |
| B1-T09 | Pada bagian jadwal per hari, lihat daftar kursus di salah satu hari | Sistem memberikan tampilan yang berbeda untuk kursus yang sudah dilaksanakan, sedang berlangsung, dan akan berlangsung | *Trivial* |
| Proses Mengubah Data Profil | | | |
| B1-T10 | Mengosongkan salah satu kolom yang wajib diisi pada halaman ubah profil dan menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* |
| B1-T11 | Mengunggah foto profil dengan format lain selain .jpg, .webp atau .png dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B1-T12 | Mengunggah foto profil dengan ukuran *file* terlalu besar (>2 MB) dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| B1-T13 | Menuliskan nama di kolom “Nama Lembaga Kursus” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B1-T14 | Menuliskan nama di kolom “Username” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* | |
| B1-T15 | Menuliskan *username* yang sudah ada di *database* dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Major* | |
| B1-T16 | Menuliskan deskripsi akun di kolom “Deskripsi” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* | |
| B1-T17 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif dan menekan tombol “Simpan” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Major* | |
| B1-T18 | Mengetikkan No. Whatsapp melebihi 20 digit karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* |
| B1-T19 | Mengetikkan No. Whatsapp yang sudah terdaftar dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* |
| B1-T20 | Menuliskan pertanyaan di kolom “Pertanyaan Tantangan” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa pertanyaan yang dimasukkan terlalu panjang | *Minor* |
| B1-T21 | Menuliskan pertanyaan di kolom “Jawaban” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jawaban yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-3)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T22 | Menuliskan nama di kolom “Nama Pemilik Akun Pembayaran” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |
| B1-T23 | Menuliskan nomor rekening atau alamat pembayaran di kolom “Nomor Rekening Pembayaran” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nomor rekening atau alamat pembayaran yang dimasukkan terlalu panjang | *Minor* |
| B1-T24 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |
| B1-T25 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B1-T26 | Menuliskan *password* yang berbeda di kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” dan menekan tombol “Simpan” | Sistem menampilkan kesalahan kepada pengguna untuk memastikan bahwa pengguna mengetikkan *password* yang sama | *Trivial* |
| B1-T27 | Tanpa mengubah apapun, tekan tombol “Simpan” | Sistem tetap menyimpan data walaupun tidak ada perubahan dan mengarahkan ke halaman profil | *Major* |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-4)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T28 | Melakukan perubahan dengan benar dan sesuai kemudian menekan tombol “Simpan” | Sistem melakukan perubahan di *database* dan mengarahkan pengguna ke halaman profil | *Minor* |
| B1-T29 | Melakukan perubahan kemudian menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman profil | *Minor* |
| Proses Menambah Kelas Kursus | | | |
| B1-T30 | Mengosongkan salah satu atau lebih kolom data atau tidak memilih salah satu opsi kemudian menekan tombol “Tambah” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi atau memilih salah satu opsi | *Minor* | |
| B1-T31 | Mengunggah *thumbnail* kelas kursus dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |
| B1-T32 | Mengunggah *thumbnail* kelas kursus dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B1-T33 | Menuliskan nama di kolom “Nama Kelas Kursus” melebihi 255 karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* | |
| B1-T34 | Menuliskan nama di kolom “Deskripsi Kelas” melebihi 255 karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-5)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T35 | Mengetikkan angka “0” di kolom “Kuota Kelas” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa kuota minimal adalah 1 | *Minor* | |
| B1-T36 | Mengetikkan angka diatas “999” di kolom “Kuota Kelas” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa kuota maksimal adalah 999 | *Minor* | |
| B1-T37 | Mengetikkan angka “0” di kolom “Jumlah Pertemuan” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jumlah minimal adalah 1 | *Minor* | |
| B1-T38 | Mengetikkan angka diatas “20” di kolom “Jumlah Pertemuan” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jumlah maksimal adalah 20 | *Minor* | |
| B1-T39 | Memasukkan harga di bawah Rp. 1.000,- di kolom “Harga Kelas” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa harga kursus terlalu kecil | *Minor* | |
| B1-T40 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Tambah” | Sistem menyimpan data kelas baru ke *database* dan mengarahkan pengguna ke daftar kelas kursus untuk menampilkan kelas kursus yang baru ditambahkan ada di urutan pertama | *Major* | |
| B1-T41 | Menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman daftar kelas kursus | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-6)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Mengubah Detail Kelas Kursus | | | | |
| B1-T42 | Mengosongkan salah satu atau lebih kolom data atau tidak memilih salah satu opsi kemudian menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi atau memilih salah satu opsi | *Minor* | |
| B1-T43 | Mengunggah *thumbnail* kelas kursus dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |
| B1-T44 | Mengunggah *thumbnail* kelas kursus dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B1-T45 | Menuliskan nama di kolom “Nama Kelas Kursus” melebihi 255 karakter kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* | |
| B1-T46 | Menuliskan nama di kolom “Deskripsi Kelas” melebihi 255 karakter kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* | |
| B1-T47 | Mengetikkan angka “0” di kolom “Kuota Kelas” kemudian menekan tombol“Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa kuota minimal adalah 1 | *Minor* | |
| B1-T48 | Mengetikkan angka diatas “999” di kolom “Kuota Kelas” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa kuota maksimal adalah 999 | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-7)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T49 | Mengetikkan angka “0” di kolom “Jumlah Pertemuan” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jumlah minimal adalah 1 | *Minor* | |
| B1-T50 | Mengetikkan angka diatas “20” di kolom “Jumlah Pertemuan” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jumlah maksimal adalah 20 | *Minor* | |
| B1-T51 | Memasukkan harga di bawah Rp. 1.000,- di kolom “Harga Kelas” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa harga kursus terlalu kecil | *Minor* | |
| B1-T52 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Simpan” | Sistem menyimpan data kelas baru ke *database* dan mengarahkan pengguna ke daftar kelas kursus untuk menampilkan kelas kursus yang baru ditambahkan ada di urutan pertama | *Major* | |
| B1-T53 | Menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman daftar kelas kursus | *Minor* | |
| Proses Menonaktifkan Kelas Kursus | | | | |
| B1-T54 | Menekan *switch* Kelas Kursus yang Aktif | Sistem mengubah kondisi *switch* menjadi Nonaktif | *Minor* | |
| B1-T55 | Menekan *switch* Kelas Kursus yang Nonaktif | Sistem mengubah kondisi *switch* menjadi Aktif | *Minor* | |
| Proses Menghapus Kelas Kursus | | | | |
| B1-T56 | Menekan tombol “Hapus” pada salah satu kelas tapi tidak ada siswa aktif | Sistem memunculkan dialog *pop-up* untuk mengkonfirmasi penghapusan | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-8)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T57 | Menekan tombol “Hapus” pada salah satu kelas tapi sedang ada siswa aktif | Sistem menampilkan pesan peringatan bahwa penghapusan gagal | *Major* | |
| B1-T58 | Menekan tombol “Ya, Hapus Kelas” pada dialog *pop-up* | Sistem menghapus data kelas kursus terkait dari *database* dan memperbarui daftar kelas tanpa menampilkan kelas yang dihapus | *Minor* | |
| B1-T59 | Menekan tombol “Batal” atau “X” pada dialog *pop-up* | Sistem menutup dialog *pop-up* | *Minor* | |
| Proses Menambah Instruktur Kursus | | | | |
| B1-T60 | Mengosongkan salah satu atau lebih kolom data atau tidak memilih salah satu opsi kemudian menekan tombol “Tambah” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi atau memilih salah satu opsi | *Minor* | |
| B1-T61 | Mengunggah sertifikat instruktur dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |
| B1-T62 | Mengunggah sertifikat instruktur dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B1-T63 | Mengunggah foto profil instruktur dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-9)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T64 | Mengunggah foto profil instruktur dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B1-T65 | Menuliskan nama di kolom “Nama Instruktur Kursus” melebihi 255 karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* | |
| B1-T66 | Menuliskan nama di kolom “Username” melebihi 255 karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* | |
| B1-T67 | Menuliskan *username* yang sudah ada di *database* kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Major* | |
| B1-T68 | Menuliskan angka dibawah 18 pada kolom “Usia Instruktur” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia minimal adalah 18 tahun | *Minor* | |
| B1-T69 | Menuliskan angka diatas 70 pada kolom “Usia Instruktur” kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia maksimal adalah 70 tahun | *Minor* | |
| B1-T70 | Menuliskan deskripsi akun di kolom “Deskripsi” melebihi 255 karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-10)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T71 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif kemudian menekan tombol “Tambah” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Major* | |
| B1-T72 | Mengetikkan No. Whatsapp melebihi 20 digit karakter kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* |
| B1-T73 | Mengetikkan No. Whatsapp yang sudah terdaftar dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* |
| B1-T74 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |
| B1-T75 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit kemudian menekan tombol “Tambah” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B1-T76 | Menuliskan *password* yang berbeda di kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kemudian menekan tombol “Tambah” | Sistem menampilkan kesalahan kepada pengguna untuk memastikan bahwa pengguna mengetikkan *password* yang sama | *Minor* |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-11)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T77 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Tambah” | Sistem menyimpan data instruktur kursus baru ke *database* dan mengarahkan pengguna ke daftar instruktur kursus untuk menampilkan instruktur kursus yang baru ditambahkan ada di urutan pertama | *Major* | |
| B1-T78 | Menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman daftar instruktur kursus | *Minor* | |
| Proses Menonaktifkan Instruktur Kursus | | | | |
| B1-T79 | Menekan *switch* Instruktur Kursus yang Aktif | Sistem mengubah kondisi *switch* menjadi Nonaktif | *Minor* | |
| B1-T80 | Menekan *switch* Instruktur Kursus yang Nonaktif dan Instruktur masih memiliki sertifikat aktif | Sistem mengubah kondisi *switch* menjadi Aktif | *Minor* | |
| B1-T81 | Menekan *switch* Instruktur Kursus yang Nonaktif tetapi Instruktur tidak memiliki sertifikat aktif | Sistem menampilkan pesan peringatan bahwa Instruktur tidak memiliki sertifikat aktif | *Minor* | |
| Proses Menghapus Instruktur Kursus | | | | |
| B1-T82 | Menekan tombol “Hapus” pada salah satu Instruktur dan Instruktur tidak memiliki jadwal kursus mendatang | Sistem memunculkan dialog *pop-up* untuk mengkonfirmasi penghapusan | *Minor* | |
| B1-T83 | Menekan tombol “Hapus” pada salah satu Instrukturtapi Instruktur masih memiliki jadwal kursus mendatang | Sistem menampilkan pesan peringatan bahwa proses tidak dapat dilanjutkan karena Instruktur masih mengajar | *Major* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-12)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T84 | Menekan tombol “Ya, Hapus Instruktur” pada dialog *pop-up* | Sistem menghapus data instruktur kursus terkait dari *database* dan memperbarui daftar instruktur tanpa menampilkan instruktur yang dihapus | *Major* | |
| B1-T85 | Menekan tombol “Batal” atau “X” pada dialog *pop-up* | Sistem menutup dialog *pop-up* | *Minor* | |
| Proses Menampilkan Daftar Siswa Aktif | | | | |
| B1-T86 | Tidak ada Siswa yang punya jadwal kursus mendatang | Sistem menampilkan pesan bahwa pengguna belum mempunyai siswa aktif | *Trivial* | |
| Proses Menampilkan Detail *Progress* Kursus Siswa | | | | |
| B1-T87 | Pada bagian informasi pertemuan | Sistem memberikan tampilan yang berbeda untuk pertemuan sudah dilaksanakan, pertemuan selanjutnya, dan sisa pertemuan mendatang | *Trivial* | |
| B1-T88 | Untuk siswa yang belum mengunggah bukti pembayaran | Sistem menampilkan peringatan bahwa siswa belum mengunggah bukti pembayaran | *Minor* | |
| B1-T89 | Untuk siswa yang sudah mengunggah bukti pembayaran tapi belum diverifikasi | Sistem menampilkan peringatan bahwa bukti pembayaran belum diverifikasi | *Minor* | |
| B1-T90 | Pada masing-masing informasi pertemuan, jadwal kursus selanjutnya kurang dari 24 jam | Sistem tidak menampilkan tombol “Ubah Jadwal” | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-13)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Menghubungi Siswa atau Instruktur Kursus | | | | |
| B1-T91 | Memilih opsi “Hubungi Siswa” atau “Hubungi Instruktur” | Sistem mengarahkan pengguna ke halaman mulai percakapan Whatsapp | *Major* | |
| Proses Menampilkan Hasil Pengisian Formulir Pendaftaran Kursus | | | | |
| B1-T92 | Membuka halaman “Formulir Pendaftaran” | Sistem menampilkan Informasi Pribadi Siswa, Instruktur Pengajar, dan Kelas Kursus yang dipilih | *Major* | |
| Proses Memverifikasi Pembayaran Kursus | | | | |
| B1-T93 | Membuka halaman Pembayaran, tetapi siswa belum mengunggah bukti pembayaran | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus siswa dan menampilkan pesan untuk coba lagi nanti | *Minor* | |
| B1-T94 | Menekan tombol “Tolak Pembayaran” di halaman Pembayaran | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus siswa dan menampilkan pesan bahwa penolakan bukti pembayaran berhasil | *Minor* | |
| B1-T95 | Menekan tombol “Verifikasi Pembayaran” di halaman Pembayaran | Sistem memunculkan dialog *pop-up* untuk mengkonfirmasi aksi | *Minor* | |
| B1-T96 | Menekan tombol “Batal” atau “X” pada dialog *pop-up* | Sistem menutup dialog *pop-up* | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-14)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T97 | Menekan tombol “Ya, Verifikasi” pada dialog *pop-up* | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus siswa dan menampilkan pesan bahwa verifikasi berhasil dan muncul icon “*checkmark*” disamping Lunasi Pembayaran pada Informasi Pertemuan untuk Pertemuan 1 | *Major* | |
| Proses Mengubah Jadwal Kursus | | | | |
| B1-T98 | Mengosongkan salah satu atau lebih kolom data atau tidak memilih salah satu opsi kemudian menekan tombol “Ajukan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi atau memilih salah satu opsi | *Minor* | |
| B1-T99 | Memilih Instruktur yang nonaktif | Sistem menampilkan peringatan kepada pengguna bahwa instruktur sudah aktif | *Trivial* | |
| B1-T100 | Mengubah jadwal kursus dimana jadwal baru akan berlangsung kurang dari 24 jam kemudian menekan tombol “Ajukan” | Sistem menampilkan peringatan kepada pengguna untuk memastikan jadwal baru tidak berlangsung kurang dari 24 jam | *Major* | |
| B1-T101 | Mengubah jadwal kursus yang bertabrakan dengan jadwal kursus lain kemudian menekan tombol “Ajukan” | Sistem menampilkan peringatan kepada pengguna bahwa sudah ada jadwal yang berlangsung pada jam tersebut | *Major* | |
| B1-T102 | Mengubah jadwal kursus dengan benar dan sesuai kemudian menekan tombol “Ajukan” | Sistem mengarahkan pengguna ke halaman detail *progress* kursus siswa dan menampilkan pesan jadwal kursus berhasil diubah | *Major* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-15)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T103 | Menekan tombol “Batal” | Sistem mengembalikan pengguna ke halaman detail *progress* kursus siswa | *Minor* | |
| Proses Mengunggah Dokumen Izin Penyelenggaraan Kursus Mengemudi | | | | |
| B1-T104 | Mengosongkan salah satu atau lebih kolom data kemudian menekan tombol “Ajukan Izin Baru” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib | *Minor* | |
| B1-T105 | Mengunggah izin kursus dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Ajukan Izin Baru” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |
| B1-T106 | Mengunggah izin kursus dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Ajukan Izin Baru” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B1-T107 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Ajukan Izin Baru” | Sistem menyimpan data izin kursus baru ke *database* dan mengarahkan pengguna ke halaman izin penyelenggaraan kursus untuk menampilkan izin kursus yang baru ditambahkan ada di urutan pertama | *Major* | |
| B1-T108 | Menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman izin penyelenggaraan kursus | *Minor* | |
| Proses Menonaktifkan Sementara Lembaga Kursus | | | | |
| B1-T109 | Menekan Menu “Nonaktifkan / Tutup Kursus” | Sistem memunculkan dialog *pop-up* untuk memilih opsi Nonaktifkan atau Tutup Kursus | *Minor* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-16)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T110 | Memilih Opsi “Tidak, Nonaktifkan Sementara” pada dialog “Tutup Kursus?” | Sistem memunculkan dialog *pop-up* untuk mengkonfirmasi aksi | *Minor* | |
| B1-T111 | Memilih Opsi “Batal” atau “X” pada dialog “Nonaktifkan Sementara?” | Sistem menutup dialog *pop-up* | *Minor* | |
| B1-T112 | Memilih Opsi “Ya, Nonaktifkan” pada dialog “Nonaktifkan Sementara?” sedangkan status saat ini masih Aktif | Sistem menutup dialog *pop-up* kemudian menampilkan pesan bahwa penonaktifan lembaga berhasil | *Major* | |
| B1-T113 | Memilih Opsi “Ya, Nonaktifkan” pada dialog “Nonaktifkan Sementara?” sedangkan status saat ini sudah Nonaktif | Sistem menutup dialog *pop-up* kemudian menampilkan pesan peringatan bahwa lembaga sudah nonaktif | *Major* | |
| Proses Menutup Lembaga Kursus | | | | |
| B1-T114 | Menekan Menu “Nonaktifkan / Tutup Kursus” | Sistem memunculkan dialog *pop-up* untuk memilih opsi Nonaktifkan atau Tutup Kursus | *Minor* | |
| B1-T115 | Memilih Opsi “Batal” atau “X” pada dialog “Tutup Kursus?” | Sistem membuka dialog “Nonaktifkan Sementara” | *Minor* | |
| B1-T116 | Memilih Opsi “Ya, Tutup Kursus” pada dialog “Tutup Kursus?” tetapi pengguna masih memiliki siswa aktif | Sistem menutup dialog *pop-up* dan menampilkan pesan gagal bahwa pengguna masih memiliki siswa aktif | *Major* | |

