

Modul Praktikum

Dasar Rekayasa

Perangkat Lunak

Perencanaan - *Planning*

Planning merupakan tahap awal sebelum diputuskan apakah sistem yang ingin di bangun memang **layak atau tidak** untuk dikembangkan. Pada fase tersebut, **user atau product owner** membawa permintaan kebutuhan sistem (**system request atau change request**) kepada **system analyst**. **System analyst** membuat **analisis kelayakan (feasibility analysis)** berdasarkan system request atau change request (Wahono, 2020). Feasibility analysis terdiri dari technical feasibility, economic feasibility dan organization feasibility. Technical feasibility merupakan analisis risiko teknis yang mengidentifikasi kefamiliaran dengan sistem, kefamiliaran dengan teknologi, ukuran proyek dan kompatibilitas sistem. Economic feasibility merupakan analisis yang mengidentifikasi risiko terkait biaya proyek. Organization feasibility merupakan analisis yang mengidentifikasi kesesuaian sistem yang akan dikembangkan pada organisasi (Dennis. Alan, 2016). Output dari fase planning adalah software proposal.

1. TIM PBL

Isilah detail proyek PBL anda melalui tautan berikut :

<https://docs.google.com/spreadsheets/d/1yPkixHN3B8oF9ON5v0UBO44dBzCJ0DLhXp7v7l1jUNo/edit?usp=sharing>

2. PERENCANAAN PROYEK PBL

Buatlah sebuah Software Proposal yang berisi informasi sebagai berikut :

A. Jelaskan permasalahan yang melatar belakangi dikerjakannya proyek PBL Anda (*Project Overview*)?

Dalam era digital saat ini, jual beli mobil masih menghadapi berbagai tantangan, terutama bagi calon pembeli yang kesulitan menemukan informasi kendaraan yang akurat dan terpercaya. Banyak transaksi masih dilakukan secara konvensional, yang sering kali tidak efisien dan memakan waktu. Selain itu, minimnya platform khusus untuk jual beli mobil yang aman dan transparan membuat banyak pembeli dan penjual ragu dalam bertransaksi.

Untuk mengatasi permasalahan ini, proyek PBL semester 2 kami akan mengembangkan aplikasi jual beli mobil berbasis web yang dapat mempertemukan penjual dan pembeli dalam satu platform terintegrasi. Aplikasi ini akan menyediakan fitur pencarian kendaraan berdasarkan kategori, filter harga dan spesifikasi, serta sistem verifikasi penjual untuk meningkatkan keamanan transaksi. Dengan adanya platform ini, diharapkan jual beli mobil menjadi lebih praktis, cepat, dan aman bagi semua pengguna.

B. Jelaskan deskripsi sistem dari aplikasi yang akan Anda buat? Dan *scope of work* dari proyek yang akan dikerjakan.

- Deskripsi Sistem

Aplikasi jual beli mobil berbasis web ini dirancang untuk mempermudah transaksi antara penjual dan pembeli kendaraan. Pengguna dapat membuat akun sebagai penjual atau pembeli, lalu mengakses fitur utama aplikasi. Penjual dapat mengunggah informasi mobil yang akan dijual, termasuk foto, harga, spesifikasi, dan kontak. Sementara itu, pembeli dapat mencari mobil berdasarkan kategori, harga, atau merek tertentu serta melihat ulasan dari pengguna lain sebelum memutuskan untuk membeli.

Fitur Utama Aplikasi Jual Beli Mobil Berbasis Web

1. Fitur untuk Penjual

- **Login & Logout** → Penjual dapat masuk dan keluar dari akun mereka dengan sistem keamanan yang terjamin.
- **Manajemen Produk** → Penjual dapat menambahkan, mengedit, dan menghapus daftar mobil yang dijual.
- **Perekapan History Pembelian** → Penjual dapat melihat riwayat transaksi yang telah dilakukan oleh pembeli.

2. Fitur untuk Pembeli

- **Registrasi & Login** → Pembeli dapat membuat akun dan masuk ke aplikasi.
- **Pengelolaan Profil** → Pembeli dapat mengedit informasi akun mereka.
- **Pencarian & Filter Produk** → Pembeli dapat mencari mobil berdasarkan kategori, harga, atau merek tertentu.
- **Wishlist/Daftar Keinginan** → Memungkinkan pembeli untuk menambahkan mobil sebelum mengajukan pembelian.
- **Pengajuan Pembelian Mobil** → Pembeli dapat melakukan pembelian secara kredit atau transfer bank secara lunas.

-
- **Verifikasi & Konfirmasi** → Mengisi data diri dan dokumen yang diperlukan (KTP, NPWP, dll).
 - **Tracking Pembelian Mobil** → Pembeli dapat melihat nomor transaksi atau status pesanan setelah melakukan pengajuan pembelian mobil. Pihak dealer atau admin website akan menghubungi pembeli.
 - **Riwayat Pembelian** → Pembeli dapat melihat invoice.
 - **Logout** → Pembeli dapat keluar dari akun mereka dengan aman.

Scope of Work

Cakupan pekerjaan proyek ini meliputi:

- **Pengembangan Frontend:**
 - Pembuatan antarmuka menggunakan HTML, CSS, dan JavaScript.
 - Desain responsif agar dapat diakses melalui perangkat mobile dan desktop.
- **Pengembangan Backend:**
 - Implementasi sistem registrasi dan login pengguna.
 - Database untuk menyimpan data pengguna, iklan mobil, dan transaksi.
 - API untuk menangani interaksi antara frontend dan backend.
- **Fitur Keamanan:**
 - Sistem enkripsi data pengguna.
 - Verifikasi akun untuk mengurangi risiko penipuan.
- **Pengujian dan Debugging:**
 - Pengujian fungsionalitas sistem untuk memastikan semua fitur berjalan dengan baik.
 - Debugging untuk mengatasi bug dan meningkatkan performa aplikasi.

Batasan proyek:

- Sistem hanya berfokus pada jual beli mobil, tidak termasuk servis atau sparepart.
- Penggunaan aplikasi terbatas, hanya bisa pada pengguna yang telah terverifikasi.

C. Pilih *Methodology* yang akan digunakan untuk menyelesaikan projectnya

Metodologi yang digunakan untuk proyek ini adalah **Software Development Life Cycle (SDLC)**. **SDLC** memberikan pendekatan sistematis dalam pengembangan perangkat lunak dengan tahapan yang jelas, yaitu :

1. **Planning** : Menentukan tujuan, kebutuhan sistem, dan analisis kebutuhan pengguna.
2. **Analysis** : Mengumpulkan kebutuhan sistem dari pengguna dan menganalisis spesifikasi teknis yang diperlukan.
3. **Design** : Mendesain arsitektur sistem, tampilan UI/UX, dan struktur database.
4. **Implementation (Coding)** : Mengembangkan sistem berdasarkan desain yang telah dibuat menggunakan teknologi yang dipilih.
5. **Testing dan Debugging** : Menguji sistem untuk memastikan tidak ada bug atau kesalahan dalam implementasi.
6. **Maintenance**: Memelihara dan memperbaiki sistem jika ditemukan masalah setelah implementasi.

