# BAB III

# METODE PENELITIAN



## Mengumpulkan dan Menganalisa Kebutuhan

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.

Namun, setelah kami lakukan wawancara dengan tiga penyedia jasa kursus mengemudi, ternyata pemenuhan persyaratan yang dibutuhkan hanya dilakukan sebagian saja. Berikut merupakan hasil analisa pemenuhan persyaratan kami dengan tiga penyedia jasa kursus mengemudi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Dokumen Persyaratan** | **Kursus ABC** | **Sie Bersaudara** | **“Hafiz”** |
| 1 | Akte pendirian perusahaan (Lembaga Kursus) atau KTP (Perorangan) | ✔ | ✔ | ✔ |
| 2 | Struktur organisasi penyelenggara pendidikan mengemudi |  | ✔ |  |
| 3 | Daftar nama personil, riwayat hidup pengelola dan instruktur & Sertifikat Instruktur | ✔ | ✔ | ✔ |
| 4 | Tata tertib penyelenggaraan pendidikan dan pelatihan mengemudi kendaraan bermotor | ✔ | ✔ |  |
| 5 | Rekomendasi penyelenggaraan kursus mengemudi dari Satlantas Polrestabes Surabaya |  |  |  |
| 6 | Kurikulum pendidikan dan pelatihan mengemudi kendaraan bermotor |  |  | ✔ |

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.



Untuk 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.



Proses kursus di kursus mengemudi ABC 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.



BPMN di atas 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.



Untuk 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.



Berbeda dengan dua kursus sebelumnya, kursus mengemudi “Hafiz” 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.



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. Sedangkan dari perspektif desain dan *prototype*, analisa kebutuhan dapat dibagi menjadi 4 aspek: bisnis/pemasaran, fungsional/fitur, teknologi, dan kegunaan.

Kebutuhan bisnis/pemasaran memberikan pemahaman tentang kebutuhan bisnis atau kondisi *marketplace* yang ada. Kebutuhan fungsional memberikan pemahaman akan fitur-fitur yang dibutuhkan untuk mendukung kebutuhan bisnis atau pemasaran. Kebutuhan fungsional juga terkadang merupakan hasil dari hasil riset pengguna dan pengujian kemudahan penggunaan. Selanjutnya spesifikasi fitur ini nantinya akan diterapkan pada *prototype*. Kebutuhan teknologi memberikan pemahaman tentang teknologi yang dibutuhkan untuk mengimplementasikan fitur yang diperlukan. Sedangkan kebutuhan kegunaan memberikan pemahaman tentang *User Experience* dan syarat-syarat kegunaan yang dibutuhkan untuk memudahkan pengguna dalam mengadopsi perangkat lunak yang baru nantinya.

|  |  |  |
| --- | --- | --- |
| **No.** | **Kebutuhan / Persyaratan** | **Aspek** |
| Sisi Admin Kursus (Lembaga) / Pemilik (Perorangan & Lembaga) | | |
| 1 | Fungsi *Login* & *Logout* sebagai admin / pemilik | Fungsional |
| 2 | Dashboard yang berisi jadwal kursus setiap Instruktur per hari | Fungsional |
| 3 | Menambah kelas kursus beserta deskripsi, harga, jumlah pertemuan, benefit, dan kategori kelas kursus | Fungsional |
| 4 | Mengubah nama, deskripsi, harga, jumlah | Fungsional |

|  |  |  |
| --- | --- | --- |
| **No.** | **Kebutuhan / Persyaratan** | **Aspek** |
|  | pertemuan, benefit, dan kategori kelas kursus |  |
| 5 | Menghapus / menonaktifkan kelas kursus | Fungsional |
| 6 | Menampilkan detail *progress* setiap siswa | Fungsional |
| 7 | Mengajukan perubahan jadwal kursus kepada siswa dan instruktur | Kegunaan |
| 8 | Menampilkan hasil pengisian formulir pendaftaran kursus | Fungsional |
| 9 | Menghubungi siswa dan instruktur | Fungsional |
| 10 | Mengkonfirmasi status pembayaran kursus dari siswa | Bisnis/Pemasaran |
| Sisi Instruktur (Lembaga) | | |
| 1 | Fungsi *Login* & *Logout* sebagai Instruktur | Fungsional |
| 2 | Dashboard yang berisi jadwal kursus per hari | Fungsional |
| 3 | Menampilkan daftar siswa aktif | Fungsional |
| 4 | Menampilkan detail *progress* siswa | Fungsional |
| 5 | Mengajukan atau mengkonfirmasi perubahan jadwal kursus kepada siswa | Kegunaan |
| 6 | Menghubungi siswa | Fungsional |