Tabel 3. 22 *Black Box Testing* untuk *Build* 1 (Lanjutan-17)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B1-T117 | Memilih Opsi “Ya, Tutup Kursus” pada dialog “Tutup Kursus?” dan pengguna sudah tidak memiliki siswa aktif | Sistem menutup dialog *pop-up* kemudian mengarahkan pengguna ke *dashboard* dan menampilkan pesan lembaga kursus berhasil di hapus dan *Password* akun pengguna diatur ulang | *Major* | |
| Proses *Log Out* | | | | |
| B1-T118 | Pemilik/Admin menekan tombol “Log Out” | Sistem menampilkan dialog *pop-up* untuk mengkonfirmasi aksi pengguna | *Major* | |
| B1-T119 | Pemilik/Admin menekan tombol “X” atau ‘Batal’ | Sistem menutup dialog *pop-up* | *Minor* | |
| B1-T120 | Pemilik/Admin menekan tombol “Ya” | Sistem mengakhiri sesi pengguna dan mengarahkan pengguna ke halaman awal aplikasi | *Major* | |

### Implementation

Fase terakhir dari rangkaian proses pengembangan perangkat lunak dengan *incremental model* adalah mengintegrasikan *build* saat ini dengan *build* lainnya. Karena saat ini kami sedang membangun *build* pertama. Maka, fase ini dapat dilewati dan dilanjutkan dengan *build* 2.

## Build 2 : Sisi Instruktur Kursus

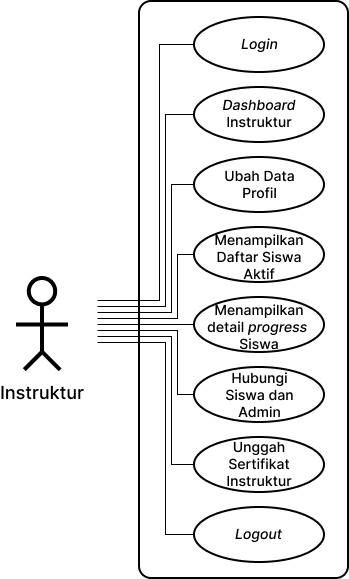
*Increment* selanjutnya adalah pengembangan aplikasi dari sisi Instruktur Kursus. Pengembangan aplikasi dari sisi Instruktur adalah lanjutan dari *increment* selanjutnya, dimana salah satu fungsi yang dimiliki Pemilik / Admin adalah menambah dan menghapus atau menonaktifkan Instruktur. Sehingga, *increment* untuk mengembangkan aplikasi dari sisi Instruktur adalah pendekatan yang paling tepat. Berikut adalah rangkaian proses untuk *build* 2.

### Design & Development

Sama dengan rangkaian proses pada pengembangan aplikasi dari sisi Pemilik / Admin yang sudah dilakukan sebelumnya. Kami menjabarkan analisa kebutuhan pada sub bab sebelumnya menjadi tiga diagram UML yang akan kami jelaskan dibawah ini. Sama halnya seperti pada *build* sebelumnya, setelah tahap perancangan selesai, selanjutnya akan dilanjutkan dengan mengimplementasikan rancangan tersebut, dimulai dengan *Use Case Diagram* dan *Use Case Scenario*.

#### Use Case

Selanjutnya, untuk mempermudah tim pengembang memahami bagaimana struktur aplikasi, kami menggunakan *use case diagram* dan *use case scenario*. Hal ini membantu pengembang untuk memahami interaksi antar pengguna akhir / *end user* dan respon sistem terhadap aksi yang dilakukan. Gambar 3.49 adalah *use case diagram* untuk pengguna Instruktur Kursus*.*



Gambar 3. *Use Case Diagram* untuk Instruktur

Instruktur Kursus berperan sebagai pihak penengah antara Siswa dan Pemilik / Admin Kursus, untuk beberapa kasus, Instruktur Kursus bisa dianggap sebagai sub-set dari Pemilik / Admin Kursus, mayoritas fungsi yang dimiliki oleh Instruktur juga dapat diakses oleh Pemilik / Admin, yang berbeda, Pemilik / Admin memiliki akses yang lebih banyak dan memiliki kendali penuh atas Instruktur yang dibawahinya.

##### Proses *Login*

Proses *login* dimulai dari pengguna berada pada halaman *login*. Proses *login* dibutuhkan untuk mencatat semua aktivitas yang dilakukan oleh pengguna. Tabel 3.23 dibawah ini menjelaskan tentang interaksi antar pengguna dengan respon sistem.

Tabel 3. *Use Case Scenario* untuk proses *Login*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_login\_instruktur\_kursus | |
| **Nama *Use Case*** | *Login* | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk memberikan autentikasi kepada pengguna sehingga, setiap operasi yang dilakukan dapat dikenali oleh sistem | |
| **Kondisi Awal** | Aktor berada di halaman *Login* | |
| **Kondisi Akhir** | Aktor berhasil memasukkan *username* atau nomor whatsapp dan *password* dengan benar | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan *form* *Login* |
| 2. Aktor mengisi kolom *username* atau nomor whatsappdan *password* |  |
|  | 3. Sistem memeriksa inputan dari Aktor dengan data yang ada pada *Database* |
|  | 4. Sistem berhasil menemukan data yang cocok dengan inputan Aktor |
|  | 5. Sistem menampilkan *modals* “Login Berhasil” dan mengarahkan Aktor ke halaman selanjutnya |

##### Proses Akses *Dashboard* Instruktur Kursus

Instruktur Kursus dapat melihat seluruh jadwal yang dimiliki. Jumlah hari yang ditampilkan dapat diatur untuk menghindari pengguna kewalahan memproses informasi. Tabel 3.24 dibawah akan menjelaskan respon sistem akan interaksi yang dilakukan oleh Aktor.

Tabel 3. *Use Case Scenario* untuk *Dashboard* Instruktur Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ dashboard\_instruktur\_kursus | |
| **Nama *Use Case*** | *Dashboard* Instruktur Kursus | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk menampilkan jadwal kursus di beranda pihak kursus | |
| **Kondisi Awal** | Aktor berada di halaman Beranda | |
| **Kondisi Akhir** | Sistem menampilkan semua jadwal | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem mengumpulkan data siswa yang kursus dengan Aktor |
|  | 2. Sistem secara *default* menampilkan jadwal untuk hari ini, dan memberikan opsi menampilkan jadwal untuk 7 hari kedepan |
|  | 3. Sistem menyediakan tombol untuk mengarahkan Aktor melihat detail progress kursus siswa yang dipilih |

##### Proses Mengubah Data Profil

Instruktur Kursus dapat mengubah data yang terkait dengan profil mereka, diantaranya adalah *username, password,* foto profil, nomor whatsapp, dll. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.25 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengubah Data Profil

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_ubah\_data\_akun\_instruktur\_kursus |
| **Nama *Use Case*** | Ubah Data Profil |
| **Aktor** | Instruktur Kursus |

Tabel 3. 25 *Use Case Scenario* untuk Mengubah Data Profil (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Deskripsi** | Skenario untuk mengubah data akun Aktor | |
| **Kondisi Awal** | Aktor berada di halaman Profil | |
| **Kondisi Akhir** | Aktor berhasil mengubah data pada akun | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Ubah Profil” yang ada di halaman profil |  |
|  | 2. Sistem menampilkan *form* untuk mengganti elemen-elemen yang terkait dengan akun Aktor (*username*, *password*, Nomor Whatsapp, foto profil Aktor, dll.) |
| 3. Aktor mengubah data-data terkait dengan akun yang ingin diubah |  |
| 4. Aktor mengkonfirmasi perubahan data |  |
|  | 5. Sistem melakukan penyimpanan data baru ke *database* dan memperbarui data di halaman *profil* sesuai dengan data baru |

##### Proses Menampilkan Daftar Siswa Aktif

Untuk menampilkan daftar siswa aktif, aktor dapat mengaksesnya melalui halaman kursus, kemudian ke tab daftar siswa. Siswa aktif yang dimaksud adalah siswa-siswa yang masih memiliki pertemuan kursus sampai hari ini. Untuk mengetahui detail alur menampilkan daftar siswa aktif akan dijelaskan menggunakan tabel 3.26 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menampilkan Daftar Siswa Aktif

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_tampilkan\_daftar\_siswa\_aktif\_instruktur\_kursus | |
| **Nama *Use Case*** | Menampilkan Daftar Siswa Aktif | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk menampilkan daftar siswa aktif yang dibawahi Aktor | |
| **Kondisi Awal** | Aktor berada di halaman kursus | |
| **Kondisi Akhir** | Sistem menampilkan semua siswa yang terkait dengan Aktor | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tab “Daftar Siswa Aktif” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman “Daftar Siswa Aktif” |
|  | 3. Sistem mengambil semua data siswa yang masih memiliki pertemuan kursus sampai hari ini |

##### Proses Menampilkan Detail *Progress* Kursus Siswa

Sama dengan Pemilik / Admin, Instruktur Kursus dapat melihat detail kursus dari siswa yang diajari. Halaman detail *progress* ini menunjukkan nama instruktur dan di pertemuan ke berapakah siswa saat ini, serta apakah siswa sudah membaca teori yang diberikan, apakah *quiz* yang diberikan sudah dikerjakan, dan capaian-capaian lain. Tabel 3.27 akan menjelaskan alur dari proses pihak kursus menampilkan detail *progress* siswa.

Tabel 3. *Use Case Scenario* untuk Menampilkan Detail *Progress* Kursus Siswa

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_akses\_detail\_progress\_kursus\_instruktur\_kursus | |
| **Nama *Use Case*** | Menampilkan Detail *Progress* Siswa | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk menampilkan detail kursus aktif | |
| **Kondisi Awal** | Aktor berada di daftar kelas kursus | |
| **Kondisi Akhir** | Aktor mengetahui informasi lebih lanjut tentang kursus siswa | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih salah satu siswa dari daftar jadwal yang ditampilkan di *dashboard* atau Daftar Siswa Aktif |  |
|  | 2. Sistem menampilkan data kursus (Nama Kelas Kursus, Instruktur yang bertugas, dll.) terkait dengan siswa yang dipilih Aktor |
|  | 3. Sistem menampilkan daftar capaian yang sudah dilakukan oleh siswa |

##### Proses Menghubungi Siswa atau Pemilik / Admin Kursus

Instruktur Kursus dapat menghubungi Siswa atau Pemilik / Admin melalui aplikasi yang selanjutnya akan diteruskan melalui Whatsapp. Nomor setiap Aktor didapatkan dari data pada akun profil masing-masing Aktor. Selanjutnya akan dijelaskan lebih jauh menggunakan tabel 3.28 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menghubungi Siswa atau Instruktur Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_komunikasi\_instruktur\_kursus | |
| **Nama *Use Case*** | Hubungi Siswa dan Pemilik / Admin Kursus | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk melakukan komunikasi daring antar Aktor | |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus Siswa | |
| **Kondisi Akhir** | Aktor berhasil sampai di tampilan percakapan Whatsapp dengan Aktor yang dimaksud | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Hubungi Pihak Lain” |  |
|  | 2. Sistem menampilkan dua opsi, untuk menghubungi Siswa atau Pemilik / Admin Kursus |
| 3. Aktor menekan opsi Aktor tujuan |  |
|  | 4. Sistem mengambil data Nomor Whatsapp dari *database* |
|  | 5. Sistem membuat link “wa.me/Nomor\_Telepon\_Aktor\_Tujuan” |
| 6. Aktor menekan tombol “Lanjut ke Chat” |  |

##### Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

Setiap Instruktur wajib memiliki sertifikat instruktur kursus mengemudi yang sah dan berlaku selama menjadi penyedia jasa kursus di aplikasi nantinya. Seandainya masa berlaku dokumen lama sudah habis atau dokumen sudah tidak berlaku, Instruktur kursus dapat mengunggah sertifikat terbarunya. Tabel 3.29 dibawah ini akan menjelaskan interaksi antara aktor dan respon sistem selama proses pengunggahan berlangsung.

Tabel 3. *Use Case Scenario* untuk Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_unggah\_sertifikat\_instruktur\_kursus | |
| **Nama *Use Case*** | Unggah Dokumen Sertifikat Instruktur Kursus | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk mengunggah dokumen sertifikat instruktur kursus | |
| **Kondisi Awal** | Aktor berada di halaman profil | |
| **Kondisi Akhir** | Dokumen sertifikat instruktur kursus yang baru tersimpan di *database* | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Sertifikat Instruktur” |  |
|  | 2. Sistem menampilkan *preview* dokumen yang tersimpan saat ini tanggal mulai dan tanggal akhir masa berlaku dokumen tersebut beserta status dokumen saat ini |
| 3. Aktor menekan tombol “Unggah Sertifikat Baru” |  |
|  | 4. Sistem menampilkan *form* untuk Aktor dapat mengunggah file baru dan satu kolom tambahan untuk tanggal awal & berakhir masa berlaku |
| 5. Aktor memilih *file* yang akan diunggah dan memilih tanggal awal & berakhir masa berlaku dokumen tersebut |  |

Tabel 3. 29 *Use Case Scenario* untuk Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 6. Sistem menyimpan inputan yang dimasukkan Aktor ke *database* |

##### Proses *Logout*

Proses *logout* dimulai dari pengguna menekan tombol “Logout” di *navbar* atau di bagian *footer* aplikasi. Tabel 3.30 dibawah akan menjelaskan alur proses *logout*.

Tabel 3. *Use Case Scenario* untuk proses *Logout*

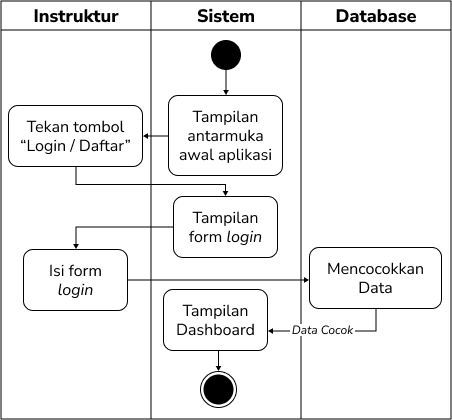
|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_logout\_instruktur\_kursus | |
| **Nama *Use Case*** | *Logout* | |
| **Aktor** | Instruktur Kursus | |
| **Deskripsi** | Skenario untuk melakukan proses *logout* | |
| **Kondisi Awal** | Aktor menekan tombol dengan label “Logout” | |
| **Kondisi Akhir** | Aktor berada di halaman awal aplikasi | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan pesan *pop-up* “Anda yakin ingin keluar dari aplikasi?” dan menampilkan dua tombol, “Ya” dan “Tidak” |
| 2. Aktor menekan tombol “Ya” |  |
|  | 3. Sistem menghentikan *session* dari akun Aktor |
|  | 4. Sistem mengarahkan Aktor ke halaman awal aplikasi |

#### Activity Diagram

Setelah pemaparan *use case scenario* diatas, yang menjelaskan interaksi antar Aktor yang terlibat dan bagaimana respon sistem terhadap interaksi tersebut. Selanjutnya, kami akan menjelaskan alur masing-masing aktifitas diatas dengan diagram aktifitas dibawah ini.

##### Activity Diagram untuk Proses Login

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.50 berikut adalah diagram aktifitas untuk proses *Login*.



