# 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* & *Logout* sebagai Admin / Pemilik |
| 2 | Dashboard yang berisi jadwal kursus setiap Instruktur |
| 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 | Menghapus / menonaktifkan kelas kursus |
| 7 | Menambah Instruktur kursus |
| 8 | Mengundang Instruktur kursus |

Tabel 3. 1 Kebutuhan Aplikasi (Lanjutan-1)

|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| 9 | Menonaktifkan / menghapus data Instruktur kursus |
| 10 | Menampilkan daftar Siswa aktif |
| 11 | Menampilkan detail *progress* setiap Siswa |
| 12 | Mengajukan atau mengkonfirmasi perubahan jadwal kursus kepada Siswa dan Instruktur |
| 13 | Menampilkan hasil pengisian formulir pendaftaran kursus |
| 14 | Mengakses bukti pembayaran kursus dari Siswa |
| 15 | Menghubungi Siswa dan Instruktur |
| 16 | Mengkonfirmasi status pembayaran kursus dari Siswa |
| 17 | Mengunggah dokumen izin penyelenggaraan kursus mengemudi yang sah |
| 18 | Menutup Lembaga Kursus |
| Sisi Instruktur (Lembaga & Perorangan) | |
| 1 | Fungsi *Login* & *Logout* sebagai Instruktur |
| 2 | Dashboard yang berisi jadwal kursus per hari |
| 3 | Mengubah data yang ada pada profil akun Instruktur |
| 4 | Menampilkan daftar Siswa aktif |
| 5 | Menampilkan detail *progress* Siswa |
| 6 | Mengajukan atau mengkonfirmasi perubahan jadwal kursus kepada Siswa dan Admin / Pemilik |

Tabel 3. 1 Kebutuhan Aplikasi (Lanjutan-2)

|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| 7 | Menghubungi Siswa dan Admin / Pemilik |
| 8 | Mengisi form data instruktur dari hasil undangan Admin / Pemilik |
| 9 | Mengunggah Sertifikat hasil Pelatihan Instruktur Kursus Mengemudi |
| Sisi *General User* | |
| 1 | Fungsi *Login* & *Logout* sebagai *General User* |
| 2 | Dashboard yang berisi rekomendasi kelas kursus, penyedia kursus dan *progress* kursus yang sedang aktif jika ada |
| 3 | Mengubah data yang ada pada profil akun *General User* |
| 4 | Melakukan pengisian formulir pendaftaran kelas kursus |
| 5 | Mengunggah bukti pembayaran kelas kursus |
| 6 | Menghubungi Admin / Pemilik dan Instruktur |
| 7 | Menampilkan detail *progress* kursus yang berlangsung |
| 8 | Mengajukan atau mengkonfirmasi perubahan jadwal kursus kepada Instruktur dan Admin / Pemilik |
| 9 | Mengakses panduan teknik berkendara dan menyelesaikan quiz |
| 10 | Menjadi pemilik kursus dengan mengunggah surat izin penyelenggaraan kursus mengemudi |

Tabel 3. 1 Kebutuhan Aplikasi (Lanjutan-3)