|  |  |  |
| --- | --- | --- |
| **No.** | **Kebutuhan / Persyaratan** | **Aspek** |
| Sisi *General User* | | |
| 1 | Fungsi *Login* & *Logout* sebagai *General User* | Fungsional |
| 2 | Dashboard yang berisi rekomendasi kelas kursus dan *progress* kursus yang sedang aktif | Fungsional |
| 3 | Mendaftarkan diri ke kursus mengemudi | Fungsional |
| 4 | Menghubungi admin / pemilik kursus dan instruktur | Fungsional |
| 5 | Menampilkan detail *progress* kursus yang berlangsung | Fungsional |
| 6 | Mengajukan perubahan jadwal kursus kepada instruktur | Kegunaan |
| 7 | Menampilkan hasil pengisian formulir pendaftaran kursus | Fungsional |
| 8 | Menjadi pemilik kursus dengan mengirimkan surat izin penyelenggaraan kursus mengemudi | Fungsional |
| Kebutuhan Teknis | | |
| 1 | Sistem mampu dijalankan melalui Smartphone semua pihak | Teknologi |
| 2 | Sistem mudah diakses di web/internet | Teknologi |

## Desain Singkat

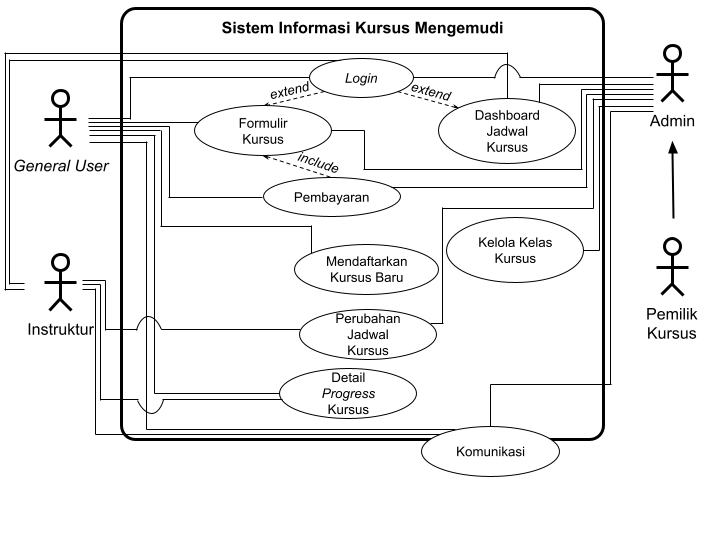
Dari analisa kebutuhan diatas, kami mulai memahami kebutuhan dan kendala-kendala yang dihadapi oleh pihak penyedia jasa kursus mengemudi. Selanjutnya, pada tahap ini, kami mulai dengan membuat rangkaian alur pengguna atau sering dikenal dengan *user flow*, alur ini nantinya akan menjadi basis bagi kami untuk membuat sketsa awal aplikasi.