Gambar 3. *Activity Diagram* untuk Proses *Login*

##### *Activity Diagram* untuk Proses Akses *Dashboard* Instruktur Kursus

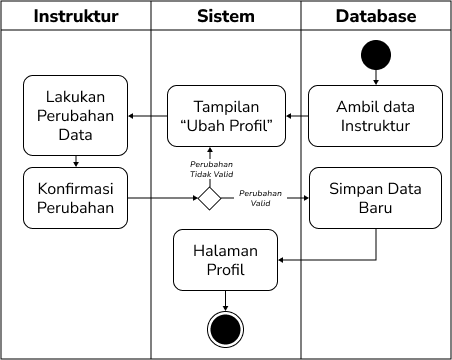
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.51 dibawah ini adalah diagram aktifitas akses *dashboard* instruktur kursus.



Gambar 3. *Activity Diagram* untuk Proses Akses *Dashboard* Instruktur Kursus

##### *Activity Diagram* untuk Proses Mengubah Data Profil

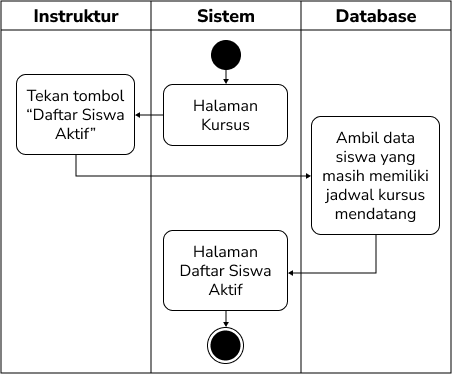
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.52 dibawah ini adalah diagram aktifitas mengubah data pada akun profil.



Gambar 3. *Activity Diagram* untuk Proses Mengubah Data Profil

##### *Activity Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

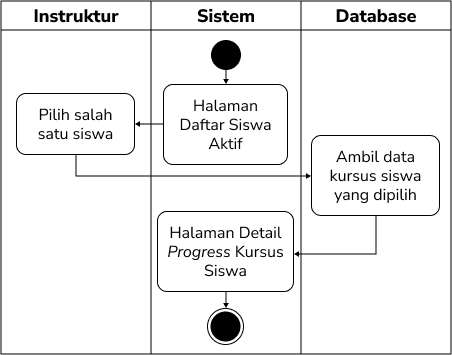
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.53 dibawah ini adalah diagram aktifitas menampilkan daftar siswa aktif.



Gambar 3. *Activity Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

##### *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

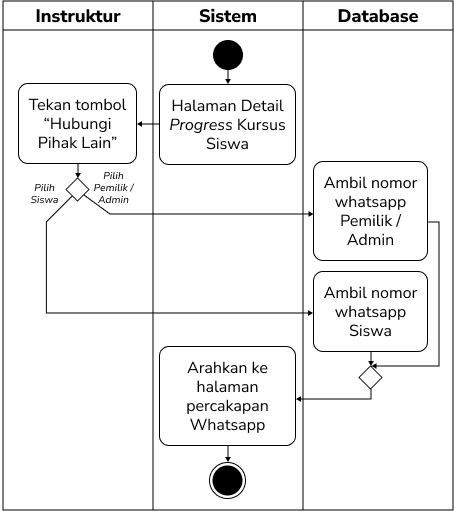
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.54 dibawah ini adalah diagram aktifitas menampilkan detail *progress* kursus Siswa.



Gambar 3. *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

##### *Activity Diagram* untuk Proses Menghubungi Siswa atau Pemilik / Admin Kursus

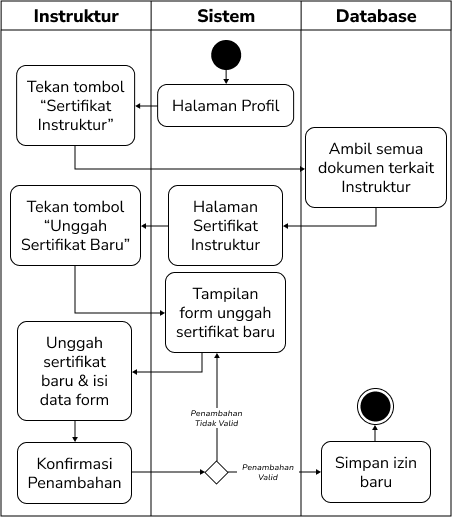
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.55 dibawah ini adalah diagram aktifitas menghubungi siswa atau pemilik / admin kursus.



Gambar 3. *Activity Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

##### *Activity Diagram* untuk Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

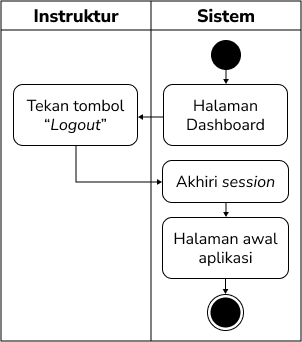
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.56 dibawah ini adalah diagram aktifitas mengunggah dokumen sertifikat instruktur kursus mengemudi.



Gambar 3. *Activity Diagram* untuk Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

##### Activity Diagram untuk Proses Logout

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.57 dibawah ini adalah diagram aktifitas *logout*.



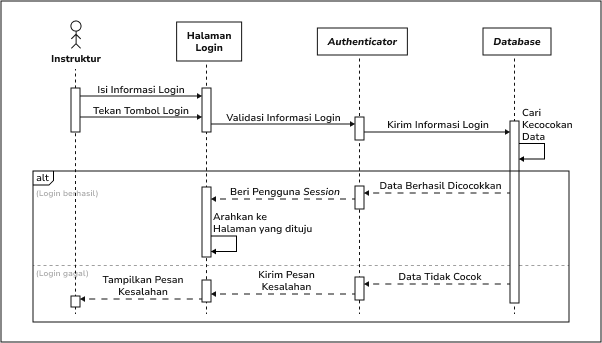
Gambar 3. *Activity Diagram* untuk Proses *Logout*

#### Sequence Diagram

Setelah pemaparan *use case scenario* dan *activity diagram* diatas, dibawah ini kami akan memaparkan jejak-jejak data ketika suatu proses berjalan menggunakan *sequence diagram*.

##### Sequence Diagram untuk Proses Login

Gambar 3.58 dibawah ini adalah alur sebuah data pada aplikasi ketika proses *login* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Login*

##### *Sequence Diagram* untuk Proses Akses *Dashboard* Instruktur Kursus

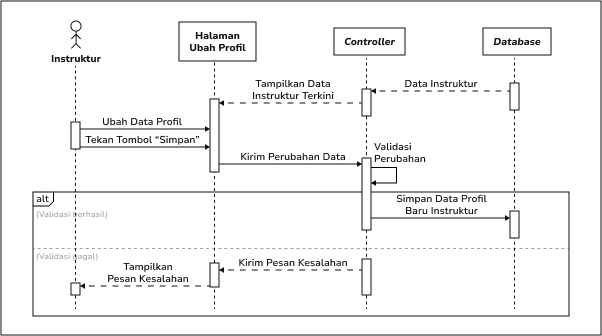
Gambar 3.59 dibawah ini adalah alur sebuah data pada aplikasi ketika proses akses *dashboard* instrukturdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Akses *Dashboard* Instruktur Kursus

##### *Sequence Diagram* untuk Proses Mengubah Data Profil

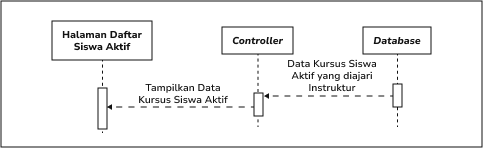
Gambar 3.60 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengubah data profildilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengubah Data Profil

##### *Sequence Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

Gambar 3.61 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan daftar siswa aktifdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menampilkan Daftar Siswa Aktif

##### *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

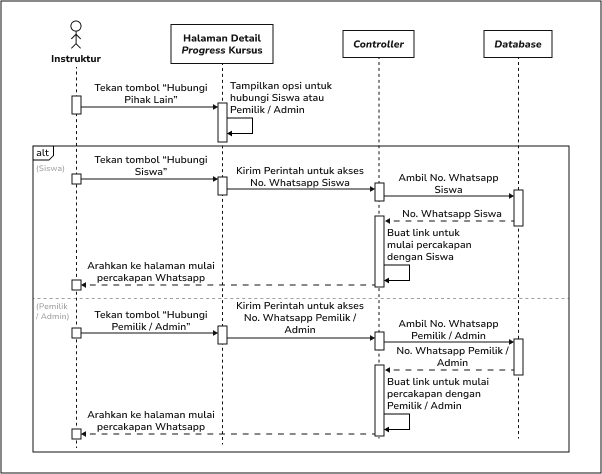
Gambar 3.62 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan detail *progress* kursus siswadilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus Siswa

##### *Sequence Diagram* untuk Proses Menghubungi Siswa atau Pemilik / Admin Kursus

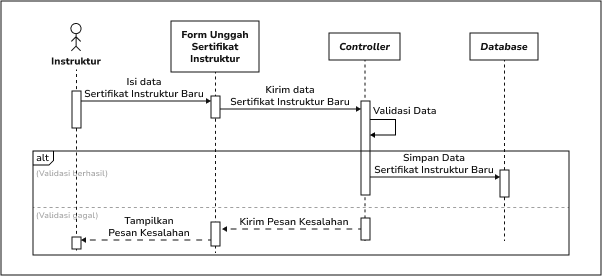
Gambar 3.63 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghubungi siswa atau pemilik / admin kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghubungi Siswa atau Pemilik / Admin Kursus

##### *Sequence Diagram* untuk Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

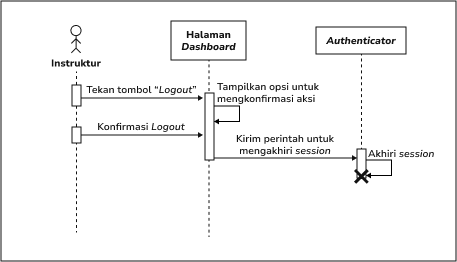
Gambar 3.64 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengunggah dokumen sertifikat instruktur kursus mengemudidilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi

##### Sequence Diagram untuk Proses Logout

Gambar 3.65 dibawah ini adalah alur sebuah data pada aplikasi ketika proses *logout* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Logout*

Proses selanjutnya dari rangkaian proses *incremental model* adalah pengembangan aplikasi berdasarkan desain yang sudah dirancang. Pengembangan ini dilakukan untuk menerjemahkan hasil desain ke bentuk aplikasi agar dapat dilakukan interaksi, yang selanjutnya akan dilakukan proses pengujian. Sebelum proses pengembangan dilakukan, sebagai acuan dalam proses pengembangan, berikut adalah gambaran atau skema relasi antar tabel yang akan dikembangkan menggunakan bantuan *class diagram* dari gambar 3.66 dibawah ini.



Gambar 3. *Class Diagram* Aplikasi *Build* 2

### Testing

Seperti pada *build* sebelumnya, untuk *build* 2 akan dilakukan pengujian untuk memvalidasi fungsi yang sudah dikembangkan menggunakan *Black Box*. Tabel 3.31 dibawah ini adalah skenario pengujian untuk *build* 2 aplikasi dari sisi Instruktur.

Tabel 3. *Black Box Testing* untuk *Build* 2

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses *Login* | | | |
| B2-T01 | Mengosongkan semua kolomkemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *input* | *Minor* |
| B2-T02 | Mengisi *username* / no. whatsapptetapi mengosongkan *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *password* | *Minor* |
| B2-T03 | Mengosongkan *username* / no. whatsapp tetapi mengisi *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *username* / no. whatsapp | *Minor* |
| B2-T04 | Mengisi *username* / no.whatsappyang belum terdaftar di *database* kemudian menekan tombol ‘Login’ | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B2-T05 | Pengguna menekan tombol ‘Login’ tapi *Password* salah | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B2-T06 | Mengisi *username* / no. whatsappdan *password* yang sesuai | Sistem mengarahkan pengguna ke halaman *dashboard* | *Major* |
| Proses Mengakses *Dashboard* Instruktur Kursus | | | |
| B2-T07 | Tidak ada kursus mendatang | Sistem menampilkan pesan bahwa tidak ada kursus mendatang | *Minor* |

Tabel 3. 31 *Black Box Testing* untuk *Build* 2 (Lanjutan-1)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Mengakses *Dashboard* Instruktur Kursus | | | |
| B2-T08 | Pada bagian jadwal per hari, tidak ada kursus pada hari yang dipilih | Sistem menampilkan pesan bahwa tidak ada kursus pada tanggal yang dipilih | *Minor* |
| B2-T09 | Pada bagian jadwal per hari, lihat daftar kursus di salah satu hari | Sistem memberikan tampilan yang berbeda untuk kursus yang sudah dilaksanakan, sedang berlangsung, dan akan berlangsung | *Minor* |
| Proses Mengubah Data Profil | | | |
| B2-T10 | Mengosongkan salah satu kolom yang wajib diisi pada halaman ubah profil dan menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* |
| B2-T11 | Mengunggah foto profil dengan format lain selain .jpg, .webp atau .png dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B2-T12 | Mengunggah foto profil dengan ukuran *file* terlalu besar (>2 MB) dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| B2-T13 | Menuliskan nama di kolom “Nama Instruktur Kursus” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |
| B2-T14 | Menuliskan nama di kolom “Username” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 31 *Black Box Testing* untuk *Build* 2 (Lanjutan-2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B2-T15 | Menuliskan *username* yang sudah ada di *database* dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Minor* | |
| B2-T16 | Menuliskan deskripsi akun di kolom “Deskripsi” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Major* | |
| B2-T17 | Menuliskan angka dibawah 18 pada kolom “Usia Instruktur” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia minimal adalah 18 tahun | *Minor* | |
| B2-T18 | Menuliskan angka diatas 70 pada kolom “Usia Instruktur” kemudian menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia maksimal adalah 99 tahun | *Minor* | |
| B2-T19 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif dan menekan tombol “Simpan” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Minor* | |
| B2-T20 | Mengetikkan No. Whatsapp melebihi 20 digit karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* | |
| B2-T21 | Mengetikkan No. Whatsapp yang sudah tersimpan di database dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* |
| B2-T22 | Menuliskan pertanyaan di kolom “Pertanyaan Tantangan” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa pertanyaan yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 31 *Black Box Testing* untuk *Build* 2 (Lanjutan-3)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B2-T23 | Menuliskan pertanyaan di kolom “Jawaban” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jawaban yang dimasukkan terlalu panjang | *Minor* |
| B2-T24 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |
| B2-T25 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B2-T26 | Menuliskan *password* yang berbeda di kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” dan menekan tombol “Simpan” | Sistem menampilkan kesalahan kepada pengguna untuk memastikan bahwa pengguna mengetikkan *password* yang sama | *Minor* |
| B2-T27 | Tanpa mengubah apapun, tekan tombol “Simpan” | Sistem tetap menyimpan data walaupun tidak ada perubahan dan mengarahkan ke halaman profil | *Trivial* |
| B2-T28 | Melakukan perubahan dengan benar dan sesuai kemudian menekan tombol “Simpan” | Sistem melakukan perubahan di *database* dan mengarahkan pengguna ke halaman profil | *Major* |
| B2-T29 | Melakukan perubahan kemudian menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman profil | *Minor* |

Tabel 3. 31 *Black Box Testing* untuk *Build* 2 (Lanjutan-4)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Menampilkan Daftar Siswa Aktif | | | | |
| B2-T30 | Tidak ada Siswa yang punya jadwal kursus mendatang | Sistem menampilkan pesan bahwa pengguna belum mempunyai siswa aktif | *Trivial* | |
| Proses Menampilkan Detail *Progress* Kursus Siswa | | | | |
| B2-T31 | Pada bagian informasi pertemuan | Sistem memberikan tampilan yang berbeda untuk pertemuan sudah dilaksanakan, pertemuan selanjutnya, dan sisa pertemuan mendatang | *Trivial* | |
| B2-T32 | Untuk siswa yang belum mengunggah bukti pembayaran | Sistem menampilkan peringatan bahwa siswa belum mengunggah bukti pembayaran | *Minor* | |
| B2-T33 | Untuk siswa yang sudah mengunggah bukti pembayaran tapi belum diverifikasi | Sistem menampilkan peringatan bahwa bukti pembayaran belum diverifikasi | *Minor* | |
| Proses Menghubungi Siswa atau Pemilik / Admin Kursus | | | | |
| B2-T34 | Memilih opsi “Hubungi Siswa” atau “Hubungi Admin” | Sistem mengarahkan pengguna ke halaman mulai percakapan Whatsapp | *Major* | |
| Proses Mengunggah Dokumen Sertifikat Instruktur Kursus Mengemudi | | | | |
| B2-T35 | Mengosongkan salah satu atau lebih kolom data kemudian menekan tombol “Ajukan Sertifikat Baru” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib | *Minor* | |

Tabel 3. 31 *Black Box Testing* untuk *Build* 2 (Lanjutan-5)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B2-T36 | Mengunggah sertifikat instruktur kursus dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Ajukan Sertifikat Baru” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* | |
| B2-T37 | Mengunggah sertifikat instruktur dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Ajukan Sertifikat Baru” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* | |
| B2-T38 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Ajukan Sertifikat Baru” | Sistem menyimpan data izin kursus baru ke *database* dan mengarahkan pengguna ke halaman sertifikat instruktur kursus untuk menampilkan sertifikat instruktur kursus yang baru ditambahkan ada di urutan pertama | *Major* | |
| B2-T39 | Menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman sertifikat instruktur kursus | *Minor* | |
| Proses *Log Out* | | | | |
| B2-T40 | Pemilik/Admin menekan tombol “Log Out” | Sistem menampilkan dialog *pop-up* untuk mengkonfirmasi aksi pengguna | *Major* | |
| B2-T41 | Pemilik/Admin menekan tombol “X” atau ‘Batal’ | Sistem menutup dialog *pop-up* | *Minor* | |
| B2-T42 | Pemilik/Admin menekan tombol ‘Ya’ | Sistem mengakhiri sesi pengguna dan mengarahkan pengguna ke halaman awal aplikasi | *Major* | |

### Implementation

Fase terakhir dari rangkaian proses pengembangan perangkat lunak dengan *incremental model* adalah mengintegrasikan *build* saat ini dengan *build* lainnya. Karena sebelumnya sudah dilakukan pengembangan aplikasi dari sisi Pemilik / Admin, selanjutnya, proses implementasi yang dapat kami lakukan adalah menggabungkan aplikasi dari *build* 1 dan *build* 2 serta memastikan kembali tidak ada *bug* atau kesalahan terjadi akibat penggabungan kedua *build* ini.