D. Buatlah *Timeline* yang jelas yang berisi *deadline* dan *project milestone*

Fase/Proses	Uraian Pekerjaan	Estimasi Waktu
Planning (Perencanaan)	<ul style="list-style-type: none">● Mahasiswa melakukan pertemuan dengan manpro terkait perencanaan yang akan dilakukan dalam pembuatan aplikasi berbasis web● Mencari referensi	2 minggu

	<p>dan mencari data yang dibutuhkan.</p> <ul style="list-style-type: none"> ● Menyusun metode dalam pengumpulan data melalui wawancara dengan manpro ● Membagi rincian tugas untuk tim 	
Analysis (Analisa)	<ul style="list-style-type: none"> ● Mahasiswa mampu menjelaskan berdasarkan permasalahan yang ada. ● Merumuskan usulan proyek ● Menyusun analisa kebutuhan sistem 	2 Minggu
Design (Perancangan)	<ul style="list-style-type: none"> ● Mahasiswa menyusun rancangan mockup untuk sistem yang akan dikembangkan. ● Mahasiswa merancang flowchart, dan UML 	2 Minggu
Implementasi	<ul style="list-style-type: none"> ● Melakukan Pengujian Fungsional Sistem. ● Mahasiswa membuat produk/ project. ● Mahasiswa 	3 Minggu

	membuat prototype tampilan sistem. <ul style="list-style-type: none"> ● Mahasiswa membuat laporan proyek 	
Testing dan Debugging	<ul style="list-style-type: none"> ● Mahasiswa menguji fitur kecil dalam aplikasi ● Mahasiswa menemukan dan memperbaiki bug/error dalam aplikasi yang ditemukan saat pengujian 	2 Minggu
Maintenance	<ul style="list-style-type: none"> ● Mahasiswa menjaga dan meningkatkan aplikasi setelah rilis, agar tetap berjalan lancar dan terus berkembang sesuai kebutuhan pengguna 	2 Minggu

E. Silahkan Eksplorasi ***Aplikasi pembanding*** sejenis dengan proyek yang akan Anda Kembangkan.

Sebagai aplikasi pembanding, kami mengacu pada Honda Indonesia (BR-V N7X), terutama pada fitur katalog produk, halaman home, navbar, dan detail spesifikasi produk. Berikut adalah fitur utama yang menjadi inspirasi:

Fitur Honda BR-V N7X	Deskripsi	Relevansi dengan Proyek
Home Page interaktif	Menampilkan informasi utama dengan desain visual yang menarik	Bisa diterapkan pada proyek agar lebih user friendly
Katalog Produk	Menampilkan berbagai model- model mobil dengan gambar, fitur utama dan harga	Fitur ini bisa menjadi referensi kami
Detail produk & Spesifikasi	Halaman khusus yang menjelaskan fitur mobil, spesifikasi teknis dan keunggulan produk	Bisa diterapkan pada halaman spesifikasi produk pada proyek kami
Desain Minimalis & Profesional	Menggunakan kombinasi warna yang sederhana tetapi tetap elegan.	Bisa diterapkan dalam proyek untuk memberikan tampilan yang bersih, profesional, dan menarik secara visual.
Navigasi yang Jelas	Menu yang terstruktur dengan baik untuk memudahkan pengguna menemukan informasi.	Harus diterapkan dalam Proyek karena memiliki navigasi yang mudah dipahami agar pengguna dapat mengakses fitur dengan cepat.
Dropdown Menu	Beberapa kategori memiliki sub-menu untuk menampilkan lebih banyak opsi tanpa mengacaukan tampilan utama.	Bisa diterapkan dalam proyek jika memiliki banyak kategori, sehingga lebih terorganisir.
Responsif untuk Desktop & Mobile	Memastikan navbar tetap mudah diakses di berbagai ukuran layar.	harus diterapkan dalam Proyek karna proyek harus mendukung tampilan yang adaptif agar nyaman digunakan di semua

		perangkat.
--	--	------------

F. Sebutkan kebutuhan **tools** untuk mengerjakan proyek PBL Anda

Fase Proyek	Tools
Planning	Microsoft word, dan Gmeet
Analysis	Microsoft word
Design	Canva dan Figma
Implementation	VSCode, Xampp dan MySQL Database
Testing & Debugging	Selenium dan Browser
Maintenance	Trello/jira dan GitHub

G. Sebutkan **peralatan dan biaya** yang diperlukan untuk menyelesaikan proyek Anda.

Fase Proyek	Peralatan	Estimasi biaya
Planning	Laptop/PC	
	Alat untuk kolaborasi (Google meet)	
Analysis	Laptop/PC	
	Software analisis data (Ms.Word)	
Design	Laptop/PC	
	Software design (Canva dan Figma)	

Implementation	Laptop/PC	
	Visual Studio Code	
	Xampp dan MySQL database	
Testing & Debugging	Laptop/PC	
	Selenium	
	Browser (Google Chrome)	
Maintenance	Laptop/PC	
	Trello	
	GitHub	

Jawab pertanyaan berikut ini sesuai dengan proyek PBL yang Anda kerjakan:

Untuk melaksanakan proyek pembangunan aplikasi tersebut.

(i) Feasibility study seperti apa yang akan Anda lakukan?

Feasibility study yang akan dilakukan untuk proyek Aplikasi Jual Beli Mobil Berbasis Web mencakup beberapa aspek utama :

Feasibility Teknis

1. Menentukan teknologi yang akan digunakan (bahasa pemrograman, framework, database, server, dll.).
2. Evaluasi kebutuhan infrastruktur (server hosting, keamanan data, dan performa).
3. Analisis integrasi dengan sistem pihak ketiga (payment gateway, dll.).

Feasibility Hukum

1. Kepatuhan terhadap regulasi perlindungan konsumen.
2. Perlindungan data pengguna sesuai dengan kebijakan privasi.

(ii) Apa keuntungan dan kerugian ketika membuat proyek diatas dari sisi waktu dan biaya?

1. Keuntungan

Dari Sisi Waktu:

-
- **Aksesibilitas yang Lebih Cepat** → Aplikasi berbasis web bisa diakses kapan saja tanpa perlu instalasi, sehingga mempercepat proses jual beli mobil.-
 - **Efisiensi dalam Pembuatan** → Dibandingkan aplikasi mobile, pengembangan web bisa lebih cepat karena tidak perlu menyesuaikan dengan berbagai platform (iOS/Android).
 - **Otomatisasi Proses** → Sistem akan menangani pencarian, pengelolaan produk, dan transaksi secara otomatis tanpa perlu interaksi manual setiap saat.
 - **Memudahkan Pengelolaan Data** → Riwayat pembelian dan transaksi dapat direkap secara otomatis, menghemat waktu bagi penjual dan pembeli.

Dari Sisi Biaya:

- **Biaya Infrastruktur Lebih Murah** → Hosting dan domain relatif lebih murah dibandingkan membangun aplikasi native untuk Android/iOS.
- **Tidak Memerlukan Hardware Khusus** → Bisa diakses dari browser, sehingga tidak perlu membeli perangkat khusus untuk menjalankan aplikasi.
- **Pemeliharaan Lebih Mudah** → Update sistem bisa langsung dilakukan di server tanpa harus meminta pengguna untuk memperbarui aplikasi mereka.
- **Tidak Perlu Biaya Distribusi** → Tidak seperti aplikasi mobile yang perlu biaya distribusi di App Store/Play Store, aplikasi web bisa langsung diakses oleh pengguna.

2. Kerugian

Dari Sisi Waktu:

- **Proses Pengembangan yang Memakan Waktu** → Membutuhkan waktu untuk membangun backend, frontend, dan mengintegrasikan sistem database.

-
- **Pengujian yang Lebih Lama** → Karena berbasis web, aplikasi harus diuji di berbagai perangkat dan browser agar tetap responsif.
 - **Pemeliharaan dan Perbaikan Butuh Waktu** → Jika ada bug atau error, developer harus memperbaiki langsung di server agar tidak mengganggu pengguna.

Dari Sisi Biaya:

- **Biaya Server & Hosting** → Memerlukan server yang cukup kuat untuk menangani banyak pengguna secara bersamaan.
- **Keamanan Data Butuh Investasi** → Perlu pengamanan ekstra untuk melindungi data pengguna dan transaksi dari ancaman cyber.
- **Pengembangan Awal Bisa Mahal** → Jika menggunakan tenaga profesional untuk pengembangan backend dan frontend, biaya awal bisa cukup besar.
- **Biaya Pemeliharaan Berkala** → Harus mengalokasikan dana untuk update fitur, backup data, dan perlindungan keamanan secara rutin.