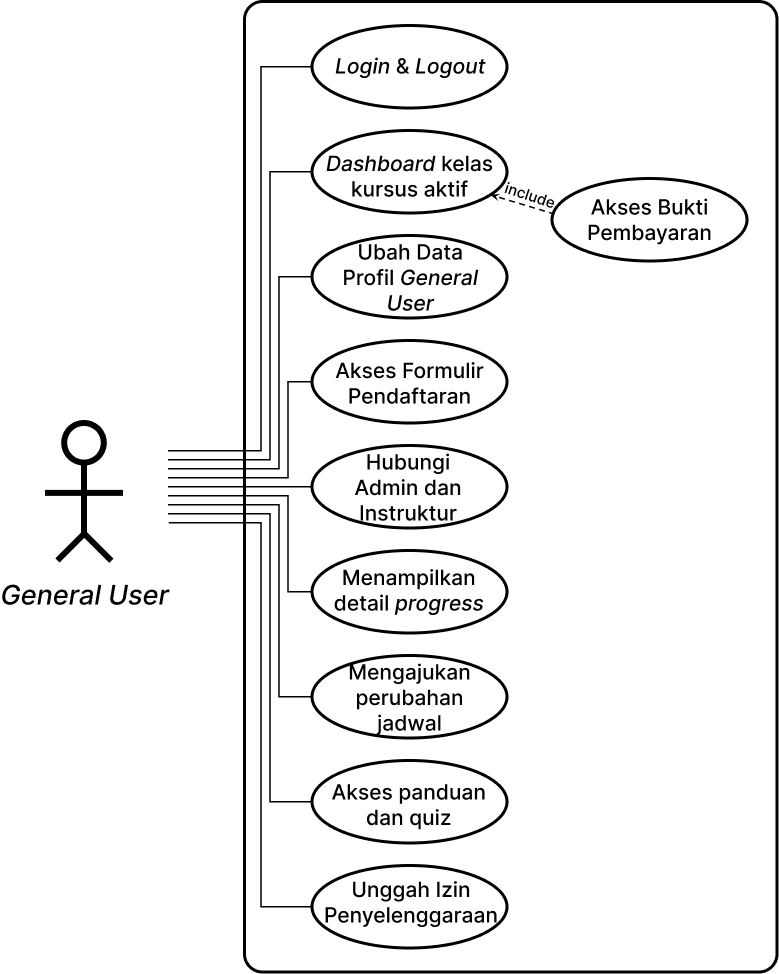
|  |  |
| --- | --- |
| **No.** | **Kebutuhan / Persyaratan** |
| Kebutuhan Teknis | |
| 1 | Menggunakan bahasa pemrograman berbasis web seperti HTML, CSS, Javascript dan PHP lebih spesifik menggunakan *Framework* Laravel |
| 2 | Menggunakan sistem pengelolaan basis data relasional MySQL |
| 3 | Aplikasi dibangun untuk semua jenis *browser* |
| 4 | Spesifikasi perangkat keras yang disarankan adalah RAM diatas 2 GB |

## Design

Selama rangkaian proses perancangan atau desain 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.

### *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 *General User.*

**

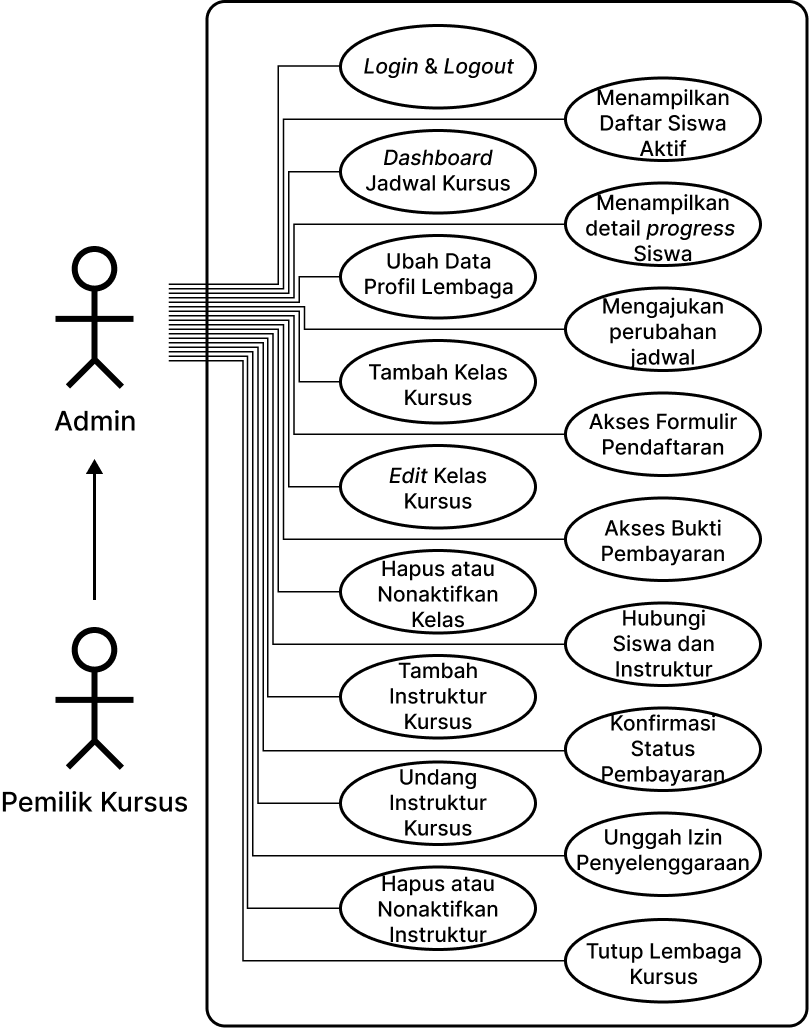
Gambar 3. *Use Case Diagram* untuk *General User*

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. Gambar 3.8 dibawah adalah *use case diagram* untuk Instruktur Kursus.