## Build 3 : Sisi General User (Siswa)

Proses *increment* yang terakhir adalah pengembangan aplikasi dari sisi *General User*. Pengguna selanjutnya kami sebut dengan *General User* karena pengguna ini selain dapat menjadi siswa dengan mendaftarkan dirinya ke salah satu kelas kursus yang ditawarkan, mereka juga dapat mendaftarkan dirinya sebagai penyedia jasa kursus mengemudi dengan mengunggah dokumen izin penyelenggaraan kursus yang sah. Sehingga, kedepannya, ketika kami menyebut *General User*, pembaca diharap mengerti bahwa pengguna adalah generalisasidari dua pihak tersebut.

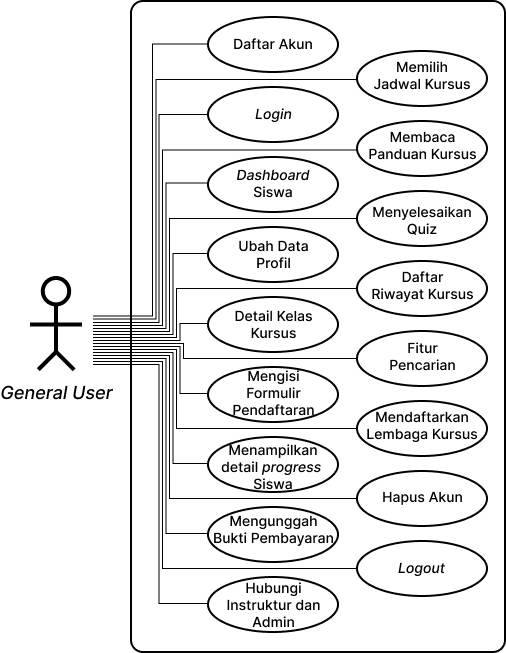
### Design & Development

Sama dengan rangkaian proses pada pengembangan aplikasi dari sisi Pemilik / Admin dan Instruktur Kursus yang sudah dilakukan sebelumnya. Kami menjabarkan analisa kebutuhan pada sub bab sebelumnya menggunakan tiga diagram UML yang akan kami jelaskan dibawah ini.

Perancangan menggunakan diagram UML ini adalah hasil dari penerjemahan fase perancangan yang lebih terstruktur dimana pada bagian pertama kami menjelaskan akses apa saja yang dimiliki oleh masing-masing pengguna sedangkan bagian kedua menjelaskan alur kerja sistem ketika sebuah fitur dieksekusi oleh pengguna, setelah tahap perancangan selesai, selanjutnya akan dilanjutkan dengan mengimplementasikan rancangan tersebut.

#### Use Case

Untuk *build* terakhir, fase desain dan pengembang dimulai dengan pemaparan diagram *use case* dan kemudian dilanjutkan dengan menyusun *use case scenario* untuk mempermudah tim pengembang memahami bagaimana struktur aplikasi, perilaku pengguna akhir, dan respon sistem saat suatu fitur sedang dieksekusi. Gambar 3.67 adalah *use case diagram* untuk pengguna *General User.*

**

Gambar 3. *Use Case Diagram* untuk *General User*

Sebagai permulaan, berikut adalah *use case diagram* untuk pengguna *General User*, penyebutan *General User* dilandasi asumsi bahwa pengguna dengan peran tersebut dianggap sebagai masyarakat umum, dimana mereka bisa mendaftar sebagai siswa kursus atau mereka bisa juga mendaftar sebagai penyedia jasa dengan mengunggah dokumen izin penyelenggaraan dan melakukan proses-proses yang dibutuhkan selanjutnya. Untuk lebih jelasnya akan kami jelaskan melalui *use case scenario* di sub-bab selanjutnya.

##### Proses Daftar Akun

Proses mendaftarkan akun dimulai dari pengguna berada pada halaman *login*. Proses pendaftaran akun dapat dilakukan apabila pengguna baru ingin menggunakan aplikasi. Tabel 3.32 dibawah ini akan menjelaskan alur pendaftaran akun.

Tabel 3. *Use Case Scenario* untuk proses Daftar Akun

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_daftar\_akun | |
| **Nama *Use Case*** | Daftar Akun | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk melakukan pendaftaran akun baru | |
| **Kondisi Awal** | Aktor berada di halaman *Login* | |
| **Kondisi Akhir** | Aktor berada di halaman Beranda | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Daftar Sekarang” |  |
|  | 2. Sistem menampilkan *form* Daftar Akun |
| 2. Aktor mengisi kolom-kolom untuk kebutuhan akun (contoh : *username*, no. Whatsapp, *password*, dll.) |  |
|  | 3. Sistem memvalidasi *input* Aktor |
|  | 4. Sistem mengarahkan Aktor ke halaman *Dashboard* |

##### Proses *Login*

Proses *login* dimulai dari pengguna berada pada halaman *login*. Proses *login* dibutuhkan untuk mencatat semua aktivitas yang dilakukan oleh pengguna. Tabel 3.33 dibawah ini menjelaskan tentang interaksi antar pengguna dengan respon sistem.

Tabel 3. *Use Case Scenario* untuk proses *Login*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_login\_general\_user | |
| **Nama *Use Case*** | *Login* | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk memberikan autentikasi kepada pengguna sehingga, setiap operasi yang dilakukan dapat dikenali oleh sistem | |
| **Kondisi Awal** | Aktor berada di halaman *Login* | |
| **Kondisi Akhir** | Aktor berhasil memasukkan *username* atau nomor whatsapp dan *password* dengan benar | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan *form* *Login* |
| 2. Aktor mengisi kolom *username* atau nomor whatsappdan *password* |  |
|  | 3. Sistem memeriksa inputan dari Aktor dengan data yang ada pada *Database* |
|  | 4. Sistem berhasil menemukan data yang cocok dengan inputan Aktor |
|  | 5. Sistem menampilkan *modals* “Login Berhasil” dan mengarahkan Aktor ke halaman selanjutnya |

##### Proses Akses Dashboard General User

*General User* akan melihat informasi kursus yang sedang diikuti di bagian awal, selain itu, sistem akan merekomendasikan kelas-kelas kursus dan penyedia-penyedia kursus yang mungkin menarik perhatian. Tabel 3.34 dibawah akan menjelaskan respon sistem akan interaksi yang dilakukan oleh Aktor.

Tabel 3. *Use Case Scenario* untuk Mengakses *Dashboard General User*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ dashboard\_general\_user | |
| **Nama *Use Case*** | *Dashboard* *General User* | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menampilkan *Dashboard* *General User* | |
| **Kondisi Awal** | Aktor berada di halaman *Dashboard* | |
| **Kondisi Akhir** | Sistem menampilkan informasi-informasi terkait kursus mengemudi | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan kelas kursus aktif yang diikuti Aktor |
|  | 2. Sistem menampilkan rekomendasi kelas kursus |
|  | 3. Sistem menampilkan rekomendasi penyedia kursus |

##### Proses Mengubah Data Profil

*General User* bisa mengubah data yang terkait dengan profil mereka, diantaranya adalah *username, password,* foto profil, nomor whatsapp, dll. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.35 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengubah Data Profil

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ubah\_data\_akun\_general\_user | |
| **Nama *Use Case*** | Ubah Data Profil | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk mengubah data akun Aktor | |
| **Kondisi Awal** | Aktor berada di halaman Profil | |
| **Kondisi Akhir** | Aktor berhasil mengubah data pada akun | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Ubah Profil” yang ada di halaman profil |  |
|  | 2. Sistem menampilkan *form* untuk mengganti elemen-elemen yang terkait dengan akun Aktor (*username*, *password*, Nama Aktor, Nomor Whatsapp, foto profil Aktor, dll.) |
| 3. Aktor mengubah data-data terkait dengan akun yang ingin diubah |  |
| 4. Aktor mengkonfirmasi perubahan data |  |
|  | 5. Sistem melakukan penyimpanan data baru ke *database* dan memperbarui data di halaman *profil* sesuai dengan data baru |

##### Proses Akses Detail Kelas Kursus

*General User* dapat melihat detail kelas kursus yang ditawarkan oleh penyedia kursus mengemudi sebelum memastikan untuk mendaftarkan diri ke kelas sesuai pilihan. Tabel 3.36 dibawah akan menjelaskan respon sistem akan interaksi yang dilakukan oleh Aktor.

Tabel 3. *Use Case Scenario* untuk Mengakses Detail Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ detail\_kelas\_kursus | |
| **Nama *Use Case*** | Detail Kelas Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menampilkan detail kelas kursus | |
| **Kondisi Awal** | Aktor berada di halaman *Dashboard* | |
| **Kondisi Akhir** | Sistem menampilkan informasi-informasi terkait kelas kursus yang dipilih | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih salah satu kelas di bagian rekomendasi kelas kursus |  |
|  | 2. Sistem menampilkan informasi terkait kelas kursus (contoh : Nama Kelas Kursus, Harga Kelas, Deskripsi Kelas, dll.) |
|  | 3. Sistem menampilkan Instruktur yang mengajar kelas kursus tersebut |
|  | 4. Sistem menampilkan rekomendasi kelas kursus serupa |

##### Proses Mengisi Formulir Pendaftaran Kelas Kursus

*General User* dapat melakukan pengisian formulir pendaftaran kelas kursus untuk mengikuti sebuah kelas. Setelah pengisian, *General User* selanjutnya akan diwajibkan untuk memilih jadwal dan melakukan pembayaran yang selanjutnya akan dibahas lebih detail di sub-bab masing-masing. Untuk detail alur pengisian formulir pendaftaran kelas kursus akan dijelaskan melalui tabel 3.37 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengisi Formulir Pendaftaran Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ isi\_formulir\_pendaftaran\_kelas\_kursus | |
| **Nama *Use Case*** | Mengisi Formulir Pendaftaran Kelas Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk mengisi formulir pendaftaran kelas kursus | |
| **Kondisi Awal** | Aktor berada di halaman detail kelas kursus | |
| **Kondisi Akhir** | Aktor berada di halaman detail *progress* kursus | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Daftar Kelas” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman formulir pendaftaran kelas kursus |
| 3. Aktor mengisi data-data terkait keperluan kursus dan memilih instruktur yang diinginkan dan menekan tombol “Daftar” |  |
|  | 4. Sistem memvalidasi *inputan* Aktor dan menyimpan data kursus yang terkait dengan Aktor dan mengarahkan Aktor ke halaman detail *progress* kursus |
|  | 5. Sistem menampilkan pesan “Pendaftaran Kursus Berhasil” |

##### Proses Menampilkan Detail *Progress* Kursus Siswa

Setelah mendaftarkan dirinya ke salah satu kelas kursus yang ada di aplikasi, *General User* dapat melihat detail *progress* kursus yang diikuti. Halaman detail *progress* ini menunjukkan nama instruktur dan di pertemuan ke berapakah Aktor saat ini, serta apakah Aktor sudah membaca teori yang diberikan, apakah *quiz* yang diberikan sudah dikerjakan, dan capaian-capaian lain. Tabel 3.38 akan menjelaskan alur dari menampilkan detail *progress* siswa.

Tabel 3. *Use Case Scenario* untuk Menampilkan Detail *Progress* Kursus Siswa

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_akses\_detail\_progress\_kursus\_general\_user | |
| **Nama *Use Case*** | Menampilkan Detail *Progress* Siswa | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menampilkan detail kursus aktif | |
| **Kondisi Awal** | Aktor baru saja mengisi formulir pendaftaran kelas kursus | |
| **Kondisi Akhir** | Aktor mengetahui informasi lebih lanjut tentang kursus yang diikuti | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem mengarahkan Aktor ke halaman detail *progress* kursus siswa |
|  | 2. Sistem menampilkan nama kelas kursus yang diikuti, instruktur yang dipilih, dan menampilkan peringatan jika Aktor belum mengunggah bukti pembayaran kursus |
|  | 3. Sistem menampilkan jadwal kursus saat ini dan daftar capaian siswa di setiap pertemuan |

##### Proses Mengunggah Bukti Pembayaran Kursus

Setelah *General User* melakukan pengisian formulir pendaftaran kelas kursus. Selanjutnya adalah proses pembayaran. Sekali lagi kami menekankan bahwa untuk mengakomodasi bermacam tipe pembayaran yang berbeda-beda yang diterapkan oleh setiap penyedia jasa kursus mengemudi, proses pembayaran hanya dapat dilakukan diluar sistem. Sistem hanya memberikan informasi tentang metode-metode yang diterima oleh para penyedia jasa kursus dan digunakan sebagai tempat untuk mengunggah bukti pembayaran dan menampilkan status pembayaran saat ini. Untuk detail alur pembayaran kelas kursus akan dijelaskan melalui tabel 3.39 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mengunggah Bukti Pembayaran Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_unggah\_bukti\_pembayaran | |
| **Nama *Use Case*** | Mengunggah Bukti Pembayaran Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk mengunggah bukti pembayaran | |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus siswa | |
| **Kondisi Akhir** | Aktor kembali berada di halaman detail *progress* kursus siswa | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Pembayaran” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman awal Pembayaran |
|  | 3. Sistem menampilkan metode pembayaran yang diterima oleh penyedia kursus |
| 4. Aktor menekan tombol “Unggah Bukti Pembayaran” |  |

Tabel 3. 39 *Use Case Scenario* untuk Mengunggah Bukti Pembayaran Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 5. Sistem menampilkan form untuk mengunggah bukti pembayaran kursus |
| 6. Aktor memilih dan mengunggah *file* bukti pembayaran |  |
|  | 7. Sistem memvalidasi *file* unggahan Aktor dan menyimpan unggahan ke *database* |
|  | 8. Sistem mengarahkan Aktor kembali ke halaman detail *progress* kursus siswa dan menampilkan pesan “Bukti Pembayaran berhasil diunggah” |

##### Proses Menghubungi Instruktur Kursus atau Pemilik / Admin Kursus

*General User* dapat menghubungi Instruktur atau Pemilik / Admin melalui aplikasi yang selanjutnya akan diteruskan melalui Whatsapp. Nomor setiap Aktor didapatkan dari data pada akun profil masing-masing Aktor. Selanjutnya akan dijelaskan lebih jauh menggunakan tabel 3.40 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Menghubungi Instruktur Kursus atau Pemilik / Admin Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_komunikasi\_general\_user |
| **Nama *Use Case*** | Hubungi Instruktur Kursus atau Pemilik / Admin Kursus |
| **Aktor** | *General User* |
| **Deskripsi** | Skenario untuk melakukan komunikasi daring antar Aktor |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus Siswa |

Tabel 3. 40 *Use Case Scenario* untuk Menghubungi Instruktur Kursus atau Pemilik / Admin Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Kondisi Akhir** | Aktor berhasil sampai di tampilan percakapan Whatsapp dengan Aktor yang dimaksud | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Hubungi Pihak Lain” |  |
|  | 2. Sistem menampilkan dua opsi, untuk menghubungi Instruktur Kursus atau Pemilik / Admin Kursus |
| 3. Aktor menekan opsi Aktor tujuan |  |
|  | 4. Sistem mengambil data Nomor Whatsapp dari *database* |
|  | 5. Sistem membuat link “wa.me/Nomor\_Telepon\_Aktor\_Tujuan” |
| 6. Aktor menekan tombol “Lanjut ke Chat” |  |

##### Proses Memilih Jadwal Kursus

*General User* berhak untuk memilih jadwal kursusnya sendiri, namun, jadwal yang dipilih oleh Aktor tetap tidak dapat bertabrakan dengan jadwal instruktur pengajar yang sudah ada. Jadwal kursus yang sudah dipilih hanya dapat diubah oleh Pemilik / Admin. Untuk lebih jelasnya, tabel 3.41 akan menjabarkan alurnya dengan detail.