(iii) Berapa jumlah anggota tim yang dibutuhkan untuk menyelesaikan proyek tersebut sesuai task yang dibutuhkan? Jelaskan alasannya.

1. Frontend Developer (1 orang)

- Mendesain tampilan aplikasi agar user-friendly dan responsif di berbagai perangkat (desktop & mobile).
- Mengimplementasikan desain UI ke dalam kode menggunakan HTML, CSS, dan JavaScript.
- Memastikan navigasi dan interaksi dalam aplikasi berjalan dengan baik.

2. Backend Developer (1 orang)

- Mengembangkan fitur utama aplikasi menggunakan PHP dan MySQL.
- Mengelola database, autentikasi pengguna, serta fitur keamanan sistem.
- Membuat API jika dibutuhkan untuk komunikasi antara frontend dan backend.

3. Full Stack Developer (2 orang)

- Bertindak sebagai penghubung antara frontend dan backend.
- Membantu backend dalam membangun sistem dan memastikan data dikirim serta diterima dengan benar di frontend.
- Mengatasi bug dan optimasi performa aplikasi.

(iv) Apakah saja risiko yang ada? Jelaskan.

Tahapan Pekerjaan	Bahaya	Resiko	Pengendalian Resiko
Desain UI/UX	Kesalahan desain yang tidak sesuai standar aksesibilitas	Pengguna dengan kebutuhan khusus kesulitan menggunakan aplikasi	<ul style="list-style-type: none"> ● Melakukan riset sebelum memulai design. ● Memiliki prinsip yang konsisten dalam mendesain. ● Fokus pada proyek yang dikerjakan. Menghindari penggunaan pop-up yang mengganggu

Penerapan integrasi database	Konflik saat mengintegrasikan database	Kegagalan sinkronisasi data	<ul style="list-style-type: none">● Menggunakan Teknik Asynchronous Communication● Gunakan transactional operations untuk memastikan integritas data di server.● Manajemen State yang Baik di Front-End● Gunakan caching yang efektif untuk mengurangi beban server dan mempercepat sinkronisasi data
Pengembangan aplikasi backend	Kesalahan coding yang menyebabkan celah keamanan	Kebocoran data pengguna	<ul style="list-style-type: none">● Gunakan Debugging Tools.● Melakukan testing dan

			validasi. <ul style="list-style-type: none"> • Dokumentasi yang baik
Pengujian aplikasi	Bug atau error yang tidak terdeteksi	Aplikasi gagal berfungsi saat digunakan	<ul style="list-style-type: none"> • Menggunakan Debugging Tools • Periksa Console Browser • Cek validitas kode • Debugging Back-End (Server-Side)
Peluncuran aplikasi	Masalah kompatibilitas browser atau perangkat	Pengguna tidak dapat mengakses aplikasi dengan lancar	<ul style="list-style-type: none"> • Optimalkan performa website dengan mengurangi waktu loading, meminimalkan ukuran file, dan menggunakan teknik caching. • Pastikan antarmuka pengguna

			(UI) dirancang dengan sederhana dan intuitif, sehingga pengguna dapat memahami fungsi aplikasi dengan cepat tanpa kebingungan . <ul style="list-style-type: none"> ● Memberikan petunjuk yang jelas Feedback yang jelas dan cepat ● Melakukan pemeliharaan dan pembaruan yang rutin.
--	--	--	---

3. KOMPONEN PBL

Tim PBL harus berdiskusi dengan **Manpro** dalam tahap perencanaan.

Manpro dan anggota tim PBL menggunakan **SIAP-PBL**

(<https://pbl.polibatam.ac.id/>) dalam mengelola proyek PBL yang akan dikerjakan. Tim PBL

Bersama Manpro menyelesaikan **RPP** sebagai dokumen perencanaan Pelaksanaan proyeknya

kemudian menguploadnya ke **SIAP-PBL** sebagai tanda tahap perencanaan telah siap dilanjutkan ke Tahap berikutnya. Semua dokumentasi penyelesaian proyek PBL dibuat berdasarkan **Panduan PBL**.

Semua dokumen dan aturan pelaksanaan PBL dapat ditemukan melalui link e-learning berikut :
[Proyek Perangkat Lunak Industri/IF419](#)

4. SYARAT Pengerjaan Modul

Setiap mahasiswa mengerjakan modul ini secara berkelompok berdasarkan Tim PBL nya. Namun setiap orang wajib memiliki peran tersendiri dalam penyusunan hasil praktikum minggu ini.

Analisis Kebutuhan -

Requirement analysis

Analisis kebutuhan dilakukan sebelum dimulainya pengembangan sistem. Tujuan dari tahap ini adalah untuk memastikan apa yang dibutuhkan oleh user di dalam sistem yang akan dibangun, sudah tergambar oleh seorang **system analyst**. Dalam **memahami proses bisnis dan knowledge** berhubungan dengan sistem yang akan dibangun, **system analyst dibantu oleh business analyst** (Dennis. Alan, 2016). Output dari tahap ini adalah **business process identification** yang digambarkan pada **use case diagram**, **business process modeling** yang digambarkan pada **business process model and notation** dan **business process realization** yang digambarkan pada **sequence diagram** (Wahono, 2020). Output tersebut merupakan bagian dari **system requirements specification**.

1. ANALYSIS KEBUTUHAN PROYEK PBL

A. Deskripsi umum sistem yang akan dibangun.

Aplikasi jual beli mobil berbasis web ini dirancang untuk mempermudah transaksi antara penjual dan pembeli kendaraan. Pengguna dapat membuat akun sebagai penjual atau pembeli, lalu mengakses fitur utama aplikasi. Penjual dapat mengunggah informasi mobil yang akan dijual, termasuk foto, harga, spesifikasi, dan kontak. Sementara itu, pembeli dapat mencari mobil berdasarkan kategori, harga, atau merek tertentu serta melihat ulasan dari pengguna lain sebelum memutuskan untuk membeli.

B. Jenis pengguna sistem.

- Penjual
- Pembeli

C. Kebutuhan Fungsional system.

No	Kode	Aktor	Keterangan
1.	FR-1	Penjual	Melakukan Login
2.	FR-2	Penjual	Melakukan Pengelolaan Produk
3.	FR-3	Penjual	Mengelola Pesanan Pembelian
4.	FR-4	Penjual	Melakukan Logout
5.	FR-5	Pembeli	Melakukan Registrasi
6.	FR-6	Pembeli	Melakukan Login
7.	FR-7	Pembeli	Melakukan Pengelolaan Profil
8.	FR-8	Pembeli	Melakukan Pencarian Produk
9.	FR-9	Pembeli	Menambahkan Produk ke Daftar Keinginan/Wishlist
10.	FR-10	Pembeli	Melakukan Pengajuan Pembelian
11.	FR-11	Pembeli	Mendapatkan Resi
12.	FR-12	Pembeli	Melakukan Logout