## Membangun *Prototype*

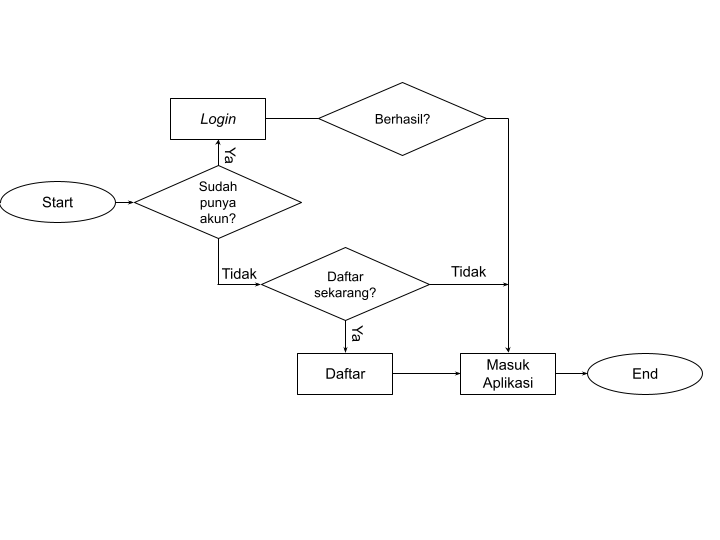
## Pengujian Awal dengan Pengguna

## Menetapkan Desain Akhir

Setelah mengetahui analisa kebutuhan diatas, langkah selanjutnya adalah menerjemahkannya ke bentuk *Use Case Diagram*. Hal ini ditujukan untuk memberikan alur yang lebih jelas tentang interaksi-interaksi yang dilakukan user dengan sistem untuk masing-masing proses yang ada pada sistem, serta sekali lagi memeriksa apakah semua fungsi yang disebutkan diatas sudah terakomodir semuanya. Dari proses-proses yang sudah kami jelaskan diatas, langkah selanjutnya adalah merancang alur sistem baru yang dapat mengakomodasi perbedaan-perbedaan dari satu lembaga kursus dengan lembaga kursus lainnya. Untuk memecah kompleksitas dan menghindari kesalahpahaman, kami akan menjelaskan setiap *use case* dengan *Flowchart* dan *Use Case Scenario.*



### Login



Untuk alur *Login*, sistem mulai dengan menanyakan apakah pengguna sudah punya akun? Jika ya, pengguna bisa langsung melakukan *Login*, jika tidak sistem mempunyai 2 opsi lanjutan, apakah pengguna ingin daftar akun sekarang atau pengguna bisa melakukan aksi ini nanti ketika sistem membutuhkan penyimpanan data untuk akun bersangkutan.

|  |  |  |
| --- | --- | --- |
| **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** | Sistem baru selesai memuat tampilan awal | |
| **Kondisi Akhir** | Aktor berada di halaman beranda/*dashboard* | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
|  | 1. Sistem menampilkan tiga tombol (*Login*, Daftar, dan |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | Nanti) |
| 2. Aktor memilih tombol *Login* |  |
|  | 3. Sistem menampilkan *form* *Login* |
| 4. Aktor memasukkan data akun yang tersimpan di database |  |
|  | 5. Sistem memverifikasi data yang diinputkan Aktor |
|  | 6. Sistem mengarahkan Aktor |
| **Alur Kejadian Alternatif** | 2a. Aktor memilih tombol “Daftar”  2b. Aktor memilih tombol “Nanti” |  |
|  | 3a. Sistem menampilkan *form* daftar akun  3b. Sistem menganggap Aktor sebagai *General User* |
| 4a. Aktor memasukkan data akun baru  4c. Aktor memasukkan data yang tidak ada di database | 4b. Sistem mengalihkan Aktor ke beranda/*dashboard* |
|  | **Aktor** | **Sistem** |
|  | 5a. Sistem menyimpan data baru yang diinputkan Aktor  5c. Sistem tidak menemukan data yang diinputkan Aktor |
|  | 6a. Sistem mengalihkan Aktor ke beranda/*dashboard*  6c. Sistem mengalihkan Aktor ke tampilan awal dan mengembalikan pesan *error* ke Aktor |

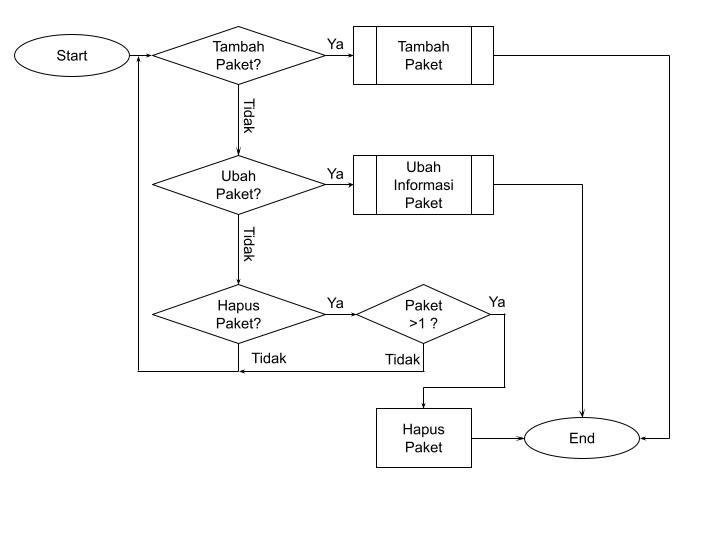
### Menampilkan *Dashboard* Jadwal Kursus

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.