Gambar 3. *Use Case Diagram* untuk Instruktur

Instruktur Kursus berperan sebagai pihak penengah antara Siswa dan Admin / Pemilik Kursus, untuk beberapa kasus, Pemilik Kursus juga bisa menjadi Instruktur Kursus. *Use case diagram* selanjutnya untuk pengguna Admin / Pemilik Kursus yang akan dijelaskan menggunakan Gambar 3.9 dibawah ini.



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* & *Logout*

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\_untuk\_semua\_user | |
| **Nama *Use Case*** | *Login* | |
| **Aktor** | *General User*, Pemilik / Admin, Instruktur | |
| **Deskripsi** | Skenario untuk menyimpan data dari segala aktivitas yang dilakukan oleh Aktor | |
| **Kondisi Awal** | Aktor berada di halaman *Login* | |
| **Kondisi Akhir** | Aktor berhasil memasukkan *username* dan *password* dengan benar | |
| **Alur Kejadian** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan *form* *Login* |
| 2. Aktor mengisi kolom *username* dan *password* |  |
|  | 3. Sistem mengecek inputan dari Aktor dengan data yang ada pada *Database* |
|  | 4. Sistem berhasil menemukan data yang cocok dengan inputan Aktor |

Tabel 3. 2 *Use Case Scenario* untuk proses *Login* (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 5. Sistem menampilkan *modals* “Login Berhasil” dan mengarahkan Aktor ke halaman selanjutnya |

Proses *logout* dimulai dari pengguna menekan tombol “Logout” di *navbar* atau di bagian *footer* aplikasi. Tabel 3.3 dibawah akan menjelaskan alur proses *logout*.

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

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_logout\_untuk\_semua\_user | |
| **Nama *Use Case*** | *Logout* | |
| **Aktor** | *General User*, Pemilik / Admin, Instruktur | |
| **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” |

Tabel 3.3 *Use Case Scenario* untuk proses *Logout* (Lanjutan-1)

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

#### Proses Akses *Dashboard* untuk Pihak Kursus

Untuk menampilkan *dashboard* pihak kursus (pemilik/admin kursus dan instruktur) harus menyelesaikan alur *Login* terlebih dulu, selanjutnya sistem akan menampilkan jadwal kursus yang dikelompokkan berdasarkan tanggal, dimulai dengan hari ini, esok hari, dan hari selanjutnya. Yang membedakan adalah jika instruktur hanya bisa melihat jadwal untuk dirinya sendiri, sedangkan pemilik/admin dapat melihat seluruh jadwal instruktur yang dibawahinya. Jumlah hari yang ditampilkan dapat diatur untuk menghindari pengguna kewalahan memproses informasi. Tabel dibawah akan menjelaskan respon sistem akan interaksi yang dilakukan oleh pengguna.

Tabel 3. *Use Case Scenario* untuk Menampilkan Jadwal Kursus bagi Pemilik / Admin dan Instruktur

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_tampilkan\_beranda\_pihak\_kursus |
| **Nama *Use Case*** | Menampilkan *dashboard* pihak kursus danjadwal kursus |
| **Aktor** | Pemilik / Admin, Instruktur |

Tabel 3. 4 *Use Case Scenario* untuk Menampilkan Jadwal Kursus bagi Pemilik / Admin dan Instruktur (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Deskripsi** | Skenario untuk menampilkan jadwal 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 3 hari (hari ini, esok hari, dan hari setelahnya) |

#### Proses Akses *Dashboard* untuk *General User*

Sama seperti pihak kursus, untuk menampilkan *dashboard* bagi *General User,* aktor terkait harus menyelesaikan alur *Login* terlebih dulu, selanjutnya sistem akan menampilkan rekomendasi kelas kursus, lembaga atau penyedia kursus mengemudi terdekat, dan *section* untuk menampilkan *progress* kursus yang sedang aktif, jika ada. Tabel 3.5 dibawah adalah alur sistem menampilkan halaman beranda *General User*.