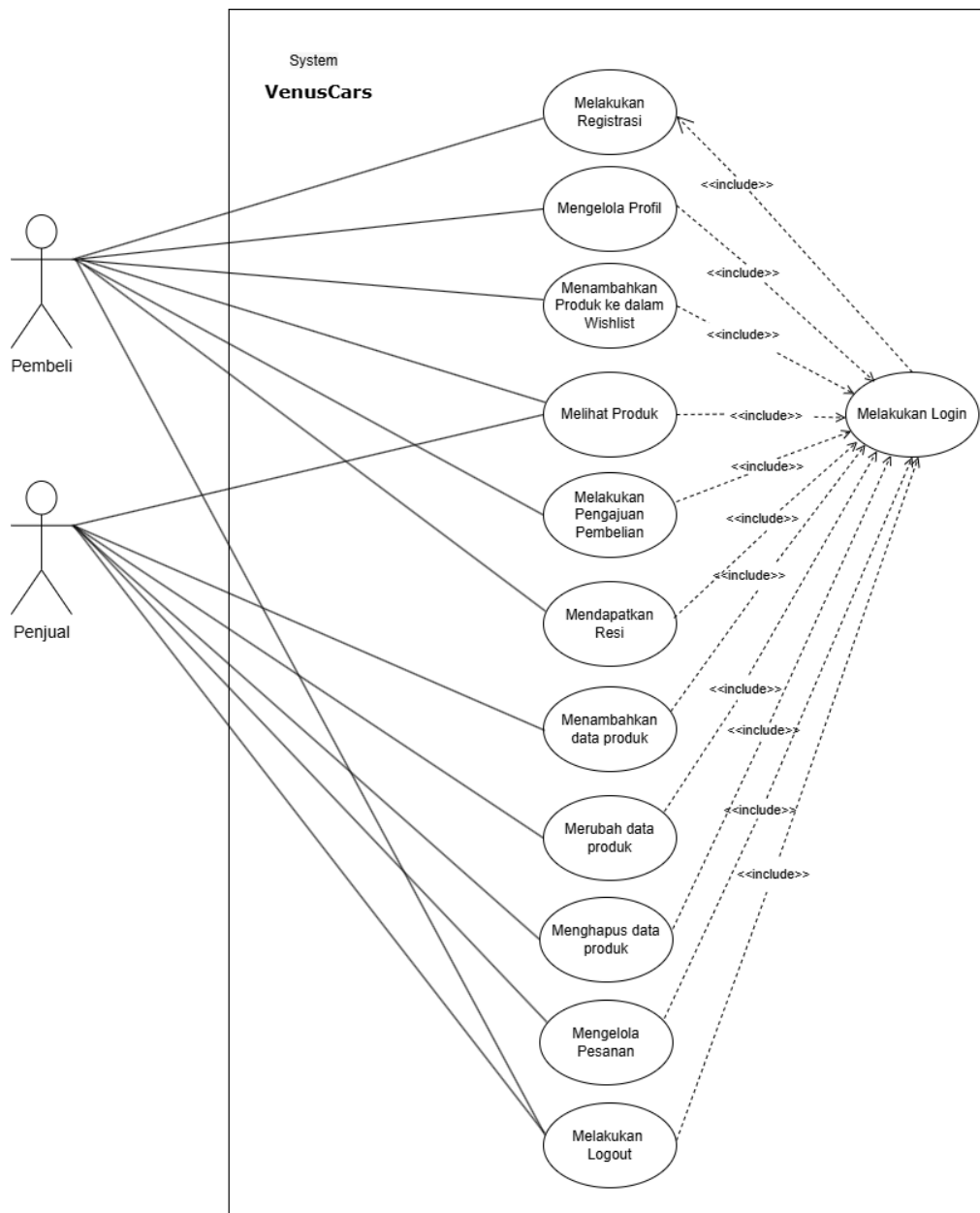
D. Kebutuhan **Non-Fungsional** system.

No	Kode	Keterangan
1.	NFR-1	Source kode aplikasi harus terstruktur dengan baik dan mudah dipelihara oleh pengembang yang berbeda dan mengikuti Konvensi Penulisan Kode Program
2.	NFR-2	Desain antarmuka aplikasi harus ramah pengguna agar mudah digunakan oleh pembeli dan penjual
3.	NFR-3	Antarmuka pengguna intuitif dan dapat di navigasi
4.	NFR-4	Aplikasi harus dapat digunakan di semua perangkat, seperti Laptop/PC, Tablet dan Handphone

E. Peta pengguna dan terhadap **kebutuhan fungsional** system.

ID	Aktor	Deskripsi	Tugas Pengguna	Prioritas	Status
FR-1	Penjual	Login	Masuk ke sistem sebagai penjual menggunakan username & password	Tinggi	Belum dikembangkan
FR-2	Penjual	Melakukan Pengelolaan Produk	Menambah, mengedit, menghapus, dan melihat daftar mobil yang dijual	Tinggi	Belum dikembangkan
FR-3	Penjual	Mengelola Pembelian Pesanan	Penjual mengakses halaman daftar pesanan Penjual memilih pesanan yang ingin di kelola Penjual memperbarui status pesanan (di konfirmasi, dikemas, dikirim, selesai) Sistem menyimpan	Tinggi	Belum dikembangkan

			perubahan status pesanan dan menampilkan notifikasi kepada pembeli		
FR-4	Penjual	Logout	Keluar dari sistem sebagai penjual	Tinggi	Belum dikembangkan
FR-5	Pembeli	Registrasi	Mendaftar akun baru untuk membeli mobil	Tinggi	Belum dikembangkan
FR-6	Pembeli	Login	Masuk ke sistem sebagai pembeli menggunakan username & password	Tinggi	Belum dikembangkan
FR-7	Pembeli	Melakukan Pengelolaan Profil	Mengubah data akun seperti nama, no hp, alamat, maupun password. Serta menambahkan foto profil.	Menengah	Belum dikembangkan
FR-8	Pembeli	Melakukan Pencarian Produk	Mencari mobil berdasarkan merek, harga, tahun, dan filter lainnya	Tinggi	Belum dikembangkan
FR-9	Pembeli	Menambahkan Produk ke Daftar Keinginan/Wishlist	Menyimpan mobil yang diminati ke daftar wishlist	Menengah	Belum dikembangkan
FR-10	Pembeli	Melakukan Pengajuan Pembelian	Mengajukan pembelian mobil ke penjual	Tinggi	Belum dikembangkan
FR-11	Pembeli	Mendapatkan Resi	Menerima bukti transaksi atau nomor resi pengiriman	Menengah	Belum dikembangkan
FR-12	Pembeli	Logout	Keluar dari sistem sebagai pembeli	Tinggi	Belum dikembangkan

F. Use case diagram.

G. Skenario Use case Diagram

Use Case Name	Login
Primary Actor	Penjual, Pembeli
Preconditions	<ol style="list-style-type: none"> 1. Pengguna sudah terdaftar di sistem 2. Pengguna memiliki kredensial login yang valid.
Postconditions	Pengguna berhasil masuk ke sistem dan dapat mengakses fitur yang sesuai dengan rolenya.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna membuka halaman login. 2. Pengguna memasukkan email dan password. 3. Sistem memverifikasi kredensial login. 4. Jika valid, pengguna diarahkan ke dashboard sesuai dengan rolenya.
Alternative Flow	<p>Jika kredensial salah:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan kesalahan. 2. Pengguna dapat mencoba login kembali atau mereset password. <p>Jika akun belum terdaftar:</p> <ol style="list-style-type: none"> 3. Sistem menampilkan opsi untuk melakukan registrasi.

Use Case Name	Registrasi
Primary Actor	Pembeli
Preconditions	Pengguna belum memiliki akun di sistem.
Postconditions	Pengguna berhasil mendaftar dan dapat login ke sistem.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna mengakses halaman registrasi. 2. Pengguna mengisi data yang diperlukan (email, fullname, username, password) 3. Sistem memvalidasi data. 4. Jika valid, sistem menyimpan data pengguna dan menampilkan notifikasi berhasil login. 5. Pengguna dapat melakukan login menggunakan akun baru.