Tabel 3. *Use Case Scenario* untuk Mengajukan Perubahan Jadwal Kursus

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_pilih\_jadwal\_kursus\_general\_user |
| **Nama *Use Case*** | Memilih Jadwal Kursus |

Tabel 3. 41 *Use Case Scenario* untuk Mengajukan Perubahan Jadwal Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk memilih jadwal kursus untuk pertama kali | |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus Siswa | |
| **Kondisi Akhir** | Jadwal kursus baru tersimpan ke *database* | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Jadwal Kursus” |  |
|  | 2. Sistem menampilkan form untuk memasukkan jadwal kursus dari pertemuan pertama sampai akhir |
| 3. Aktor mengisi tanggal kursus dan memilih opsi jam yang tersedia untuk setiap pertemuan kemudian menekan tombol “Ajukan” |  |
|  | 4. Sistem memvalidasi *input* Aktor |
|  | 5. Sistem menyimpan jadwal kursus yang dipilih ke *database* |
|  | 6. Sistem mengarahkan Aktor kembali ke halaman detail *progress* kursus dan menampilkan pesan “Jadwal Kursus berhasil dibuat” |

##### Proses Membaca Panduan Kursus

*General User* atau dalam hal ini Siswa, setelah menyelesaikan pendaftaran kelas kursus, walaupun belum menyelesaikan pembayaran, Siswa sudah dapat mengakses panduan berkendara. Fitur ini ditujukan untuk memberikan pemahaman lebih dalam tentang teknik-teknik berkendara khususnya bagi Siswa pemula. Tabel 3.42 dibawah akan menjelaskan alur interaksi Aktor dengan respon yang diberikan oleh Sistem saat proses akses panduan berkendara dieksekusi.

Tabel 3. *Use Case Scenario* untuk Membaca Panduan Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_baca\_panduan\_kursus | |
| **Nama *Use Case*** | Membaca Panduan Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk membaca panduan kursus | |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus | |
| **Kondisi Akhir** | Daftar capaian untuk Baca Panduan ditandai sebagai selesai | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Baca Panduan” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman“Panduan Kursus” untuk pertemuan saat ini |
|  | 3. Sistem menampilkan materi sesuai dengan pertemuan saat ini |
| 4. Aktor selesai membaca semua panduan dan menekan tombol “Selesai” |  |
|  | 5. Sistem mengembalikan Aktor ke halaman “Detail *Progress* Kursus” dan memberikan *checklist* di daftar capaian “Baca Panduan” |

##### Proses Menyelesaikan *Quiz*

Jika *General User* selesai membaca panduan berkendara, selanjutnya Aktor dapat mengakses *Quiz* untuk menguji pemahamannya. Meskipun lazimnya Aktor harus membaca panduan terlebih dahulu kemudian mengakses *Quiz*, proses membaca panduan tetap dapat dilewati dan *Quiz* masih dapat diakses walaupun Aktor belum membaca panduan kursus. Tabel 3.43 dibawah akan menjelaskan alur interaksi Aktor dengan respon yang diberikan oleh Sistem saat proses menyelesaikan *quiz* dieksekusi.

Tabel 3. *Use Case Scenario* untuk Menyelesaikan *Quiz*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_quiz | |
| **Nama *Use Case*** | Menyelesaikan *Quiz* | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menyelesaikan *quiz* | |
| **Kondisi Awal** | Aktor berada di halaman detail *progress* kursus | |
| **Kondisi Akhir** | Daftar capaian untuk Selesaikan Quiz ditandai sebagai selesai | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Quiz” |  |
|  | 2. Sistem mengarahkan Aktor ke halaman“Quiz” |
|  | 3. Sistem menampilkan pertanyaan-pertanyaan dan pilihan jawaban sesuai dengan pertemuan saat ini |
| 4. Aktor selesai menjawab semua pertanyaan |  |
|  | 5. Sistem mengembalikan Aktor ke halaman “Detail *Progress* Kursus” dan memberikan *checklist* di daftar capaian “Selesaikan Quiz” |

##### Proses Akses Daftar Riwayat Kursus

*General User* dapat mengikuti lebih dari satu kelas kursus. Oleh karena itu, kami rasa penting untuk memberikan Aktor fitur ini, dimana Aktor nantinya dapat mengetahui *progress* setiap kursus yang pernah diikuti dan informasi-informasi terkait kursus yang pernah diikuti yang lain. Tabel 3.44 dibawah akan menjelaskan alur interaksi Aktor dengan respon yang diberikan oleh Sistem.

Tabel 3. *Use Case Scenario* untuk Mengakses Daftar Riwayat Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ riwayat\_kursus | |
| **Nama *Use Case*** | Daftar Riwayat Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menampilkan Daftar Riwayat Kursus | |
| **Kondisi Awal** | Aktor berada di halaman Profil | |
| **Kondisi Akhir** | Sistem menampilkan kelas-kelas kursus yang pernah diikuti Aktor | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan kelas kursus aktif yang diikuti Aktor |
|  | 2. Sistem menampilkan rekomendasi kelas kursus |
|  | 3. Sistem menampilkan rekomendasi penyedia kursus |

##### Proses Pencarian Kelas Kursus atau Penyedia Kursus

*General User* yang ingin mencari kelas kursus atau penyedia kursus secara spesifik bisa menggunakan fitur pencarian ini. Tabel 3.45 berikut akan menjelaskan alur interaksi Aktor dengan respon yang diberikan oleh Sistem.

Tabel 3. *Use Case Scenario* untuk Mengakses Daftar Riwayat Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ cari\_kursus | |
| **Nama *Use Case*** | Mencari Kelas Kursus atau Penyedia Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk melakukan pencarian kelas kursus atau penyedia kursus | |
| **Kondisi Awal** | Aktor berada di halaman *Dashboard* | |
| **Kondisi Akhir** | Sistem menampilkan hasil pencarian kelas kursus atau penyedia kursus | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol pencarian dibagian atas kanan aplikasi |  |
|  | 2. Sistem mengarahkan Aktor ke halaman pencarian. Jika Aktor sudah pernah melakukan pencarian, tampilkan riwayat pencarian Aktor |
| 3. Aktor memasukkan kata kunci pencarian di kolom pencarian |  |
|  | 4. Sistem melakukan pencarian kelas kursus dan penyedia kursus berdasarkan kata kunci.  Sistem menyertakan penyedia kursus yang kelas kursusnya masuk ke hasil pencarian |
|  | 4. Sistem menampilkan kata kunci yang dicari Aktor dan hasil pencarian |

##### Proses Mendaftarkan Diri sebagai Penyedia Kursus

Untuk menjadi penyedia jasa kursus baru di aplikasi, *General User* harus mengunggah dokumen izin penyelenggaraan kursus mengemudi terlebih dahulu. Dokumen ini nantinya akan diverifikasi oleh admin sistem secara manual. Jika berhasil, pengguna selanjutnya diarahkan untuk memperbarui informasi di akun Aktor dan menambahkan metode pembayaran yang diterima sehingga langkah selanjutnya adalah menambahkan kelas kursus dan instruktur yang ditugaskan. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.46 dibawah ini.

Tabel 3. *Use Case Scenario* untuk Mendaftarkan Diri sebagai Penyedia Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ daftar\_penyedia\_kursus | |
| **Nama *Use Case*** | Mendaftarkan Diri sebagai Penyedia Kursus | |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk mendaftarkan diri sebagai penyedia kursus | |
| **Kondisi Awal** | Aktor berada di halaman Profil | |
| **Kondisi Akhir** | Aktor berada di *Dashboard* Pemilik / Admin | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Penyedia Kursus Mengemudi?” |  |
|  | 2. Sistem mengarahkan pengguna ke halaman awal Ajukan Kursus Mengemudi Baru dan menampilkan syarat-syarat menjadi penyedia kursus mengemudi |
| 3. Aktor menekan tombol “Unggah Izin Kursus” |  |
|  | 4. Sistem menampilkan *form* untuk Aktor dapat mengunggah |

Tabel 3. 46 *Use Case Scenario* untuk Mendaftarkan Diri sebagai Penyedia Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | file baru dan satu kolom tambahan untuk tanggal awal & berakhir masa berlaku |
| 5. Aktor memilih *file* yang akan diunggah dan memilih tanggal awal & berakhir masa berlaku dokumen tersebut kemudian menekan tombol “Kirim Izin Kursus” |  |
|  | 6. Sistem memvalidasi inputan dan menyimpan data ke *database* kemudian mengarahkan Aktor kembali ke halaman Profil |
| 7. Aktor menekan tombol “Penyedia Kursus Mengemudi?” lagi untuk memeriksa apakah izin kursus hasil verifikasi |  |
|  | 8. Sistem menampilkan *preview* dokumen yang diunggah dan status verifikasi dokumen berhasil |
| 9. Aktor menekan tombol “Lanjut” |  |
|  | 10. Sistem mengarahkan Aktor ke halaman Informasi Akun dan menampilkan form Informasi Lembaga Kursus |
| 11. Aktor mengubah data yang diinginkan dan mengisi Jam Buka & Jam Tutup kemudian menekan tombol “Simpan” |  |

Tabel 3. 46 *Use Case Scenario* untuk Mendaftarkan Diri sebagai Penyedia Kursus (Lanjutan-2)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 12. Sistem memvalidasi inputan dan menyimpan data akun baru kemudian mengarahkan Aktor ke halaman Tambah Metode Pembayaran |
| 13. Aktor mengisi form Tambah Metode Pembayaran dengan benar dan menekan tombol “Simpan” |  |
|  | 14. Sistem memvalidasi inputan dan menyimpan data metode pembayaran kemudian mengarahkan Aktor ke halaman *dashboard* Pemilik / Admin |

##### Proses Menghapus Akun

*General User* dapat menghapus akun dari aplikasi, namun, Aktor tidak bisa menghapus akun ditengah mengikuti kursus. Tabel 3.47 dibawah akan menjelaskan alur proses menghapus akun.

Tabel 3. *Use Case Scenario* untuk Menghapus Akun

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_hapus\_akun |
| **Nama *Use Case*** | Menghapus Akun |
| **Aktor** | *General User* |
| **Deskripsi** | Skenario untuk menghapus akun |
| **Kondisi Awal** | Aktor berada di halaman profil |
| **Kondisi Akhir** | Aktor berada di halaman awal aplikasi |

Tabel 3. 47 *Use Case Scenario* untuk Menghapus Akun (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor memilih menu “Hapus Akun KEMUDI” |  |
|  | 2. Sistem menampilkan *pop-up* untuk mengkonfirmasi apakah Aktor yakin ingin menghapus akun |
| 3. Aktor memilih opsi “Ya, Hapus Akun” |  |
|  | 4. Sistem menghapus data-data Aktor dan mengarahkan Aktor ke halaman awal aplikasi |

##### Proses *Logout*

Proses *logout* dimulai dari pengguna menekan tombol “Logout” di *navbar* atau di bagian *footer* aplikasi. Tabel 3.48 dibawah akan menjelaskan alur proses *logout*.

Tabel 3. *Use Case Scenario* untuk proses *Logout*

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_logout\_pemilik\_kursus | |
| **Nama *Use Case*** | *Logout* | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk melakukan proses *logout* | |
| **Kondisi Awal** | Aktor menekan tombol dengan label “Logout” | |
| **Kondisi Akhir** | Aktor berada di halaman awal aplikasi | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan pesan *pop-up* “Anda yakin ingin keluar dari aplikasi?” dan menampilkan dua tombol, “Ya” |

Tabel 3. 48 *Use Case Scenario* untuk proses *Logout* (Lanjutan-1)

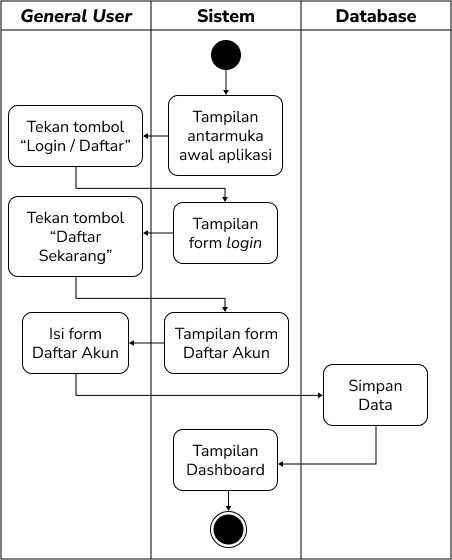
|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | dan “Tidak” |
| 2. Aktor menekan tombol “Ya” |  |
|  | 3. Sistem menghentikan *session* dari akun Aktor |
|  | 4. Sistem mengarahkan Aktor ke halaman awal aplikasi |

#### Activity Diagram

Setelah pemaparan *use case scenario* diatas, yang menjelaskan interaksi antar Aktor yang terlibat dan bagaimana respon sistem terhadap interaksi tersebut. Selanjutnya, kami akan menjelaskan alur masing-masing aktifitas diatas dengan diagram aktifitas dibawah ini.

##### *Activity Diagram* untuk Proses Daftar Akun

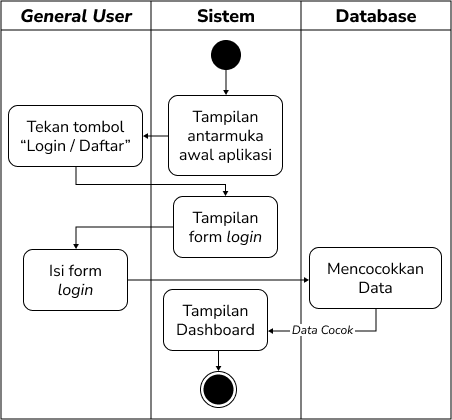
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.68 berikut adalah diagram aktifitas untuk proses daftar akun.



Gambar 3. *Activity Diagram* untuk Proses *Login*

##### Activity Diagram untuk Proses Login

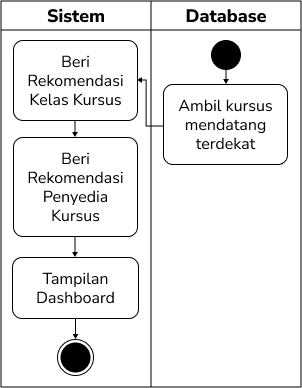
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.69 berikut adalah diagram aktifitas untuk proses *Login*.



Gambar 3. *Activity Diagram* untuk Proses *Login*

##### Activity Diagram untuk Proses Akses Dashboard General User

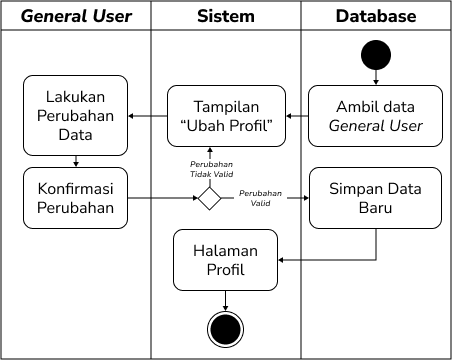
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.70 dibawah ini adalah diagram aktifitas akses *dashboard* *general user*.



Gambar 3. *Activity Diagram* untuk Proses Akses *Dashboard General User*

##### *Activity Diagram* untuk Proses Mengubah Data Profil

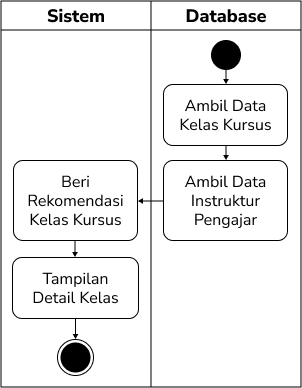
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.71 dibawah ini adalah diagram aktifitas mengubah data pada akun profil.



Gambar 3. *Activity Diagram* untuk Proses Mengubah Data Profil

##### *Activity Diagram* untuk Proses Akses Detail Kelas Kursus

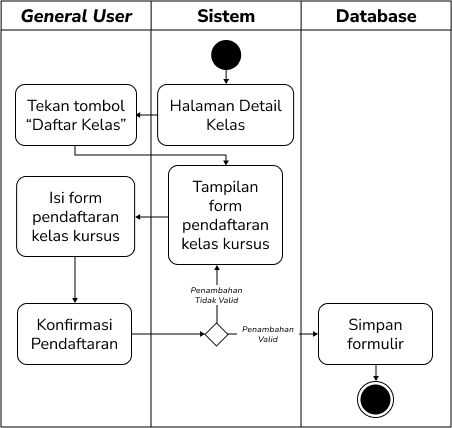
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.72 dibawah ini adalah diagram aktifitas akses detail kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Akses Detail Kelas Kursus

##### *Activity Diagram* untuk Proses Mengisi Formulir Pendaftaran Kelas Kursus

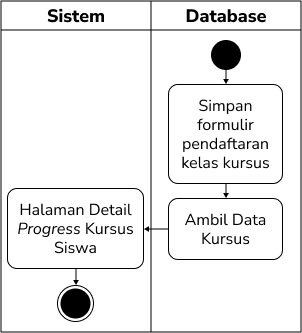
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.73 dibawah ini adalah diagram aktifitas akses mengisi formulir pendaftaran kelas kursus.