|  |  |  |
| --- | --- | --- |
| **Kode Use Case** | UC\_tampilkan\_jadwal\_kursus | |
| **Nama Use Case** | Menampilkan *dashboard* jadwal kursus | |
| **Aktor** | Pemilik / Admin, Instruktur | |
| **Deskripsi** | Skenario untuk menampilkan jadwal kursus per hari | |
| **Kondisi Awal** | Sistem selesai memverifikasi data akun yang diinputkan Aktor pada proses login | |
| **Kondisi Akhir** | Sistem menampilkan semua jadwal sesuai batas hari yang ditentukan Aktor | |
| **Alur Kejadian Normal** | **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) |
|  | 3. Sistem memberikan opsi lain untuk menampilkan jadwal dalam 15 hari dan 30 hari |
| **Alur Kejadian Alternatif** | **Aktor** | **Sistem** |
| 4a. Aktor memilih opsi untuk tampilkan jadwal dalam 15 hari |  |
|  | 5a. Sistem menampilkan jadwal dimulai dengan hari ini sampai 14 hari kedepan |
| 4b. Aktor memilih opsi untuk tampilkan jadwal dalam 30 |  |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
| hari |  |
|  | 5b. Sistem menampilkan jadwal dimulai dengan hari ini sampai 29 hari kedepan |

### Pengelolaan kelas Kursus



Pemilik/Admin dari suatu kursus dapat melakukan penambahan atau menonaktifkan/hapus kelas kursus yang ada di halaman kursus mereka selama masih ada 1 kelas kursus tersisa. Selain itu, pemilik/admin ini dapat mengubah informasi-informasi yang berkaitan dengan kelas kursus, seperti deskripsi kursus, lama kursus, kategori kursus, benefit, harga, dan lain-lain.

|  |  |  |
| --- | --- | --- |
| **Kode Use Case** | UC\_pengelolaan\_kelas\_kursus | |
| **Nama Use Case** | Kelola kelas Kursus | |
| **Aktor** | Pemilik / Admin, Instruktur | |
| **Deskripsi** | Skenario untuk menambah, mengubah, dan menghapus/menonaktifkan kelas kursus yang ditawarkan | |
| **Kondisi Awal** | Aktor berada di halaman kursus | |
| **Kondisi Akhir** | Aktor berhasil menambah kelas kursus baru | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol “Tambah kelas” pada halaman kursus |  |
|  | 2. Sistem mengalihkan Aktor ke *form* tambah kelas |
| 3. Aktor mengisi semua *form* dengan lengkap | 3. Sistem memverifikasi bahwa semua *form* terisi |
| 4. Aktor menekan tombol “Tawarkan kelas” |  |
|  | 5. Sistem menyimpan *entry* yang dilakukan Aktor |
|  | 6. Sistem mengarahkan Aktor ke halaman kursus dan menampilkan kelas yang baru ditambahkan pada |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | halaman kursus |
| **Alur Kejadian Alternatif** |  | 1a. Sistem menampilkan tombol “Ubah kelas” di masing-masing kelas kursus  1b. Sistem menampilkan tombol “Nonaktfikan kelas” jika kondisi terpenuhi |
|  | 2a. Aktor menekan tombol “Ubah kelas” di kelas Kursus yang dipilih  2b. Aktor menekan tombol “Nonaktifkan kelas” di kelas Kursus yang dipilih |  |
|  | 3c. Aktor belum mengisi semua *form* dengan lengkap | 3a. Sistem mengalihkan Aktor ke *form* ubah kelas  3b. Sistem memeriksa apakah kelas kursus yang ditawarkan > 1  3c. Sistem membuat tombol “Tawarkan kelas” tidak bisa diklik |
|  | 4a. Aktor mengubah semua informasi yang ingin diubah  4c. Aktor menekan tombol “Simpan Draft” | 4b-0, jika kelas kursus hanya 1, sistem akan mengembalikan pesan error dan menyarankan apakah Aktor lebih memilih untuk |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | menutup kursus  4b-1, jika kelas kursus > 1, sistem akan menonaktifkan kelas yang dipilih |
| 5a. Aktor menekan tombol “Ubah kelas” | 5c. Sistem menyimpan *entry* yang dilakukan Aktor |
|  | 6a. Sistem menyimpan *entry* yang dilakukan Aktor  6c. Sistem mengalihkan Aktor ke halaman kursus |
|  | 7a. Sistem mengalihkan Aktor ke halaman kursus |