Alternative Flow	Jika data tidak valid: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan kesalahan. 2. Pengguna dapat memperbaiki data dan mencoba kembali.
-------------------------	--

Use Case Name	Mengelola Profil
Primary Actor	Pembeli
Preconditions	<ol style="list-style-type: none"> 1. Pengguna telah login ke sistem. 2. Profil pengguna sudah ada dalam database.
Postconditions	Jika berhasil, sistem menyimpan perubahan profil pengguna. Jika gagal, sistem menampilkan pesan kesalahan dan tidak menyimpan perubahan.
Main Succes Scenario	<p>Melihat Profil :</p> <ol style="list-style-type: none"> 1. Pengguna mengakses menu edit profil 2. Sistem menampilkan data profil pengguna saat ini. <p>Mencari data profil yang ingin di kelola :</p> <ol style="list-style-type: none"> 1. Pengguna memilih bagian data profil tertentu yang ingin di kelola. <p>Menghapus data profil :</p> <ol style="list-style-type: none"> 1. Pengguna memilih opsi “detele” pada bagian yang ingin dihapus 2. Sistem meminta konfirmasi 3. Jika dikonfirmasi, sistem menghapus data tersebut dan menampilkan pesan “Data berhasil dihapus” <p>Menambah atau Mengubah informasi profil :</p> <ol style="list-style-type: none"> 1. Pengguna memasukkan data baru atau mengedit data yang sudah ada 2. Sistem memverifikasi input 3. Jika valid, pengguna menyimpan perubahan <p>Menyimpan perubahan :</p> <ol style="list-style-type: none"> 1. Pengguna menekan tombol save 2. Sistem memperbarui data di database 3. Sistem menampilkan notifikasi “Profil berhasil diperbarui”
Alternative Flow	Pengguna membatalkan perubahan : <ol style="list-style-type: none"> 1. Pengguna memilih tombol “batal” sebelum menyimpan.

	<ol style="list-style-type: none"> 2. Sistem membatalkan perubahan dan kembali ke halaman edit profil <p>Kesalahan validasi data :</p> <ol style="list-style-type: none"> 1. Pengguna memasukkan format data yang salah 2. Sistem menampilkan pesan kesalahan dan meminta pengguna memperbaiki input. 3. Pengguna memperbaiki data dan mencoba lagi.
--	--

Use Case Name	Menambahkan Produk ke dalam Wishlist
Primary Actor	Pembeli
Preconditions	Pengguna telah login ke sistem.
Postconditions	Produk berhasil ditambahkan ke wishlist pengguna.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna mengakses halaman shop 2. Pengguna memilih produk yang ingin ditambahkan ke wishlist. 3. Pengguna menekan tombol "Tambahkan ke Wishlist". 4. Sistem menyimpan produk ke dalam wishlist pengguna. 5. Pengguna dapat melihat produk di halaman wishlist.
Alternative Flow	<p>Jika produk sudah ditambahkan ke wishlist:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa produk sudah berhasil ditambahkan ke wishlist.

Use Case Name	Melihat Produk
Primary Actor	Penjual, Pembeli
Preconditions	Pengguna telah login ke sistem.
Postconditions	Pengguna berhasil melihat daftar produk yang tersedia.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna mengakses halaman produk. 2. Sistem menampilkan produk yang tersedia. 3. Pengguna dapat mencari atau memfilter produk berdasarkan kategori. 4. Pengguna memilih produk untuk melihat detailnya.
Alternative Flow	<p>Jika tidak ada produk tersedia:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa tidak ada produk yang tersedia.

Use Case Name	Melakukan Pengajuan Pembelian
----------------------	-------------------------------

Primary Actor	Pembeli
Preconditions	<ol style="list-style-type: none"> 1. Pengguna telah login ke sistem. 2. Produk tersedia dalam stok.
Postconditions	Sistem menerima dan mencatat pengajuan pembelian.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna memilih produk yang ingin dibeli. 2. Pengguna menekan tombol "Ajukan Pembelian". 3. Sistem meminta pengguna untuk mengisi informasi pengiriman. 4. Pengguna mengisi data dan mengkonfirmasi pembelian. 5. Sistem mencatat pengajuan pembelian dan menampilkan statusnya.
Alternative Flow	Jika stok habis: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa produk tidak tersedia.

Use Case Name	Mendapatkan Resi
Primary Actor	Pembeli
Preconditions	Pengguna telah melakukan pembelian dan pesanan sedang diproses.
Postconditions	Pengguna mendapatkan nomor resi.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna mengakses halaman transaksi 2. Sistem menampilkan status pesanan dan nomor resi.
Alternative Flow	Jika terjadi kesalahan dalam sistem: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa produk tidak tersedia.

Use Case Name	Menambahkan Data Produk
Primary Actor	Penjual
Preconditions	Penjual telah login ke sistem.
Postconditions	Data produk berhasil ditambahkan ke dalam sistem.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Penjual mengakses halaman daftar produk. 2. Penjual menekan tombol "Tambah Produk". 3. Penjual mengisi informasi produk. 4. Penjual menyimpan data produk. 5. Sistem menyimpan dan menampilkan produk di daftar produk.
Alternative Flow	Jika data produk tidak valid:

	1. Sistem menampilkan pesan bahwa produk tidak tersedia.
--	--

Use Case Name	Merubah Data Produk
Primary Actor	Penjual
Preconditions	Penjual telah login ke sistem.
Postconditions	Data produk berhasil diperbarui di sistem.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Data produk berhasil diperbarui di sistem. 2. Penjual memilih produk yang ingin diubah. 3. Penjual mengedit informasi produk. 4. Penjual menyimpan perubahan. 5. Sistem memperbarui dan menampilkan data yang telah diperbarui.
Alternative Flow	Jika data produk tidak valid: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa produk tidak tersedia.

Use Case Name	Menghapus Data Produk
Primary Actor	Penjual
Preconditions	<ol style="list-style-type: none"> 1. Penjual telah login ke sistem. 2. Produk yang akan dihapus sudah terdaftar dalam sistem.
Postconditions	Produk berhasil dihapus dari sistem.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Penjual mengakses halaman daftar produk. 2. Penjual memilih produk yang ingin dihapus. 3. Penjual menekan tombol "DELETE". 4. Sistem menampilkan konfirmasi penghapusan. 5. Jika dikonfirmasi, sistem menghapus produk dari database.
Alternative Flow	Jika produk tidak ditemukan: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa produk tidak tersedia. 2. Sistem menampilkan pesan error dan meminta penjual mencoba kembali nanti.

Use Case Name	Mengelola pesanan
Primary Actor	Penjual
Preconditions	<ol style="list-style-type: none"> 1. Penjual telah login ke sistem 2. Ada pesanan yang masuk ke sistem

Postconditions	Penjual berhasil memperbarui status pesanan
Main Succes Scenario	<ol style="list-style-type: none"> 1. Penjual mengakses halaman daftar pesanan 2. Penjual memilih pesanan yang ingin di kelola 3. Penjual memperbarui status pesanan (di konfirmasi, dikemas, dikirim, selesai) 4. Sistem menyimpan perubahan status pesanan dan menampilkan notifikasi kepada pembeli
Alternative Flow	Jika pesanan tidak ditemukan: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan bahwa pesanan tidak tersedia

Use Case Name	Logout
Primary Actor	Penjual, Pembeli
Preconditions	Pengguna telah login ke sistem.
Postconditions	Pengguna berhasil keluar dari sistem dan sesi berakhir.
Main Succes Scenario	<ol style="list-style-type: none"> 1. Pengguna menekan tombol "Logout" 2. Sistem menghapus sesi pengguna. 3. Pengguna diarahkan ke halaman login.
Alternative Flow	Jika terjadi kesalahan saat logout: <ol style="list-style-type: none"> 1. Sistem menampilkan pesan error dan meminta pengguna mencoba kembali nanti.

2. CHECKPOINT

- a. Berapa jumlah pengguna aplikasi?