Tabel 3. *Use Case Scenario* untuk Menampilkan Jadwal Kursus bagi *General User*

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_tampilkan\_beranda\_*General*\_*User* |
| **Nama *Use Case*** | Menampilkan *dashboard* untuk *General User* |

Tabel 3. 5 *Use Case Scenario* untuk Menampilkan Jadwal Kursus bagi *General User* (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **Aktor** | *General User* | |
| **Deskripsi** | Skenario untuk menampilkan beranda *General User* | |
| **Kondisi Awal** | Aktor berada di halaman Beranda | |
| **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 penyedia kursus yang terdekat dari lokasi Aktor |

#### Proses Mengubah Data pada Akun Profil

Ketiga aktor bisa mengubah data yang terkait dengan profil mereka, diantaranya adalah *username, password,* tentang pengguna, foto profil, nomor whatsapp, dll. Untuk lebih jelasnya akan kami jelaskan menggunakan tabel 3.6 dibawah ini.

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

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_ubah\_data\_akun | |
| **Nama *Use Case*** | Mengubah data terkait dengan akun Aktor | |
| **Aktor** | *General User*, Pemilik / Admin, Instruktur | |
| **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 memeriksa apakah Aktor sudah *login,* jika belum, sistem akan mengarahkan ke halaman *login* |
|  | 3. Sistem menampilkan *form* untuk mengganti elemen-elemen yang terkait dengan akun Aktor (*username*, *password*, Nama Aktor, Nomor Whatsapp, foto profil Aktor, dll.) |

Tabel 3. 6 *Use Case Scenario* untuk Mengubah Data pada Akun Profil (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
| 4. Aktor mengubah data-data terkait dengan akun yang ingin diubah |  |
| 5. Aktor mengkonfirmasi perubahan data |  |
|  | 6. 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 3.7 dibawah ini.

Tabel 3. *Use Case Scenario* bagi Pemilik / Admin untuk Menambah Kelas Kursus

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_tambah\_kelas\_kursus | |
| **Nama *Use Case*** | Menambah kelas kursus yang ditawarkan pihak kursus | |
| **Aktor** | Pemilik / Admin Kursus | |
| **Deskripsi** | Skenario untuk menambah kelas kursus | |
| **Kondisi Awal** | Aktor berada di halaman kursus | |
| **Kondisi Akhir** | Sistem menyimpan tambahan kelas baru | |
| **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, beberapa diantaranya “Nama Kelas Kursus, Jumlah Pertemuan, Instruktur yang tersedia, Harga Kelas, Deskripsi Kelas, Fitur-fitur dan *Benefit* kelas, dan Kategori dll.” |  |

Tabel 3. 7 *Use Case Scenario* bagi Pemilik / Admin untuk Menambah Kelas Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
|  | 4. Aktor mengkonfirmasi penambahan kelas dan mengaktifkan *Checkbox* “Tawarkan Kelas” |  |
|  | 5. Sistem menambahkan kelas baru ke *database* |
|  |  | 6. Sistem mengarahkan Aktor ke halaman kursus dan menampilkan kelas kursus baru yang ditambahkan. |

#### Proses Mengubah 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 3.8 dibawah ini.

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

|  |  |
| --- | --- |
| **Kode *Use Case*** | UC\_ubah\_kelas\_kursus |
| **Nama *Use Case*** | Mengubah kelas kursus yang ditawarkan pihak kursus |
| **Aktor** | Pemilik / Admin |
| **Deskripsi** | Skenario untuk mengubah detail kelas kursus |
| **Kondisi Awal** | Aktor berada di halaman kursus |

Tabel 3. 8 *Use Case Scenario* untuk Mengubah Detail Kelas Kursus (Lanjutan-1)

|  |  |  |
| --- | --- | --- |
| **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 halaman kursus dan menampilkan kelas kursus sesuai dengan data baru |

#### Proses Menghapus / Menonaktifkan Kelas Kursus

Pemilik / Admin Kursus dapat melakukan penghapusan atau 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 penghapusan dan penonaktifan kelas kursus akan dijelaskan menggunakan tabel 3.9 dibawah ini.