### Menampilkan Detail *Progress* Kursus

Pihak kursus dan siswa dengan kursus aktif dapat melihat detail dari kursus yang diikuti. Halaman detail *progress* ini guna menunjukkan sedang berada di pertemuan ke berapakah siswa saat ini, didalamnya siswa juga dapat mempelajari teori mengemudi lebih lanjut dengan memilih ‘Baca Panduan’ dan mengerjakan ‘Quiz’ yang disediakan oleh Sistem.

|  |  |  |
| --- | --- | --- |
| **Kode Use Case** | UC\_detail\_progress\_kursus\_instruktur | |
| **Nama Use Case** | Menampilkan detail *progress* kursus | |
| **Aktor** | Pemilik / Admin, Instruktur | |
| **Deskripsi** | Skenario untuk menampilkan detail kursus aktif | |
| **Kondisi Awal** | Aktor berada di *dashboard* | |
| **Kondisi Akhir** | Aktor memperoleh informasi detail kursus dari sistem | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor memilih salah satu siswa dari daftar jadwal yang ditampilkan di *dashboard* |  |
|  | 2. Sistem menampilkan data kursus terkait dengan siswa yang dipilih Aktor |
|  | 3. Sistem menampilkan daftar capaian yang sudah dilakukan oleh siswa |
| **Alur Kejadian Alternatif** | - | - |

|  |  |
| --- | --- |
| **Kode Use Case** | UC\_detail\_progress\_kursus\_siswa |
| **Nama Use Case** | Menampilkan detail *progress* kursus |
| **Aktor** | *General User* |
| **Deskripsi** | Skenario untuk menampilkan detail kursus aktif |
| **Kondisi Awal** | Aktor berada di *dashboard* |
| **Kondisi Akhir** | Aktor bisa memilih untuk perubahan jadwal, membaca panduan teori mengemudi, menyelesaikan Quiz, dan proses-proses lain |

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor menekan ‘Progress Kursus’ pada bagian atas *dashboard* |  |
|  | 2. Sistem menampilkan informasi kursus yang dijalani siswa saat ini |
|  | 3. Sistem menampilkan capaian-capaian yang sudah dilakukan oleh siswa |
| **Alur Kejadian Alternatif** | 1. Aktor berpindah ke halaman profil |  |
| 2. Aktor menekan menu ‘Kursus Saya’ |  |
|  | 3. Sistem menampilkan informasi kursus yang dijalani siswa saat ini |
|  | 4. Sistem menampilkan capaian-capaian yang sudah dilakukan oleh siswa |

### Komunikasi

Baik pihak kursus dan siswa dapat menghubungi satu sama lain dengan aplikasi Whatsapp diluar sistem, sistem sebatas menempelkan link pada tombol yang disediakan pada halaman ‘Detail Kursus’ yang selanjutnya bisa diklik oleh Aktor yang kemudian akan diarahkan Whatsapp untuk mengkonfirmasi apakah Aktor benar ingin melakukan komunikasi dengan Nomor Telepon bersangkutan.

|  |  |  |
| --- | --- | --- |
| **Kode Use Case** | UC\_komunikasi | |
| **Nama Use Case** | Proses Komunikasi antar Aktor | |
| **Aktor** | *General User*, Instruktur, Pemilik/Admin | |
| **Deskripsi** | Skenario untuk berkomunikasi antar Aktor | |
| **Kondisi Awal** | Aktor berada di tampilan ‘Detail Kursus’ | |
| **Kondisi Akhir** | Aktor berada di tampilan percakapan Whatsapp dengan Aktor tujuan | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol ‘Hubungi Siswa/Instruktur/Admin’ |  |
|  | 2a. Untuk nomor Whatsapp Siswa, sistem mengambil nomor dari *Form* pendaftaran kursus  2b. Untuk nomor Whatsapp Instruktur, sistem mengambil nomor dari *Form* undangan mengajar yang dikirimkan Pemilik/Admin  2c. Untuk nomor Whatsapp Pemilik/Admin didapatkan dari *Form* pendaftaran kursus |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | baru yang dikirimkan Aktor |
|  | 3. Sistem mengarahkan Aktor ke halaman konfirmasi percakapan Whatsapp |
| 4. Aktor menekan tombol “Continue to Chat” pada halaman konfirmasi |  |
| **Alur Kejadian Alternatif** | - | - |