Aplikasi ini digunakan oleh dua jenis pengguna, yaitu Penjual dan Pembeli.

- b. Berapa jumlah aktor pada diagram usecase yang anda buat dalam diagram usecase?

Diagram yang dibuat menunjukkan 2 aktor, yaitu Pembeli dan Penjual.

- c. Apakah jumlahnya sama? Jika tidak, mengapa?

Ya, jumlah pengguna dan aktor dalam diagram use case yang dibuat adalah sama, yaitu 2.

-
- d. Berapa jumlah kebutuhan fungsional PBL Anda?
Kebutuhan fungsional dalam PBL kami berjumlah 12.
- e. Berapa jumlah usecase yang anda buat dalam diagram usecase?
Jumlah usecase yang telah dibuat dalam diagram use case berjumlah 12.
- f. Apakah Jumlahnya sama? Jika tidak mengapa?
Jumlahnya sama, jumlah use case dalam diagram sesuai dengan kebutuhan fungsional yang telah ditentukan.
- g. Apakah semua usecase sudah terhubung dengan aktor atau usecase lain? Jika tidak mengapa?
Ya, semua use case memiliki aktor (Penjual atau Pembeli), jadi tidak ada use case yang berdiri sendiri tanpa aktor. Beberapa use case sudah menggunakan hubungan <<include>> ke "Melakukan Login", yang benar karena fitur tersebut membutuhkan autentikasi.
- h. Perhatikan Peta pengguna dan terhadap kebutuhan fungsional system. Yang telah anda buat di minggu sebelumnya
- i. Apakah peta tersebut sesuai dengan keterhubungan usecase dengan aktor? Jika tidak mengapa?
Ya, peta pengguna terhadap kebutuhan fungsional sistem sudah sesuai dengan keterhubungan use case dengan aktor.
- j. Ada berapa usecase terhubung dengan usecase lain? Berapa usecase <> dan berapa <>?
Usecase yang terhubung dengan usecase lain ada 11.
Usecase <<include>> : ada 11
Usecase <<extend>> : tidak ada
- k. Apakah skenario usecase sejumlah usecase? Jika tidak mengapa?
Ya, jumlah skenario usecase sama dengan jumlah usecase
- l. Apakah preconditions dan postconditions di setiap skenario usecase sudah tercapai? Jika tidak mengapa?
Tidak, karena aplikasi belum benar-benar dikembangkan. Sistem masih dalam tahap rancangan, sehingga preconditions dan

postconditions belum bisa diuji secara nyata.

3. SYARAT Pengerjaan Modul

Setiap mahasiswa mengerjakan modul ini secara berkelompok berdasarkan Tim PBL nya. Namun setiap orang wajib memiliki peran tersendiri dalam penyusunan hasil praktikum minggu ini.

Perancangan - *Design*

Analisis kebutuhan dilakukan sebelum dimulainya pengembangan sistem. Tujuan dari tahap ini adalah untuk memastikan apa yang dibutuhkan oleh user di dalam sistem yang akan dibangun, sudah tergambar oleh seorang **system analyst**. Dalam **memahami proses bisnis dan knowledge** berhubungan dengan sistem yang akan dibangun, **system analyst dibantu oleh business analyst** (Dennis. Alan, 2016). Output dari tahap ini adalah **business process identification** yang digambarkan pada **use case diagram, business process modeling** yang digambarkan pada **business process model and notation** dan **business process realization** yang digambarkan pada **sequence diagram** (Wahono, 2020). Output tersebut merupakan bagian dari **system requirements specification**.

1. PERANCANGAN PBL

Lihat kembali dan lanjutkan tugas Anda dari minggu lalu.

Berdasarkan SKPL yang telah anda susun bersama tim minggu lalu, silahkan buat rancangan arsitekturnya.

*Lanjutkan pengerjaan modul praktikum minggu sebelumnya. Lengkapi dengan menyelesaikan modul praktikum minggu ini.

Lengkapi SKPL/ SRS yang berisi informasi PBL Anda. SKPL dilengkapi dengan beberapa perancangan untuk pertemuan minggu ini silahkan menyelesaikan

:

- Perancangan arsitektur sistem

Untuk membangun aplikasi jual beli mobil berbasis web, saya menggunakan arsitektur Model-View-Controller (MVC) yang disediakan oleh framework Laravel.

Arsitektur ini membantu memisahkan antara tampilan, logika aplikasi, dan pengelolaan data, sehingga sistem menjadi lebih terstruktur dan mudah dikembangkan.

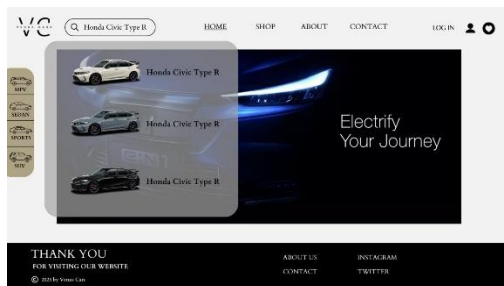
Struktur ini juga selaras dengan konsep 3-tier architecture, yang membagi sistem menjadi 3 lapisan utama :

- Presentation Layer (View)
 - Menyajikan tampilan antarmuka kepada pengguna
 - Dibangun menggunakan Blade, HTML, Tailwind CSS
- Application Layer (Controller & Logic)
 - Mengatur alur data, validasi, proses bisnis
 - Ditangani oleh Controller, Middleware, Service di Laravel
- Data Layer (Model & Database)
 - Menyimpan dan mengambil data
 - Menggunakan Eloquent ORM dan database MySQL

Dengan pendekatan ini, pengembangan aplikasi menjadi modular, fleksibel, dan lebih mudah untuk pengujian serta pemeliharaan jangka panjang.

■ Perancangan antarmuka (minimal 5 fitur aplikasi)

1. Fitur pencarian

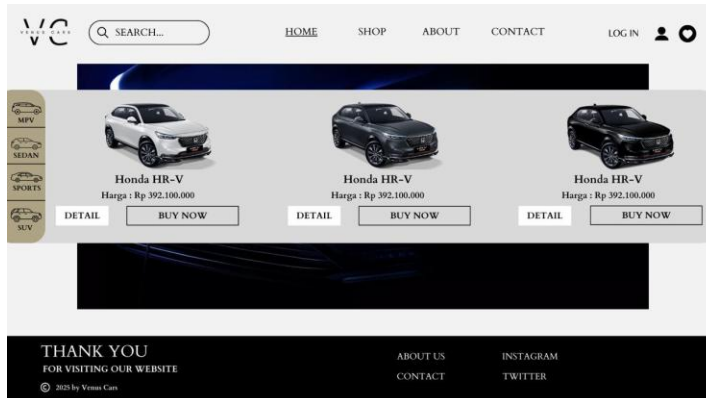


Fitur pencarian ini memungkinkan pengguna:

- Menemukan mobil berdasarkan kata kunci yang mereka masukkan.
- Menampilkan hasil dalam bentuk gambar dan nama mobil, sehingga memudahkan pengguna memilih produk yang mereka inginkan.

- Disusun dengan desain modern dan user-friendly, membuat pencarian menjadi cepat dan mudah dipahami.

2. Fitur kategori



Fitur kategori ini memungkinkan pengguna:

- Menjelajahi mobil berdasarkan tipe kendaraan, seperti MPV, SEDAN, SPORTS, dan SUV, sehingga pengguna dapat langsung menemukan jenis mobil sesuai kebutuhan mereka.
- Mempercepat proses pencarian, karena pengguna tidak perlu menelusuri semua produk satu per satu cukup memilih kategori yang sesuai di bagian kiri halaman.
- Disusun dengan ikon dan teks yang intuitif, membuat navigasi menjadi lebih mudah, cepat, dan nyaman digunakan bahkan untuk pengguna baru.