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

|  |  |  |
| --- | --- | --- |
| **Kode *Use Case*** | UC\_hapus\_kelas\_kursus | |
| **Nama *Use Case*** | Menghapus atau Menonaktifkan kelas kursus yang ditawarkan pihak kursus | |
| **Aktor** | Pemilik / Admin | |
| **Deskripsi** | Skenario untuk menghapus atau menonaktifkan kelas kursus | |
| **Kondisi Awal** | Aktor berada di halaman 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. Pemilik / Admin memilih kelas yang akan dihapus atau dinonaktifkan dari daftar kelas, dengan menekan tombol “Hapus” atau menekan tombol *switch* dengan label “Kelas Aktif” yang ada di bagian kanan daftar kelas |  |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 3. Sistem memastikan bahwa jumlah kelas yang ditawarkan lebih dari 1 |
|  | 4. Sistem menampilkan *pop-up* untuk mengkonfirmasi aksi yang dilakukan |
| 5. Aktor menekan tombol “Ya” |  |
|  | 6. Sistem melakukan penghapusan atau penonaktifan kelas sesuai yang dipilih Aktor |

### *Activity Diagram*

### *Class Diagram*

Untuk rancangan desain *database* dari aplikasi, kami menggambarkan struktur *database* dengan *class diagram*. Pada *class diagram* nantinya akan berisi atribut-atribut yang dimiliki masing-masing Aktor dan objek-objek lain, metode-metode yang dapat dieksekusi, dan relasi antar Aktor dan objek pada sistem nantinya. Berikut adalah rancangan *class diagram* untuk aplikasi penyedia jasa kursus mengemudi.



Gambar 3. *Class Diagram* Aplikasi untuk Penyedia Jasa Kursus Mengemudi

Dapat dilihat pada gambar diatas, *database* aplikasi untuk penyedia jasa kursus mengemudi nantinya akan terdapat 8 tabel, 4 tabel utama yang terdiri dari tabel Siswa, Pemilik/Admin, Instruktur, dan Kelas Kursus. 1 tabel untuk menggeneralisasi tabel-tabel pengguna, tabel yang dimaksud adalah tabel Siswa, Instruktur, dan Pemilik/Admin. 2 tabel yang bergantung pada tabel lain, tabel Sertifikat Instruktur bergantung pada tabel Instruktur, tabel Sertifikat Instruktur sendiri dibuat untuk mengakses Sertifikat Instruktur yang sudah di-*upload* ke sistem. Sama halnya dengan tabel Surat Izin Penyelenggaraan, digunakan untuk mengakses Surat Izin Penyelenggaraan Kursus, yang digantungkan dengan tabel Pemilik/Admin, untuk sebuah akun dapat menjadi Pemilik/Admin, akun tersebut harus memiliki Surat Izin Penyelenggaraan yang sah. Dan yang terakhir 1 tabel agregasi untuk mengkategorikan kelas kursus, sebagai contoh, apakah kelas yang ditawarkan untuk mobil dengan transmisi manual atau otomatis? atau apakah kelas yang ditawarkan sifatnya kursus kilat, kursus privat, atau kursus *reguler*? dan sebagainya.

## Development

Pada tahap selanjutnya, tim pengembang akan mengkonversi hasil desain menjadi sistem yang dapat dioperasikan di berbagai jenis perangkat, terutama smartphone semua pengguna. Pengembangan ini akan dilakukan berbasis web, sehingga tim pengembang akan menggunakan teknologi HTML, CSS, Javascript, Framework Laravel, dan MySQL untuk sistem manajemen database. Proses konversi desain menjadi sistem web akan melalui beberapa langkah berikut:

1. **Pemrograman *front-end***: Tim pengembang akan menggunakan HTML, CSS, dan *Javascript* untuk membangun tampilan dan interaksi antarmuka pengguna.
2. **Pemrograman *back-end***: Tim pengembang akan menggunakan *Framework* Laravel untuk membangun logika dan fungsionalitas sistem.
3. **Integrasi *database***: Tim pengembang akan menggunakan MySQL untuk menyimpan dan mengelola data sistem.
4. **Pengujian dan penyempurnaan**: Tim pengembang akan melakukan pengujian menyeluruh untuk memastikan sistem berjalan dengan baik dan sesuai dengan kebutuhan pengguna.