Gambar 3. *Activity Diagram* untuk Proses Mengisi Formulir Pendaftaran Kelas Kursus

##### *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus

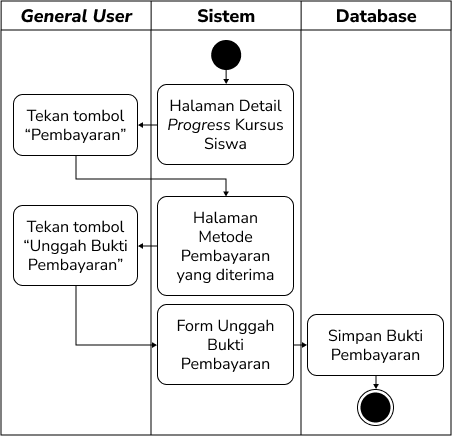
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.74 dibawah ini adalah diagram aktifitas menampilkan detail *progress* kursus.



Gambar 3. *Activity Diagram* untuk Proses Menampilkan Detail *Progress* Kursus

##### *Activity Diagram* untuk Proses Mengunggah Bukti Pembayaran Kursus

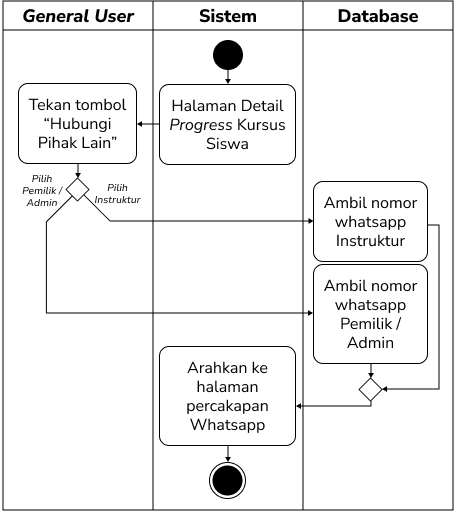
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.75 dibawah ini adalah diagram aktifitas mengunggah bukti pembayaran kursus.



Gambar 3. *Activity Diagram* untuk Proses Mengunggah Bukti Pembayaran Kursus

##### *Activity Diagram* untuk Proses Menghubungi Instruktur atau Pemilik / Admin Kursus

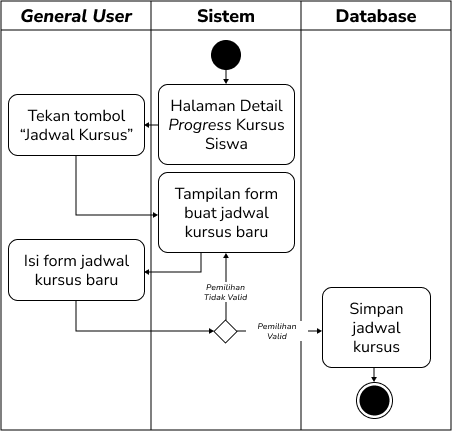
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.76 dibawah ini adalah diagram aktifitas menghubungi instruktur atau pemilik / admin kursus.



Gambar 3. *Activity Diagram* untuk Proses Menghubungi Siswa atau Instruktur Kursus

##### *Activity Diagram* untuk Proses Memilih Jadwal Kursus

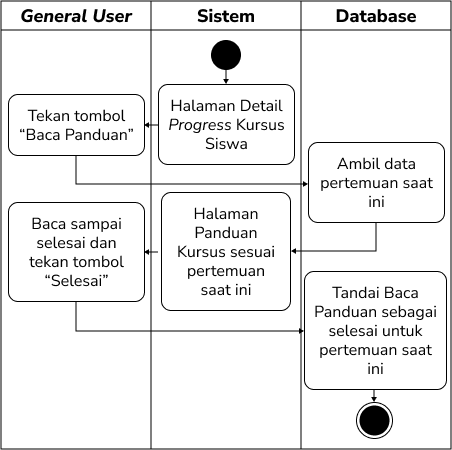
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.77 dibawah ini adalah diagram aktifitas memilih jadwal kursus.



Gambar 3. *Activity Diagram* untuk Proses Memilih Jadwal Kursus

##### *Activity Diagram* untuk Proses Membaca Panduan Kursus

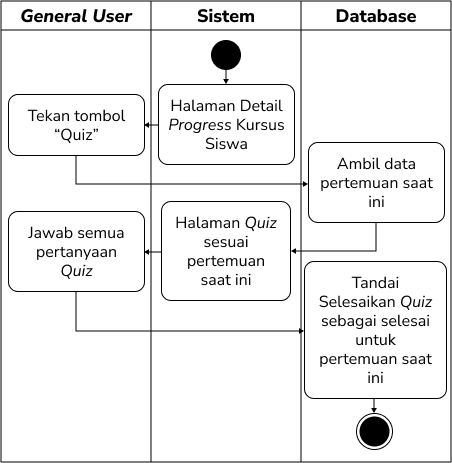
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.78 dibawah ini adalah diagram aktifitas membaca panduan kursus.



Gambar 3. *Activity Diagram* untuk Proses Membaca Panduan Kursus

##### *Activity Diagram* untuk Proses Menyelesaikan *Quiz*

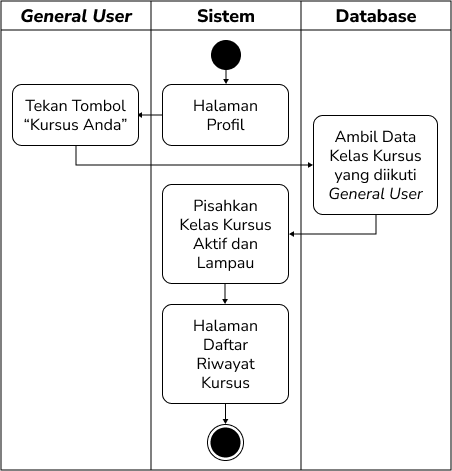
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.79 dibawah ini adalah diagram aktifitas menyelesaikan *quiz*.



Gambar 3. *Activity Diagram* untuk Proses Menyelesaikan *Quiz*

##### *Activity Diagram* untuk Proses Akses Daftar Riwayat Kursus

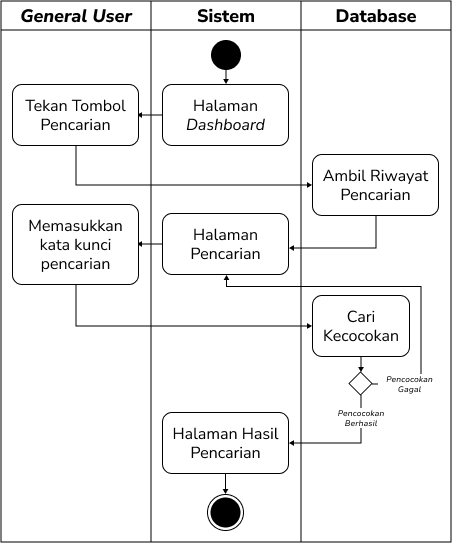
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.80 dibawah ini adalah diagram aktifitas akses daftar riwayat kursus.



Gambar 3. *Activity Diagram* untuk Proses Akses Daftar Riwayat Kursus

##### *Activity Diagram* untuk Proses Pencarian Kelas Kursus atau Penyedia Kursus

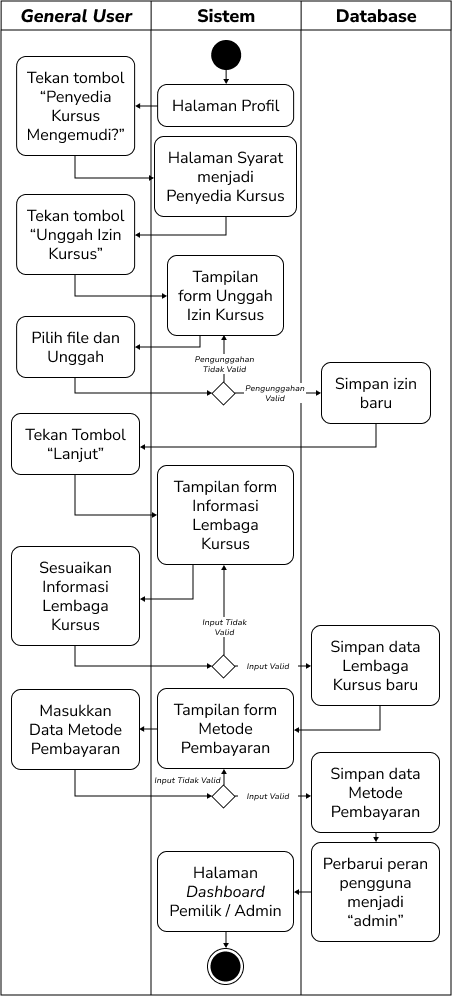
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.81 dibawah ini adalah diagram aktifitas pencarian kelas kursus atau penyedia kursus.



Gambar 3. *Activity Diagram* untuk Proses Pencarian Kelas Kursus atau Penyedia Kursus

##### *Activity Diagram* untuk Proses Mendaftarkan Diri sebagai Penyedia Kursus

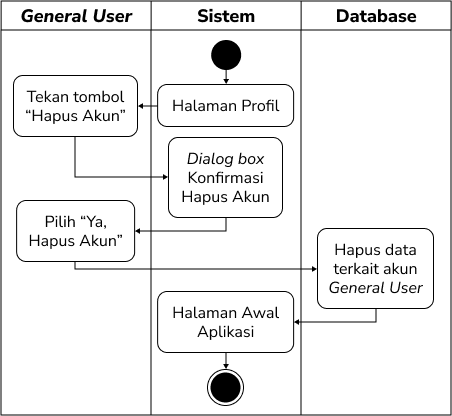
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.82 dibawah ini adalah diagram aktifitas mendaftarkan diri sebagai penyedia kursus.



Gambar 3. *Activity Diagram* untuk Proses Mendaftarkan Diri sebagai Penyedia Kursus

##### *Activity Diagram* untuk Proses Menghapus Akun

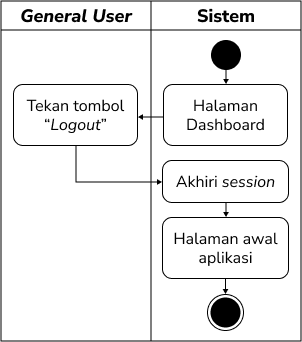
Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.83 dibawah ini adalah diagram aktifitas menghapus akun.



Gambar 3. *Activity Diagram* untuk Proses Menghapus Akun

##### Activity Diagram untuk Proses Logout

Untuk melengkapi penjelasan dari *use case scenario* sebelumnya, Gambar 3.84 dibawah ini adalah diagram aktifitas *logout*.



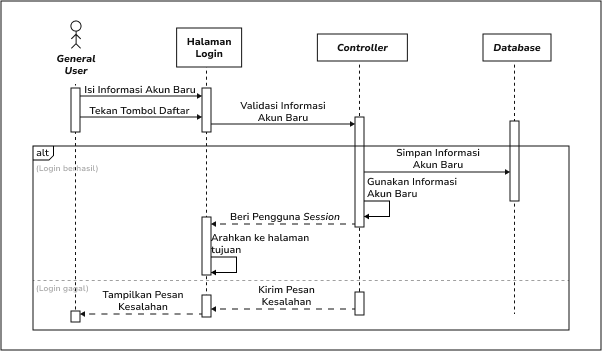
Gambar 3. *Activity Diagram* untuk Proses *Logout*

#### Sequence Diagram

Setelah pemaparan *use case scenario* dan *activity diagram* diatas, dibawah ini kami akan memaparkan jejak-jejak data ketika suatu proses berjalan menggunakan *sequence diagram*.

##### *Sequence Diagram* untuk Proses Daftar Akun

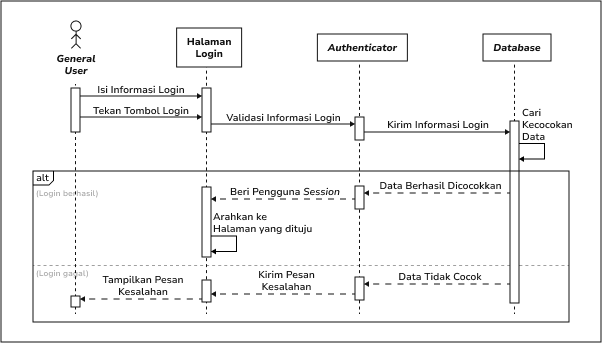
Gambar 3.85 dibawah ini adalah alur sebuah data pada aplikasi ketika proses daftar akundilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Daftar Akun

##### Sequence Diagram untuk Proses Login

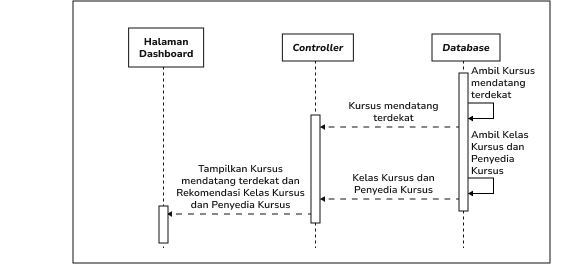
Gambar 3.86 dibawah ini adalah alur sebuah data pada aplikasi ketika proses *login* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Login*

##### Sequence Diagram untuk Proses Akses Dashboard General User

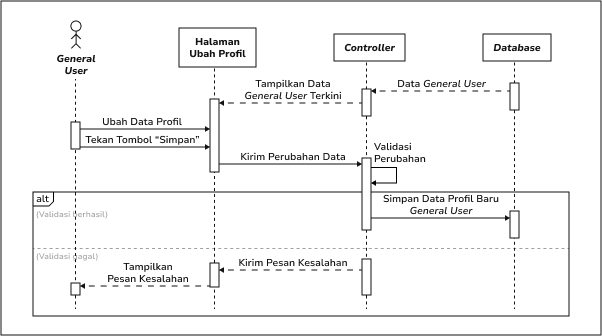
Gambar 3.87 dibawah ini adalah alur sebuah data pada aplikasi ketika proses akses *dashboard general user* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Akses *Dashboard General User*

##### *Sequence Diagram* untuk Proses Mengubah Data Profil

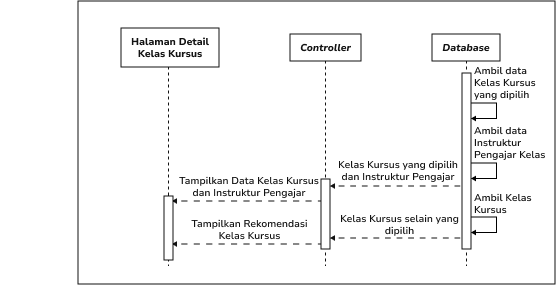
Gambar 3.88 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengubah data profildilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengubah Data Profil

##### *Sequence Diagram* untuk Proses Akses Detail Kelas Kursus

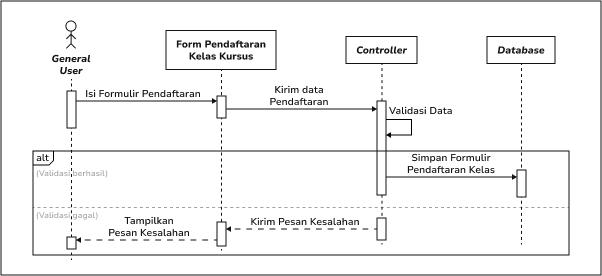
Gambar 3.89 dibawah ini adalah alur sebuah data pada aplikasi ketika proses akses detail kelas kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Akses Detail Kelas Kursus

##### *Sequence Diagram* untuk Proses Mengisi Formulir Pendaftaran Kelas Kursus

Gambar 3.90 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengisi formulir pendaftaran kelas kursus dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengisi Formulir Pendaftaran Kelas Kursus

##### *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus

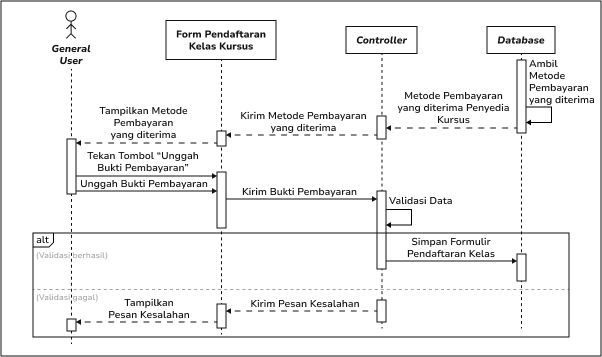
Gambar 3.91 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menampilkan detail *progress* kursus dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menampilkan Detail *Progress* Kursus