3. Fitur show produk



Fitur Show Produk ini memungkinkan pengguna:

- Melihat daftar produk mobil yang tersedia secara lengkap, mulai dari tipe, gambar, hingga harga awal setiap mobil, seperti Honda Mobilio, New Honda City, Honda Civic

Type R, dan Honda HR-V.

- Menampilkan informasi produk secara visual dan informatif, sehingga pengguna dapat langsung membandingkan mobil berdasarkan tampilan dan harga.
- Disertai tombol “Add to Wishlist”, yang memudahkan pengguna menyimpan produk favorit mereka untuk dibuka kembali nanti.
- Didesain dengan tata letak yang rapi dan elegan, memudahkan pengguna menjelajahi berbagai pilihan tanpa merasa bingung atau kewalahan.

4. Fitur CRUD

VC

TRADING CENTER

Seller Center

Log out

Dashboard

Daftar Produk

Daftar Pesanan

Daftar Produk

TAMBAH PRODUK

NO	PRODUK	PENJUALAN	HARGA	STOK	AKSI
1	 Honda Civic Type R 1 unit	Rp 1,455,500,000	2	<div><div></div><div></div></div>	
2	 Honda Civic Type R 1 unit	Rp 1,455,500,000	2	<div><div></div><div></div></div>	
3	 Honda Civic Type R 1 unit	Rp 1,455,500,000	1	<div><div></div><div></div></div>	
4	 Honda HR -V 1 unit	Rp 392,100,000	1	<div><div></div><div></div></div>	
5	 Honda HR -V 1 unit	Rp 392,100,000	3	<div><div></div><div></div></div>	
6	 Honda HR -V 1 unit	Rp 392,100,000	4	<div><div></div><div></div></div>	

Fitur ini memungkinkan penjual untuk mengelola daftar produk secara menyeluruh, mencakup empat aksi utama CRUD:

Create, Read, Update, dan Delete.

- Create (Tambah Produk)

Penjual dapat menambahkan produk baru ke dalam daftar menggunakan tombol "TAMBAH PRODUK".

- Read (Lihat Daftar Produk)

Menampilkan semua produk yang dijual, lengkap dengan informasi:

Gambar produk

Nama produk

Jumlah penjualan

Harga

Stok tersedia



- Update (Edit Produk)

Tombol "EDIT" memungkinkan penjual untuk mengubah detail produk seperti nama, harga, atau stok.

- Delete (Hapus Produk)

Tombol "DELETE" digunakan untuk menghapus produk dari daftar.

5. Fitur menampilkan semua pesanan

No	Produk	Total Pesanan	Status	Aksi
1	 Honda Mobilio	Rp 243,300,000	Dikemas	Print
2	 New Honda City	Rp 402,000,000	Dikemas	Print

Fitur ini menampilkan semua pesanan yang masuk ke penjual, sehingga penjual bisa memantau, memproses, dan mengelola setiap transaksi secara efisien.

Fungsi dan Komponen Utama:

- Daftar pesanan ditampilkan dalam bentuk tabel yang memuat informasi penting:

Nomor urut

Gambar dan nama produk

Total harga pesanan

Status pesanan (misalnya: Dikemas)

Tombol aksi (contohnya: Print)

- Filter Status Pesanan

Terdapat tombol filter di bagian atas untuk memudahkan penjual menyaring pesanan berdasarkan status:

Semua

Konfirmasi

Dikemas

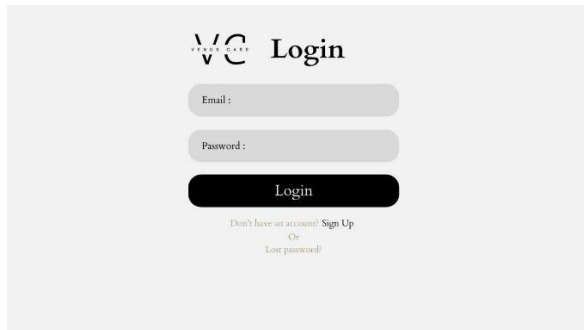
Dikirim

Selesai

- Tombol "Print"

Memungkinkan penjual mencetak data pesanan untuk keperluan pengemasan atau pengiriman.

6. Fitur Login



Fitur ini menampilkan halaman login dari sistem Venus Cars, yang digunakan untuk mengautentikasi pengguna sebelum mereka bisa mengakses fitur-fitur utama dalam aplikasi.

- Logo & Judul

Terdapat logo VC (Venus Cars) dan teks "Login" sebagai penanda halaman login.

Input Form:

- Email:

Field untuk memasukkan alamat email pengguna.

- Password:

Field untuk memasukkan kata sandi yang terkait dengan akun.

- Tombol Login:

Tombol besar berwarna hitam bertuliskan "Login" digunakan untuk mengirim data email dan password ke sistem untuk proses autentikasi.

- Navigasi Akun:

Tautan ke Sign Up bagi pengguna baru yang belum memiliki akun.

- Tautan Lost password?

untuk membantu pengguna memulihkan akun mereka jika lupa password.

7. Fitur register

Fitur ini menampilkan fitur registrasi (Sign Up) pada aplikasi Venus Cars. Di halaman ini, pengguna yang belum memiliki akun bisa mendaftar dengan mengisi beberapa data yang diperlukan, yaitu:

- Email – untuk keperluan verifikasi dan login
- Full Name – nama lengkap pengguna
- Username – sebagai identitas unik pengguna dalam system
- Password – untuk keamanan akun saat login

Setelah semua data diisi, pengguna bisa klik tombol Sign Up untuk membuat akun baru. Setelah berhasil registrasi, pengguna bisa langsung login dan mengakses fitur-fitur yang tersedia, seperti menambah produk, melihat pesanan, dan lainnya.

- Buat spesifikasi kebutuhan *hardware* dan *software* aplikasi Anda

1. Kebutuhan Perangkat Keras (Hardware)

a. Untuk Pengembang :

Komponen	Spesifikasi Minimum
Prosesor	Intel Core i5
RAM	20 GB
Hard Disk	512 GB SSD
Monitor	Resolusi 1920 x 1080
Perangkat Tambahan	Mouse

b. Untuk Pengguna :

Komponen	Spesifikasi Minimum
Perangkat	Laptop / PC / Smartphone
Browser	Chrome, Firefox, Microsoft Edge, Safari
RAM	2 GB
Koneksi Internet	Stabil (minimal 1 Mbps)

2. Kebutuhan Perangkat Lunak (Software)**c. Untuk Pengembang :**

Software	Keterangan
OS	Windows 11
Web Server	Laragon (Apache, MySQL, PHP)
Framework	Laravel (PHP Framework)
Editor Code	Visual Studio Code
Browser	Google Chrome, Microsoft Edge
DBMS	MySQL / phpMyAdmin

a. Untuk Pengguna :

Software	Keterangan
Web Browser	Chrome / Firefox / Microsoft Edge / Safari
OS	Android / Windows / iOS

2. CHECKPOINT

Jawab pertanyaan berikut :

- a. Dari lima pola arsitektur yang diperkenalkan dalam materi minggu ini, pola mana saja yang dapat diterapkan di sistem Anda (Pilih 2 yang paling sesuai)? Jelaskan dengan singkat penerapan dari pola arsitektur tersebut, seperti diberikan dalam contoh tiap pola di materi.

1. Model-View-Controller (MVC)

Sistem kami menggunakan framework Laravel yang menerapkan pola arsitektur MVC.

Model bertanggung jawab mengelola data mobil, akun pengguna, dan transaksi melalui Eloquent ORM.

View dibangun dengan Blade Template, HTML, dan Tailwind CSS untuk menyajikan antarmuka kepada pengguna.