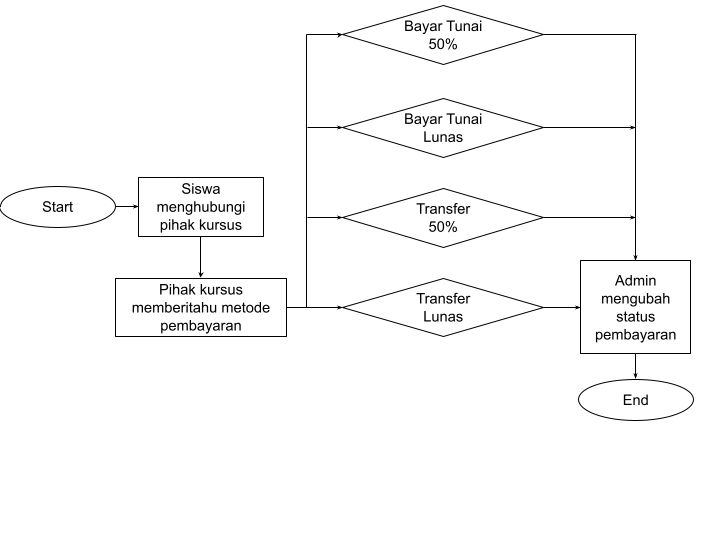
### Siswa mendaftar kelas kursus

Untuk alur mendaftar kursus, Aktor dengan peran *General User* memilih kelas kursus yang dimaksud terlebih dahulu, selanjutnya sistem akan melakukan pengecekkan apakah pengguna sudah melakukan proses *login* sebelumnya. Proses *login* diperlukan agar sistem mengetahui data yang akan di-*entry* oleh pengguna ini tersimpan bersama dengan data akun pengguna.

|  |  |
| --- | --- |
| **Kode Use Case** | UC\_daftar\_kelas\_kursus |
| **Nama Use Case** | Mendaftar kelas kursus |
| **Aktor** | *General User* & Pemilik/Admin |
| **Deskripsi** | Skenario untuk mendaftarkan diri mengikuti kursus mengemudi |
| **Kondisi Awal** | Calon siswa sudah memilih kelas kursus yang diinginkan |

|  |  |  |
| --- | --- | --- |
| **Kondisi Akhir** | Calon siswa menyelesaikan pengisian formulir pendaftaran | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Calon Siswa menekan tombol “Daftar Kelas” |  |
|  | 2. Sistem mengalihkan Calon Siswa ke formulir pendaftaran kursus |
| 3. Calon Siswa mengisi formulir pendaftaran dengan lengkap |  |
|  | 4. Sistem menyimpan *entry* yang dilakukan Calon Siswa |
|  | 5. Sistem mengirimkan pemberitahuan ke Pemilik/Admin |
| 6. Pemilik/Admin memverifikasi formulir yang diisi calon siswa |  |
| **Alur Kejadian Alternatif** | 3. Calon Siswa tidak jadi meneruskan pendaftaran |  |
|  | 4. Sistem mengalihkan Calon Siswa ke beranda/*dashboard* |

### Pembayaran

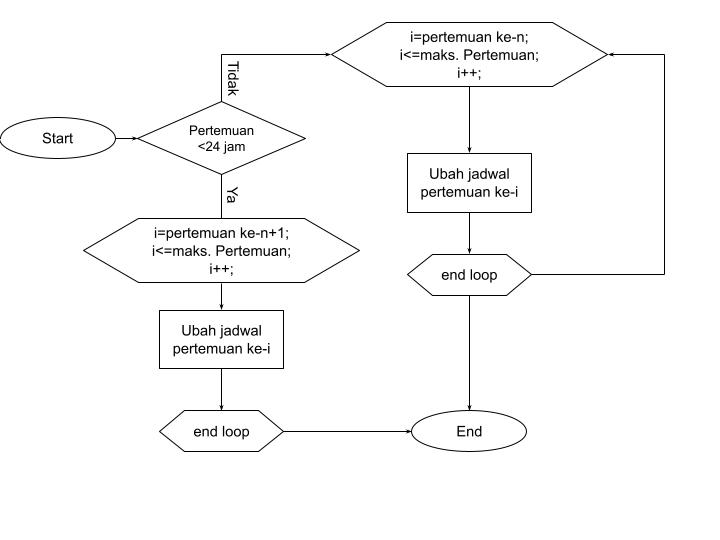


Pada sistem yang kami kembangkan proses pembayaran terjadi di luar sistem, sistem hanya memberitahukan kedua pihak status pembayaran saat ini.