Detail lebih lanjut tentang proses pengembangan sistem web ini, termasuk teknologi yang digunakan, arsitektur sistem, dan hasil pengujian, akan dibahas pada Bab IV.

## Test

Salah satu metode pengujian yang kami lakukan adalah *Black Box Testing*. *Black Box Testing* adalah pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak setelah proses pengembangan selesai (Zidan, M. et al., 2022). Fokus dari pengujian ini adalah menguji kinerja fungsi dari sistem, apakah sudah bekerja sesuai dengan yang diharapkan. Berikut adalah penggalan tabel *black box testing* dari aplikasi kami nantinya.

Tabel 3. Contoh Tabel *Black Box Testing*

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Skenario Tes** | **Hasil yang Diharapkan** | **Sesuai** |
| B01 | *Form* Login/Daftar Akun tidak diisi apa-apa | Sistem tidak dapat mengambil inputan yang diberikan dan menampilkan pesan “Pengisian *form* tidak sesuai” | ✔ |
| B02 | Pada *form* *login*, mengisi kolom *username* = “janedoe@gmail.com” dan *password* = “userlogin123” | Sistem mengambil inputan yang diberikan pengguna dan mengarahkan pengguna ke halaman beranda | ✔ |
| B03 | Pada *form* *login*, mengisi kolom *username* = “janedoe@gmail.com” dan *password* = “user1234” | Sistem mengambil inputan yang diberikan pengguna dan mengembalikan pesan *error* “Password yang dimasukkan tidak sesuai” | ✔ |
| B04 | Pada *form login*, mengisi kolom *username =* “abcabc” dan *password* = “userlogin123” | Sistem mengambil inputan yang diberikan pengguna dan mengembalikan pesan *error* “Data Pengguna tidak ditemukan” | ✔ |

Tabel 3. Contoh Tabel *Black Box Testing* (Lanjutan-1)

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Skenario Tes** | **Hasil yang Diharapkan** | **Sesuai** |
| **Jumlah Pengujian** | | | 4 |
| **Sesuai** | | | 2 |
| **Tidak Sesuai** | | | 2 |
| **Tingkat Keberhasilan** | | | 50% |

Selanjutnya, pada tahun 1996, John Brooke, mengembangkan sebuah skala untuk mengukur atau menilai kegunaan yang dirasakan peserta pengujian, yang dikenal dengan *System Usability Scale* (SUS). (Vlachogianni, P. et al., 2021) beranggapan mengapa SUS sangat populer bahkan sampai saat ini adalah karena SUS adalah alat pengukur psikologi yang handal dan tidak membutuhkan biaya dengan tingkat validasi tinggi dan dapat diandalkan. SUS terdiri dari 10 pernyataan yang secara bergantian menguji kesan positif dan negatif dari perasaan responden atau peserta pengujian terhadap sistem yang diujikan. Dibawah ini adalah 10 pertanyaan yang dirumuskan oleh John Brooke.

Tabel 3. Contoh Tabel Pengujian *System Usability Scale*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Sangat Tidak Setuju** |  |  |  | **Sangat Setuju** |
| **No.** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| 1 | Saya berpikir akan menggunakan sistem ini lagi |  |  |  |  |  |

Tabel 3. Contoh Tabel Pengujian *System Usability Scale* (Lanjutan-1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Sangat Tidak Setuju** |  |  |  | **Sangat Setuju** |
| **No.** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| 2 | Saya merasa sistem ini rumit untuk digunakan |  |  |  |  |  |
| 3 | Saya merasa sistem ini mudah untuk digunakan |  |  |  |  |  |
| 4 | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini |  |  |  |  |  |
| 5 | Saya merasa fitur-fitur sistem ini berjalan dengan semestinya |  |  |  |  |  |
| 6 | Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini |  |  |  |  |  |
| 7 | Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat |  |  |  |  |  |
| 8 | Saya merasa sistem ini membingungkan |  |  |  |  |  |
| 9 | Saya merasa tidak ada hambatan dalam menggunakan sistem ini |  |  |  |  |  |

Tabel 3. Contoh Tabel Pengujian *System Usability Scale* (Lanjutan-2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Sangat Tidak Setuju** |  |  |  | **Sangat Setuju** |
| **No.** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| 10 | Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini |  |  |  |  |  |

## Implementation