##### *Sequence Diagram* untuk Proses Mengunggah Bukti Pembayaran Kursus

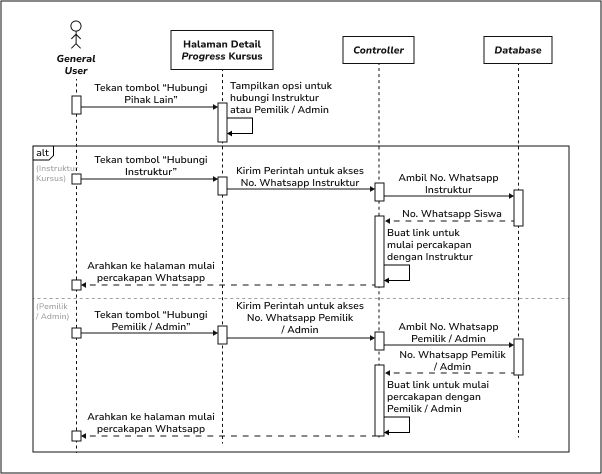
Gambar 3.92 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mengunggah bukti pembayaran kursus dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mengunggah Bukti Pembayaran Kursus

##### *Sequence Diagram* untuk Proses Menghubungi Instruktur atau Pemilik / Admin Kursus

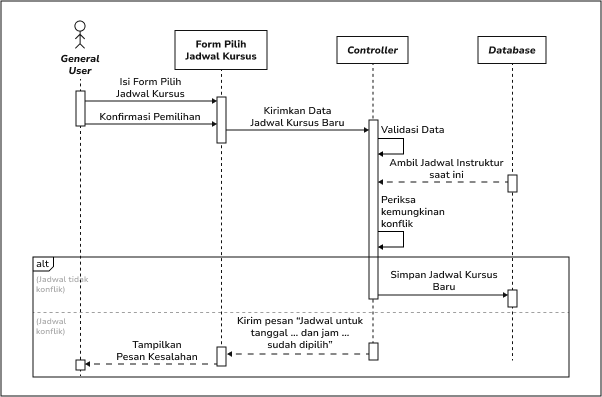
Gambar 3.93 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghubungi instruktur atau pemilik / admin kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghubungi Instruktur atau Pemilik / Admin Kursus

##### *Sequence Diagram* untuk Proses Memilih Jadwal Kursus

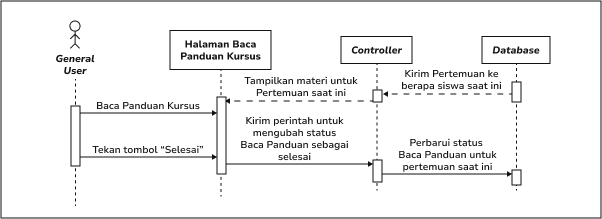
Gambar 3.94 dibawah ini adalah alur sebuah data pada aplikasi ketika proses memilih jadwal kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Memilih Jadwal Kursus

##### *Sequence Diagram* untuk Proses Membaca Panduan Kursus

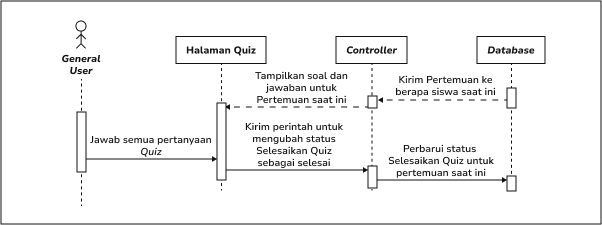
Gambar 3.95 dibawah ini adalah alur sebuah data pada aplikasi ketika proses membaca panduan kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Membaca Panduan Kursus

##### *Sequence Diagram* untuk Proses Menyelesaikan *Quiz*

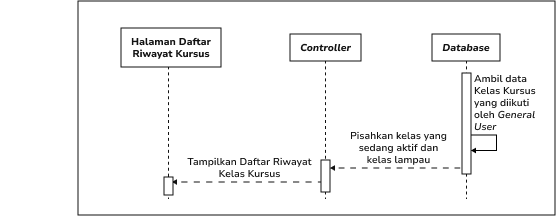
Gambar 3.96 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menyelesaikan *quiz* dilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menyelesaikan *Quiz*

##### *Sequence Diagram* untuk Proses Akses Daftar Riwayat Kursus

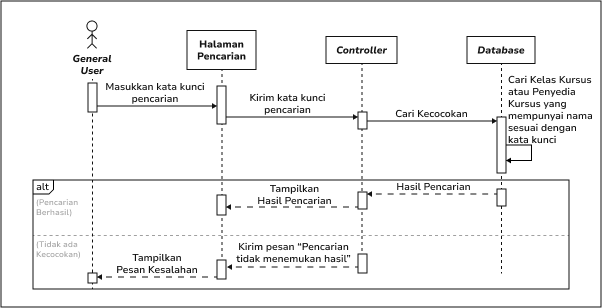
Gambar 3.97 dibawah ini adalah alur sebuah data pada aplikasi ketika proses akses daftar riwayat kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Akses Daftar Riwayat Kursus

##### *Sequence Diagram* untuk Proses Pencarian Kelas Kursus atau Penyedia Kursus

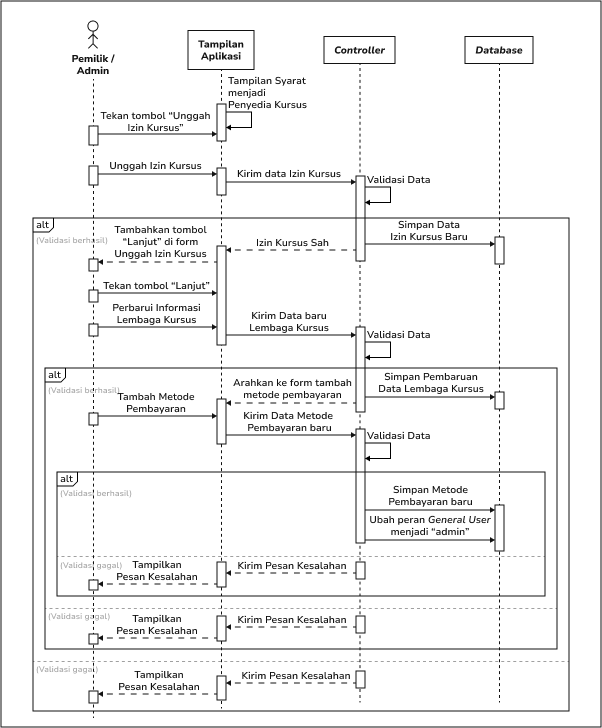
Gambar 3.98 dibawah ini adalah alur sebuah data pada aplikasi ketika proses pencarian kelas kursus atau penyedia kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Pencarian Kelas Kursus atau Penyedia Kursus

##### *Sequence Diagram* untuk Proses Mendaftarkan Diri sebagai Penyedia Kursus

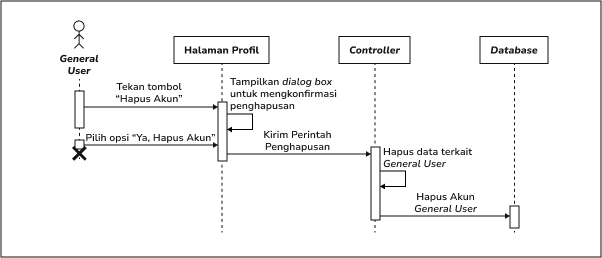
Gambar 3.99 dibawah ini adalah alur sebuah data pada aplikasi ketika proses mendaftarkan diri sebagai penyedia kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Mendaftarkan Diri sebagai Penyedia Kursus

##### *Sequence Diagram* untuk Proses Menghapus Akun

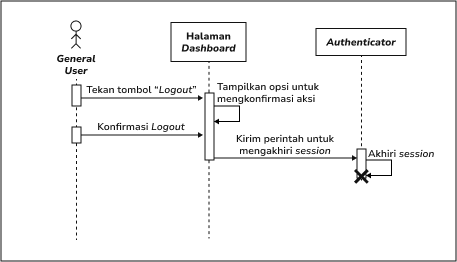
Gambar 3.100 dibawah ini adalah alur sebuah data pada aplikasi ketika proses menghapus akundilakukan.



Gambar 3. *Sequence Diagram* untuk Proses Menghapus Akun

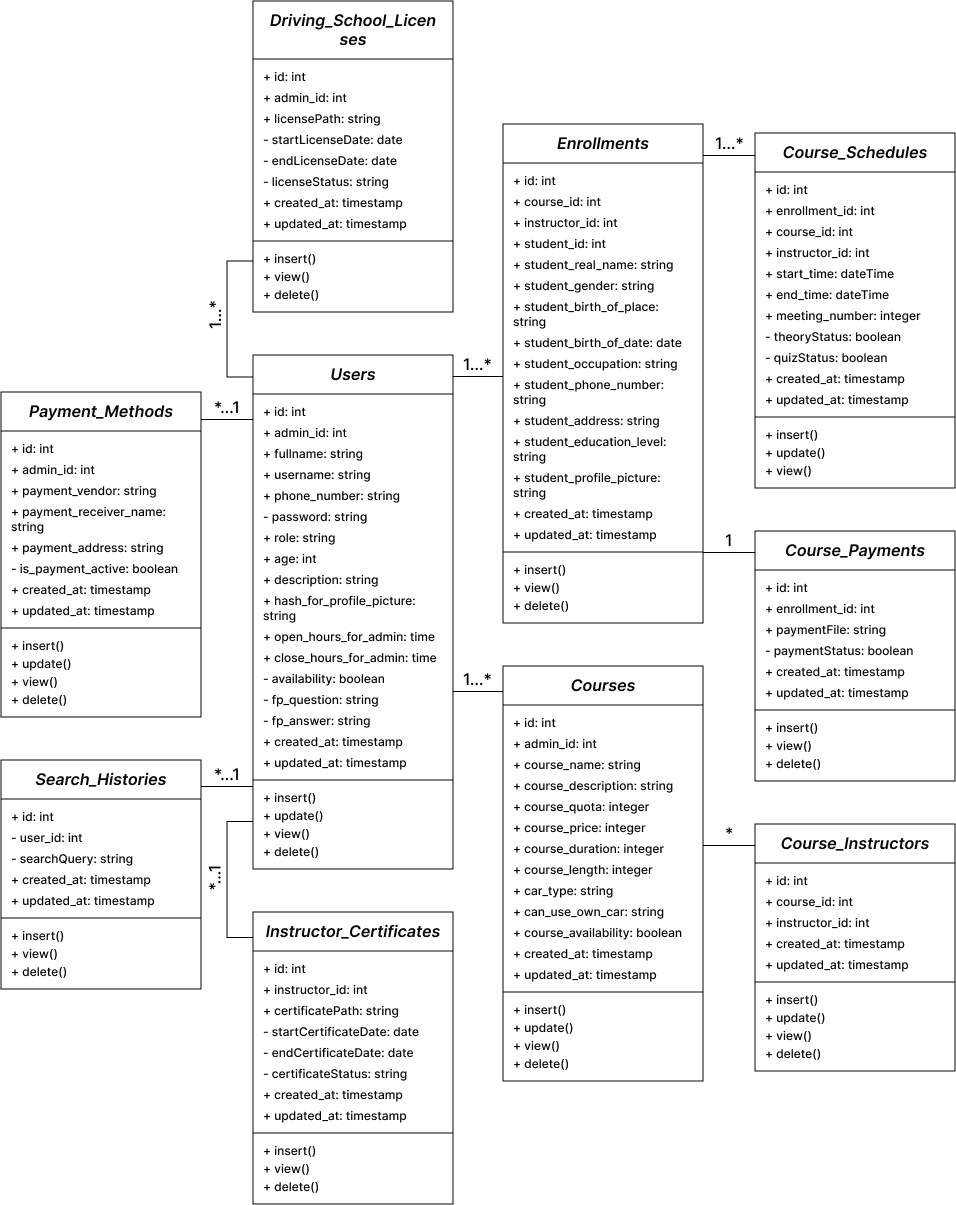
##### Sequence Diagram untuk Proses Logout

Gambar 3.101 dibawah ini adalah alur sebuah data pada aplikasi ketika proses pencarian kelas kursus atau penyedia kursusdilakukan.



Gambar 3. *Sequence Diagram* untuk Proses *Logout*

Proses selanjutnya dari rangkaian proses *incremental model* adalah pengembangan aplikasi berdasarkan desain yang sudah dirancang. Pengembangan ini dilakukan untuk menerjemahkan hasil desain ke bentuk aplikasi agar dapat dilakukan interaksi, yang selanjutnya akan dilakukan proses pengujian. Sebelum proses pengembangan dilakukan, sebagai acuan dalam proses pengembangan, berikut adalah gambaran atau skema relasi antar tabel yang akan dikembangkan menggunakan bantuan *class diagram* dari gambar 3.102 dibawah ini.



Gambar 3. *Class Diagram* Aplikasi *Build* 3

### Testing

Untuk *build* terakhir sama seperti pada *build* sebelumnya, untuk *build* 3 akan dilakukan pengujian untuk memvalidasi fungsi yang sudah dikembangkan menggunakan *Black Box*. Tabel 3.49 dibawah ini adalah skenario pengujian untuk *build* 3 aplikasi dari sisi *General User*.