|  |  |  |
| --- | --- | --- |
| **Kode Use Case** | UC\_status\_pembayaran | |
| **Nama Use Case** | Status Pembayaran | |
| **Aktor** | *General User* & Pemilik/Admin | |
| **Deskripsi** | Skenario pemberitahuan status pembayaran | |
| **Kondisi Awal** | Calon siswa baru saja menyelesaikan *form* pendaftaran | |
| **Kondisi Akhir** | Calon siswa mendapatkan informasi status pembayaran | |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
|  | 1. Sistem mengarahkan Calon siswa ke halaman detail kursus |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
| 2. Siswa menghubungi pihak kursus menanyakan tentang proses pembayaran |  |
| 3. Siswa melakukan proses pembayaran |  |
| 4. Pemilik/Admin memverifikasi pembayaran |  |
| 5. Pemilik/Admin mengubah status pembayaran |  |
|  | 6. Sistem menampilkan status pembayaran yang dipilih Admin |
| **Alur Kejadian Alternatif** | - | - |

### Perubahan Jadwal



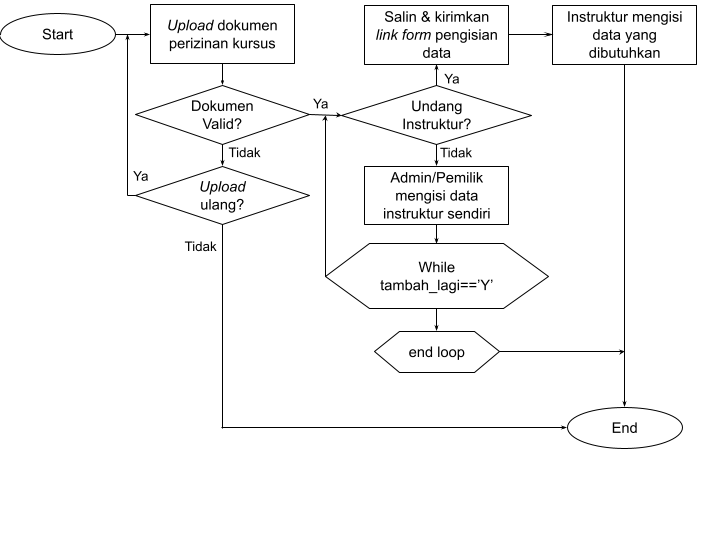
Jika siswa ingin mengubah jadwal, sistem masih bisa melanjutkan proses apabila pertemuan selanjutnya lebih dari 24 jam, jadwal yang diubah nantinya akan mengubah pertemuan selanjutnya sampai pertemuan akhir. Sebagai contoh, siswa memilih kursus dengan 5 pertemuan, pada pertemuan ke-3 siswa tidak dapat hadir sesuai dengan jadwal, jika siswa memilih untuk melakukan perubahan jadwal, maka siswa dapat mengubah jadwal kursus dari pertemuan ke-3 sampai pertemuan ke-5.

|  |  |
| --- | --- |
| **Kode Use Case** | UC\_perubahan\_jadwal |
| **Nama Use Case** | Pengajuan perubahan jadwal |
| **Aktor** | *General User* & Pemilik/Admin |
| **Deskripsi** | Skenario pengajuan perubahan jadwal |
| **Kondisi Awal** | Siswa berada di halaman detail *progress* kursus |
| **Kondisi Akhir** | Siswa berhasil mengubah rangkaian jadwal kursus |

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Aktor menekan tombol ‘Jadwal Kursus’ |  |
|  | 2. Sistem mengalihkan Aktor ke halaman jadwal kursus siswa terkait |
| 3. Aktor menekan tombol ‘Ubah Jadwal’ |  |
|  | 4. Sistem memeriksa apakah pertemuan selanjutnya < 24 jam |
|  | 5a. Jika pertemuan selanjutnya < 24 jam, jadwal yang dapat diubah dimulai dari pertemuan setelah pertemuan mendatang  5b. Jika pertemuan selanjutnya > 24 jam, perubahan jadwal dapat dimulai dari pertemuan mendatang |
| 6. Aktor menyesuaikan jadwal yang tersedia |  |
| 7. Aktor menekan tombol ‘Ajukan Perubahan Jadwal’ |  |
|  | 8. Sistem mengirim |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | pemberitahuan ke Aktor lain untuk dikonfirmasi pengajuan jadwal yang dikirim |
| 9. Aktor lainnya mengkonfirmasi perubahan jadwal |  |
|  | 10. Sistem menyimpan perubahan jadwal |
| **Alur Kejadian Alternatif** | 6. Aktor membatalkan perubahan jadwal |  |
|  | 7. Sistem mengarahkan Aktor ke halaman detail *progress* kursus |
| 9. Aktor lainnya menolak perubahan jadwal |  |
|  | 10. Sistem mengirimkan pemberitahuan penolakan pengajuan jadwal ke Aktor |