Tabel 3. *Black Box Testing* untuk *Build* 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| Proses Daftar Akun | | | | |
| B3-T01 | Mengosongkan salah satu kolom yang wajib diisi pada *form* Daftar Akun dan menekan tombol “Daftar” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* | |
| B3-T02 | Menuliskan nama di kolom “Nama Lengkap” melebihi 255 karakter dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* | |
| B3-T03 | Menuliskan angka dibawah 18 pada kolom “Usia” kemudian menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia minimal adalah 18 tahun | *Minor* | |
| B3-T04 | Menuliskan angka diatas 70 pada kolom “Usia” kemudian menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa usia maksimal adalah 70 tahun | *Minor* | |
| B3-T05 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif dan menekan tombol “Daftar” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Major* |
| B3-T06 | Mengetikkan No. Whatsapp melebihi 20 digit karakter dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* | |
| B3-T07 | Mengetikkan No. Whatsapp yang sudah terdaftar di database dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* | |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B3-T08 | Menuliskan nama di kolom “Username” melebihi 255 karakter dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* | |
| B3-T09 | Menuliskan *username* yang sudah ada di *database* dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Major* |
| B3-T10 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |
| B3-T11 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B3-T12 | Menuliskan *password* yang berbeda di kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” dan menekan tombol “Daftar” | Sistem menampilkan kesalahan kepada pengguna untuk memastikan bahwa pengguna mengetikkan *password* yang sama | *Minor* |
| B3-T13 | Menuliskan pertanyaan di kolom “Pertanyaan Tantangan” melebihi 255 karakter dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa pertanyaan yang dimasukkan terlalu panjang | *Minor* |
| B3-T14 | Menuliskan pertanyaan di kolom “Jawaban” melebihi 255 karakter dan menekan tombol “Daftar” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jawaban yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-2)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B3-T15 | Mengisi semua kolom dan sesuai dengan format kemudian menekan tombol “Daftar” | Sistem menyimpan data pengguna baru dan mengarahkan pengguna ke halaman *dashboard* | *Major* |
| Proses *Login* | | | |
| B3-T16 | Mengosongkan semua kolomkemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *input* | *Minor* |
| B3-T17 | Mengisi *username* / no. whatsapptetapi mengosongkan *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *password* | *Minor* |
| B3-T18 | Mengosongkan *username* / no. whatsapp tetapi mengisi *password* kemudian menekan tombol ‘Login’ | Sistem menampilkan peringatan kepada pengguna untuk mengisi semua kolom *username* / no. whatsapp | *Minor* |
| B3-T19 | Mengisi *username* / no.whatsappyang belum terdaftar di *database* kemudian menekan tombol ‘Login’ | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B3-T20 | Pengguna menekan tombol ‘Login’ tapi *Password* salah | Sistem menampilkan pesan kesalahan untuk memeriksa ulang *username* / no. whatsapp yang di *input* | *Minor* |
| B3-T21 | Mengisi *username* / no. whatsappdan *password* yang sesuai kemudian menekan tombol ‘Login’ | Sistem mengarahkan pengguna ke halaman *dashboard* | *Major* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-3)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Mengakses *Dashboard General User* | | | |
| B3-T22 | Tidak ada kursus mendatang | Sistem tidak menampilkan bagian Kursus Berlangsung | *Minor* |
| Proses Mengubah Data Profil | | | |
| B3-T23 | Mengosongkan salah satu kolom yang wajib diisi pada halaman ubah profil dan menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* |
| B3-T24 | Mengunggah foto profil dengan format lain selain .jpg, .webp atau .png dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B3-T25 | Mengunggah foto profil dengan ukuran *file* terlalu besar (>2 MB) dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| B3-T26 | Menuliskan nama di kolom “Nama Lembaga Kursus” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |
| B3-T27 | Menuliskan nama di kolom “Username” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* |
| B3-T28 | Menuliskan *username* yang sudah ada di *database* dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Major* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B3-T29 | Menuliskan deskripsi akun di kolom “Deskripsi” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* | |
| B3-T30 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif dan menekan tombol “Simpan” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Major* | |
| B3-T31 | Mengetikkan No. Whatsapp melebihi 20 digit karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* |
| B3-T32 | Mengetikkan No. Whatsapp yang sudah ada di database dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* |
| B3-T33 | Menuliskan pertanyaan di kolom “Pertanyaan Tantangan” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa pertanyaan yang dimasukkan terlalu panjang | *Minor* |
| B3-T34 | Menuliskan pertanyaan di kolom “Jawaban” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa jawaban yang dimasukkan terlalu panjang | *Minor* |
| B3-T35 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B3-T36 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B3-T37 | Menuliskan *password* yang berbeda di kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” dan menekan tombol “Simpan” | Sistem menampilkan kesalahan kepada pengguna untuk memastikan bahwa pengguna mengetikkan *password* yang sama | *Minor* |
| B3-T38 | Tanpa mengubah apapun, tekan tombol “Simpan” | Sistem tetap menyimpan data walaupun tidak ada perubahan dan mengarahkan ke halaman profil | *Trivial* |
| B3-T39 | Melakukan perubahan dengan benar dan sesuai kemudian menekan tombol “Simpan” | Sistem melakukan perubahan di *database* dan mengarahkan pengguna ke halaman profil | *Major* |
| B3-T40 | Melakukan perubahan kemudian menekan tombol “Batal” | Sistem langsung mengarahkan pengguna ke halaman profil | *Minor* |
| Proses Mengakses Detail Kelas Kursus | | | | |
| B3-T41 | Menekan tombol “Daftar Kelas” tetapi kuota kelas sudah penuh | Sistem memberikan pesan kesalahan kepada pengguna bahwa kelas sudah penuh | *Major* | |
| B3-T42 | Menekan tombol “Daftar Kelas” dan kuota kelas masih tersedia | Sistem mengarahkan pengguna ke formulir pendaftaran kelas kursus | *Major* | |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-6)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Menampilkan Detail *Progress* Kursus Siswa | | | |
| B3-T43 | Pada bagian informasi pertemuan | Sistem memberikan tampilan yang berbeda untuk pertemuan sudah dilaksanakan, pertemuan selanjutnya, dan sisa pertemuan mendatang | *Trivial* |
| B3-T44 | Untuk siswa yang belum mengunggah bukti pembayaran | Sistem menampilkan peringatan bahwa siswa belum mengunggah bukti pembayaran | *Minor* |
| B3-T45 | Untuk siswa yang sudah mengunggah bukti pembayaran tapi belum diverifikasi | Sistem menampilkan peringatan bahwa bukti pembayaran belum diverifikasi | *Minor* |
| Proses Mengunggah Bukti Pembayaran Kelas Kursus | | | |
| B3-T46 | Membuka *accordion* salah satu metode pembayaran | Sistem menampilkan langkah-langkah pembayaran | *Minor* |
| B3-T47 | Mengunggah bukti pembayaran dengan format lain selain .jpg, .webp atau .png dan menekan tombol “Kirim Bukti Pembayaran” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B3-T48 | Mengunggah bukti pembayaran dengan ukuran *file* terlalu besar (>2 MB) dan menekan tombol “Kirim Bukti Pembayaran” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| Proses Menghubungi Siswa atau Instruktur Kursus | | | |
| B3-T49 | Memilih opsi “Hubungi Instruktur” atau “Hubungi Admin” | Sistem mengarahkan pengguna ke halaman mulai percakapan Whatsapp | *Major* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-7)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Memilih Jadwal Kursus | | | |
| B3-T50 | Mengosongkan salah satu atau lebih kolom data atau tidak memilih salah satu opsi kemudian menekan tombol “Simpan Jadwal” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi atau memilih salah satu opsi | *Minor* |
| B3-T51 | Mengubah jadwal kursus dimana jadwal baru akan berlangsung kurang dari 24 jam kemudian menekan tombol “Simpan Jadwal” | Sistem menampilkan peringatan kepada pengguna untuk memastikan jadwal baru tidak berlangsung kurang dari 24 jam dan mengembalikan nilai yang diinputkan sebelumnya | *Major* |
| B3-T52 | Mengubah jadwal kursus yang bertabrakan dengan jadwal kursus lain kemudian menekan tombol “Simpan Jadwal” | Sistem menampilkan peringatan kepada pengguna bahwa sudah ada jadwal yang berlangsung pada jam tersebut dan mengembalikan nilai yang diinputkan sebelumnya | *Major* |
| B3-T53 | Mengubah jadwal kursus dengan benar dan sesuai kemudian menekan tombol “Simpan Jadwal” | Sistem mengarahkan pengguna ke halaman detail *progress* kursus siswa dan menampilkan pesan jadwal kursus berhasil diubah | *Major* |
| B3-T54 | Menekan tombol “Batal” | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus | *Minor* |
| Proses Membaca Panduan Kursus | | | |
| B3-T55 | Pengguna selesai membaca panduan kursus dan menekan tombol “Selesai” | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus dan memberikan *checklist* di Baca Panduan pada daftar capaian pertemuan saat ini | *Trivial* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-8)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| Proses Menyelesaikan *Quiz* | | | |
| B3-T56 | Pengguna menjawab semua pertanyaan | Sistem mengarahkan pengguna kembali ke halaman detail *progress* kursus dan memberikan *checklist* di Selesaikan *Quiz* pada daftar capaian pertemuan saat ini | *Trivial* |
| Proses Mengakses Daftar Riwayat Kursus | | | |
| B3-T57 | Pengguna masuk ke halaman Daftar Riwayat Kursus | Sistem memisahkan antara kursus yang sedang aktif dan kursus lampau | *Trivial* |
| Proses Mencari Kelas Kursus atau Penyedia Kursus | | | |
| B3-T58 | Pengguna masuk di halaman pencarian untuk pertama kalinya | Sistem tidak menampilkan riwayat pencarian | *Trivial* |
| B3-T59 | Pengguna masuk di halaman pencarian dan sudah pernah melakukan pencarian sebelumnya | Sistem menampilkan riwayat pencarian | *Trivial* |
| B3-T60 | Pengguna memilih salah satu riwayat pencarian | Sistem mengeksekusi pencarian menggunakan kata kunci yang dipilih | *Minor* |
| B3-T61 | Pengguna menekan tombol “X” di samping riwayat pencarian | Sistem menghapus riwayat pencarian | *Minor* |
| B3-T62 | Pengguna mengetikkan kata kunci di kolom pencarian kemudian menekan “Enter” tapi tidak menemukan hasil | Sistem menampilkan pesan peringatan bahwa pencarian tidak ditemukan dan kata kunci yang dimasukkan sebelumnya tidak ditampilkan sebagai riwayat pencarian | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-9)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B3-T63 | Pengguna mengetikkan kata kunci di kolom pencarian kemudian menekan “Enter” dan cocok | Sistem mengarahkan pengguna ke halaman hasil pencarian dan menampilkan kata kunci yang digunakan dan kelas kursus atau penyedia kursus hasil pencarian | *Minor* |
| Proses Mendaftarkan Diri sebagai Penyedia Kursus | | | |
| B3-T64 | Mengosongkan salah satu atau lebih kolom data kemudian menekan tombol “Kirim Izin Kursus” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib | *Minor* |
| B3-T65 | Mengunggah izin kursus dengan format lain selain .jpg, .webp atau .png kemudian menekan tombol “Kirim Izin Kursus” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B3-T66 | Mengunggah izin kursus dengan ukuran *file* terlalu besar (>2 MB) kemudian menekan tombol “Kirim Izin Kursus” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| B3-T67 | Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Kirim Izin Kursus” | Sistem menyimpan data izin kursus baru ke *database* dan mengarahkan pengguna ke halaman profil dan tunggu verifikasi izin kursus dari admin sistem | *Major* |
| B3-T68 | Untuk izin kursus yang masih menunggu verifikasi | Sistem menampilkan pesan kepada pengguna untuk menunggu beberapa saat lagi | *Minor* |
| B3-T69 | Untuk izin kursus yang gagal diverifikasi | Sistem menampilkan pesan kepada pengguna untuk mengunggah ulang izin kursus | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-10)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B3-T70 | Untuk izin kursus yang berhasil diverifikasi | Sistem menampilkan pesan kepada pengguna untuk melanjutkan proses | *Minor* |
| B3-T71 | Mengosongkan salah satu kolom yang wajib diisi pada halaman informasi akun dan menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* |
| B3-T72 | Mengunggah foto profil dengan format lain selain .jpg, .webp atau .png dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna memberitahukan format yang didukung hanya .jpg, .webp dan .png | *Minor* |
| B3-T73 | Mengunggah foto profil dengan ukuran *file* terlalu besar (>2 MB) dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *file* yang diunggah harus dibawah 2 MB | *Minor* |
| B3-T74 | Menuliskan nama di kolom “Nama Lembaga Kursus” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |
| B3-T75 | Menuliskan nama di kolom “Username” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* yang dimasukkan terlalu panjang | *Minor* |
| B3-T76 | Menuliskan *username* yang sudah ada di *database* dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *username* sudah digunakan | *Major* |
| B3-T77 | Menuliskan deskripsi akun di kolom “Deskripsi” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa deskripsi yang dimasukkan terlalu panjang | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-11)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B3-T78 | Mengetikkan karakter non-numerik pada kolom No. Whatsapp Aktif dan menekan tombol “Simpan” | Pengguna tidak bisa mengetikkan karakter non-numerik | *Minor* | |
| B3-T79 | Mengetikkan No. Whatsapp melebihi 20 digit karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp terlalu panjang | *Major* |
| B3-T80 | Mengetikkan No. Whatsapp yang sudah ada di database dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa No. Whatsapp sudah terdaftar | *Major* |
| B3-T81 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru” kurang dari 5 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* minimal berisi 5 karakter | *Minor* |
| B3-T82 | Menuliskan *password* pada kolom “Masukkan Password Baru” dan “Ketik Ulang Password Baru”lebih dari 255 digit dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa *password* terlalu panjang | *Minor* |
| B3-T84 | Melakukan perubahan dengan benar dan sesuai kemudian menekan tombol “Simpan” | Sistem melakukan perubahan di *database* dan mengarahkan pengguna ke halaman form tambah metode pembayaran | *Major* |
| B3-T85 | Melakukan perubahan kemudian menekan tombol “Kembali” | Sistem langsung mengarahkan pengguna ke halaman form unggah izin kursus | *Minor* |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-12)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** | |
| B3-T86 | Mengosongkan salah satu kolom yang wajib diisi pada halaman tambah rek. pembayaran dan menekan tombol “Simpan” | Sistem menampilkan peringatan kepada pengguna untuk mengisi kolom *input* yang wajib diisi tersebut | *Minor* | |
| B3-T87 | Menuliskan nama di kolom “Nama Pemilik Akun Pembayaran” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nama yang dimasukkan terlalu panjang | *Minor* |
| B3-T88 | Menuliskan nomor rekening atau alamat pembayaran di kolom “Nomor Rekening Pembayaran” melebihi 255 karakter dan menekan tombol “Simpan” | Sistem menampilkan pesan kesalahan kepada pengguna bahwa nomor rekening atau alamat pembayaran yang dimasukkan terlalu panjang | *Minor* |
| B3-T89 | Melakukan penambahan dengan benar dan sesuai kemudian menekan tombol “Simpan” | Sistem melakukan perubahan di *database* dan mengarahkan pengguna ke halaman *dashboard* Pemilik / Admin | *Major* |
| B3-T90 | Melakukan penambahan kemudian menekan tombol “Kembali” | Sistem langsung mengarahkan pengguna ke halaman form informasi akun | *Minor* |
| Proses Menghapus Akun | | | | |
| B3-T91 | Menekan Menu “Hapus Akun” tapi pengguna tidak mempunyai kursus aktif | Sistem menampilkan pesan kesalahan bahwa pengguna masih memiliki kursus aktif | *Minor* | |
| B3-T92 | Menekan Menu “Hapus Akun” tapi pengguna masih mempunyai kursus aktif | Sistem memunculkan dialog *pop-up* untuk mengkonfirmasi aksi | *Minor* | |
| B3-T93 | Memilih Opsi “Batal” pada dialog “Hapus Akun?” | Sistem menutup dialog *pop-up* | *Major* | |

Tabel 3. 49 *Black Box Testing* untuk *Build* 3 (Lanjutan-13)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Skenario Pengujian** | **Hasil yang diharapkan** | **Tingkat Pengujian** |
| B3-T94 | Memilih Opsi “Ya, Hapus Akun” pada dialog “Hapus Akun?” | Sistem menutup dialog *pop-up* dan menampilkan pesan gagal bahwa pengguna masih memiliki kursus aktif | *Major* |
| Proses *Log Out* | | | |
| B3-T95 | *General User* menekan tombol “Log Out” | Sistem menampilkan dialog *pop-up* untuk mengkonfirmasi aksi pengguna | *Major* |
| B3-T96 | Pemilik/Admin menekan tombol “X” atau ‘Batal’ | Sistem menutup dialog *pop-up* | *Minor* |
| B3-T97 | Pemilik/Admin menekan tombol “Ya” | Sistem mengakhiri sesi pengguna dan mengarahkan pengguna ke halaman awal aplikasi | *Major* |

### Implementation

Untuk menutup proses *increment*, tim pengembang akan menggabungkan *build-build* sebelumnya dengan *build* saat ini. Penggabungan ini juga memastikan bahwa setiap fungsi masih berfungsi sebagaimana mestinya. Setelah proses ini selesai, selanjutnya adalah pengujian menggunakan ISO 25010:2023.

## Pengujian Aplikasi dengan ISO 25010

Untuk menguji apakah fitur yang sudah dibangun untuk setiap *build* sudah sesuai dengan yang dirumuskan pada analisa kebutuhan, tim pengembang selanjutnya akan melakukan pengujian menggunakan ISO 25010 utamanya pada faktor *functional suitability* dan *usability*. Tabel 3.43 berikut adalah daftar pertanyaan untuk pengujian ISO 25010:2023.

Tabel 3. Daftar Pertanyaan Uji Kelayakan Aplikasi dengan ISO 25010

|  |  |
| --- | --- |
| **Sub-Karakteristik** | **Indikator** |
| ***Functional Suitability*** | |
| *Functional Completeness* | Apakah KEMUDI menyediakan semua fungsi yang dibutuhkan oleh Anda untuk mengelola, mendaftar atau melakukan proses kursus mengemudi yang diinginkan? |
| *Functional Correctness* | Apakah KEMUDI memberikan hasil yang benar dan konsisten sesuai dengan harapan Anda? |
| *Functional Appropriateness* | Apakah fungsi-fungsi yang disediakan oleh KEMUDI membantu Anda menyelesaikan mengelola, mendaftar atau melakukan proses kursus mengemudi dengan efisien? |
| ***Interaction Capability*** | |
| *Appropriateness Recognizability* | Apakah Anda setuju bahwa KEMUDI membantu anda dalam mengelola, mendaftar atau melakukan kursus mengemudi? |
| *Learnability* | Apakah Anda merasa bahwa Anda membutuhkan waktu untuk membiasakan diri dengan KEMUDI? |
| *Operability* | Apakah Anda dapat mengoperasikan KEMUDI dengan mudah dan tanpa kesulitan? |
| *User Error Protection* | Apakah KEMUDI membantu Anda mencegah kesalahan dan memberikan bantuan saat terjadi kesalahan? |
| *User Engagement* | Apakah KEMUDI berhasil membuat Anda ingin kembali menggunakannya secara terus menerus? |
| *Inclusivity* | Seberapa mudahkah Anda bernavigasi dan memahami berbagai fitur yang ditawarkan oleh KEMUDI, terlepas dari keahlian teknis Anda? |
| *User Assistance* | Seberapa puaskah Anda dengan ketersediaan dan kualitas bantuan (tutorial, pertanyaan-pertanyaan umum tentang aplikasi, dll.) yang diberikan oleh KEMUDI? |
| *Self-Descriptiveness* | Apakah label-label, petunjuk, instruksi, dan pesan kesalahan yang ditampilkan oleh KEMUDI sudah jelas dan mudah dipahami? |