### Mendaftarkan Kursus Baru



Seperti proses-proses sebelumnya, karena proses ini memerlukan sistem untuk menyimpan data yang di*-entry* oleh pengguna, sebelum melakukan proses upload dokumen perizinan kursus, sistem akan memeriksa apakah pengguna sudah *login*? Jika sudah, proses akan berlanjut, jika belum, sistem akan mengarahkan pengguna ke halaman *login*.

|  |  |
| --- | --- |
| **Kode Use Case** | UC\_pendaftaran\_kursus\_baru |
| **Nama Use Case** | Pendaftaran Kursus Baru |
| **Aktor** | Instruktur, *General User* & *System* Admin |
| **Deskripsi** | Skenario untuk mendaftarkan jasa kursus mengemudi baru |
| **Kondisi Awal** | Calon Pemilik berada di halaman beranda/*dashboard* |
| **Kondisi Akhir** | Calon Pemilik berada pada halaman beranda/*dashboard* Pemilik/Admin |

|  |  |  |
| --- | --- | --- |
| **Alur Kejadian Normal** | **Aktor** | **Sistem** |
| 1. Calon Pemilik berpindah ke halaman profil |  |
|  | 2. Sistem menampilkan data dan tombol-tombol yang nantinya ditampilkan di halaman profil |
| 3. Calon Pemilik menekan menu ‘Ingin jadi Instruktur?’ |  |
|  | 4. Sistem mengalihkan Aktor ke halaman *upload* dokumen perizinan |
| 5. Calon Pemilik mengupload dokumen perizinan penyelenggaraan kursus |  |
|  | 6. *System* Admin memverifikasi keaslian dokumen yang diupload oleh Calon Pemilik |
|  | 7. Sistem mengirim pesan berhasil ke Calon Pemilik kemudian mengarahkan Calon Pemilik untuk mengisi data Instruktur |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | 8. Sistem memberikan opsi kepada pengguna untuk mengisi data Instruktur sendiri atau diisi oleh Calon Instruktur bersangkutan |
| 9a. Calon Pemilik memilih mengisi data Instruktur sendiri  9b. Calon Pemilik memilih agar diisi oleh Calon Instruktur bersangkutan |  |
| 10a. Calon Pemilik mengisi semua data Instruktur dengan lengkap  10b. Calon Pemilik mengirimkan link *form* penambahan Instruktur ke Calon Instruktur bersangkutan |  |
| 11b. Calon Instruktur menerima *link* undangan mengajar | 11a. Sistem menyimpan *entry* data dari Calon Pemilik, kemudian mengarahkan Calon Pemilik ke tampilan beranda/*dashboard* |
|  | 12b. Sistem menampilkan |

|  |  |  |
| --- | --- | --- |
|  | **Aktor** | **Sistem** |
|  | halaman pendaftaran akun untuk diisi Calon Instruktur |
| 13b. Calon Instruktur mengisi *form* penambahan Instruktur |  |
|  | 14b. Sistem menyimpan *entry* yang dilakukan Instruktur |
| **Alur Kejadian Alternatif** |  | 6. *System* admin menolak keaslian dokumen atau dokumen yang di-*upload* dianggap kurang jelas |
|  | 7. Sistem mengirim pesan gagal ke Calon Pemilik kemudian mengarahkan Calon Pemilik untuk melakukan *upload* ulang |
| 10a. Calon Pemilik tidak mengisi *form* dengan lengkap atau membatalkan pengisian |  |
| 13b. Instruktur tidak mengisi *form* dengan lengkap atau membatalkan pengisian | 11a. Sistem mengalihkan Calon Pemilik ke halaman profil |
|  | 14b. Sistem mengalihkan Calon Instruktur ke halaman |
|  | **Aktor** | **Sistem** |
|  | profil |