Controller mengatur alur data antara model dan view, serta menangani logika aplikasi seperti proses pencarian mobil dan pengelolaan transaksi.

Pola ini memisahkan komponen logika, tampilan, dan data sehingga sistem lebih mudah dikembangkan dan dipelihara.

2. Layered Architecture (3-Tier Architecture)

Sistem kami juga menerapkan konsep 3-tier architecture yang membagi sistem menjadi tiga lapisan utama:

Presentation Layer: Menyajikan antarmuka pengguna, dibangun dengan HTML, Tailwind CSS, dan Blade.

Application Layer: Mengatur logika bisnis dan proses data, ditangani oleh Controller, Middleware, dan Service di Laravel.

Data Layer: Mengelola penyimpanan data menggunakan Eloquent ORM dan database MySQL.

Penerapan struktur ini membantu sistem menjadi modular dan lebih mudah dalam pengujian serta pengelolaan jangka panjang.

b. Pertimbangan apa yang Anda gunakan dalam merancang antar muka aplikasi?

Dalam merancang antarmuka aplikasi, kami mempertimbangkan beberapa hal berikut:

-
1. Kemudahan Penggunaan (User-Friendly): Antarmuka harus mudah dipahami dan digunakan oleh pengguna awam maupun pengguna berpengalaman.
 2. Responsif di Berbagai Perangkat: Desain harus bisa menyesuaikan dengan berbagai ukuran layar, baik desktop maupun smartphone.
 3. Konsistensi Visual: Warna, ikon, dan tata letak harus konsisten di seluruh halaman untuk menjaga kenyamanan dan kejelasan navigasi.
 4. Fokus pada Tujuan Pengguna: Fitur-fitur utama seperti pencarian mobil, filter harga, dan proses transaksi ditampilkan secara jelas dan mudah diakses.
 5. Aksesibilitas: Memastikan teks mudah dibaca, tombol cukup besar untuk diklik, dan navigasi jelas untuk semua kalangan pengguna.

c. **Jelaskan langkah perancangan yang anda lakukan?**

Langkah-langkah perancangan aplikasi yang kami lakukan adalah:

1. Menganalisis Kebutuhan Pengguna: Mengidentifikasi fitur utama yang dibutuhkan, seperti daftar mobil, detail mobil, login/registrasi, dan sistem transaksi.
2. Pembuatan Wireframe: Membuat sketsa awal tampilan halaman menggunakan tools seperti Canva untuk menyusun layout antarmuka.
3. Desain UI: Mengembangkan desain antarmuka menggunakan HTML, Tailwind CSS, dan Blade Laravel, berdasarkan wireframe yang telah dibuat.
4. Pengujian Desain (Usability Testing): Menguji tampilan dan alur penggunaan bersama beberapa pengguna untuk mendapatkan feedback.
5. Iterasi dan Perbaikan: Memperbaiki desain berdasarkan hasil pengujian, hingga antarmuka terasa optimal dan nyaman digunakan.

d. **Apakah hasil akhir perancangan yang Anda lakukan**

Hasil akhir dari perancangan yang kami lakukan adalah sebuah aplikasi jual beli mobil berbasis web yang fungsional, responsif, dan memiliki antarmuka pengguna yang intuitif. Aplikasi ini memungkinkan pengguna untuk dengan mudah mencari mobil, melihat detail, melakukan transaksi, dan mengelola akun mereka. Tampilan antarmuka nya dirancang dengan konsisten dan modern menggunakan HTML, Tailwind CSS, serta Blade Laravel, sehingga memberikan pengalaman pengguna yang nyaman dan mudah digunakan di berbagai perangkat. Selain itu, struktur sistem yang modular memudahkan pengembangan lanjutan serta proses pemeliharaan aplikasi ke depannya.

3. SYARAT Pengerjaan Modul

Setiap mahasiswa mengerjakan modul ini secara berkelompok berdasarkan Tim PBL nya. Namun setiap orang wajib memiliki peran tersendiri dalam penyusunan hasil praktikum minggu ini.

Implementasi

Pada tahap implementasi **system requirements specification** diberikan oleh **system analyst** kepada programmer untuk **dilakukan konstruksi (coding)** (Dennis. Alan, 2016). Hasil konstruksi berupa kode program. Kode program tersebut diberikan kepada **software tester** untuk dilakukan **pengujian sistem (unit, integration, system, user acceptance testing)** (Wahono et al., 2016). Setelah itu **change management analyst** akan memastikan dokumentasi yang memadai serta memberikan pelatihan kepada **user atau pengguna sistem** tentang sistem baru (Wahono, 2022).

1. Implementasi PBL

Lihat kembali dan lanjutkan tugas Anda dari minggu lalu.

Berdasarkan SKPL yang telah anda susun bersama tim minggu lalu, silahkan buat rancangan arsitekturnya.

*Lanjutkan pengerjaan modul praktikum minggu sebelumnya. Lengkapi dengan menyelesaikan modul praktikum minggu ini.

Dokumen SKPL/ SRS yang berisi informasi PBL Anda harusnya sudah dilengkapi pada paktikum minggu sebelumnya. SKPL dilengkapi dengan beberapa perancangan yaitu :

- Perancangan arsitektur sistem

Untuk membangun aplikasi jual beli mobil berbasis web, saya menggunakan arsitektur Model-View-Controller (MVC) yang disediakan oleh framework Laravel.

Arsitektur ini membantu memisahkan antara tampilan, logika aplikasi, dan pengelolaan data, sehingga sistem menjadi lebih terstruktur dan mudah dikembangkan.

Struktur ini juga selaras dengan konsep 3-tier architecture, yang membagi sistem menjadi 3 lapisan utama :

- Presentation Layer (View)
 - Menyajikan tampilan antarmuka kepada pengguna
 - Dibangun menggunakan Blade, HTML, Tailwind CSS
- Application Layer (Controller & Logic)
 - Mengatur alur data, validasi, proses bisnis
 - Ditangani oleh Controller, Middleware, Service di Laravel
- Data Layer (Model & Database)
 - Menyimpan dan mengambil data
 - Menggunakan Eloquent ORM dan database MySQL

Dengan pendekatan ini, pengembangan aplikasi menjadi modular, fleksibel, dan lebih mudah untuk pengujian serta pemeliharaan jangka panjang.

▪ Perancangan antarmuka

8. Fitur pencarian



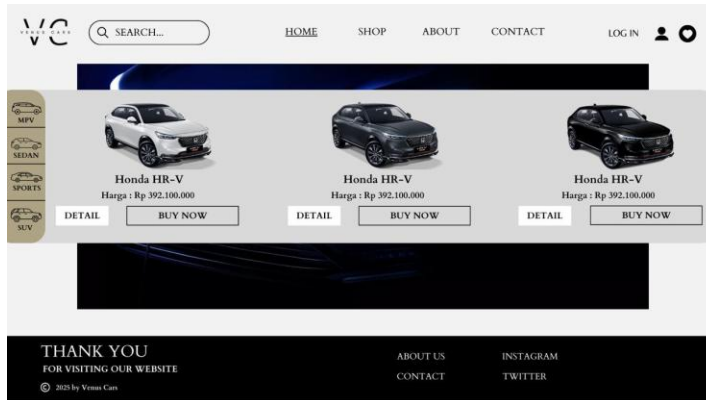
Fitur pencarian ini memungkinkan pengguna:

- Menemukan mobil berdasarkan kata kunci yang mereka masukkan.
- Menampilkan hasil dalam bentuk gambar dan nama mobil, sehingga memudahkan

pengguna memilih produk yang mereka inginkan.

- Disusun dengan desain modern dan user-friendly, membuat pencarian menjadi cepat dan mudah dipahami.

9. Fitur kategori



Fitur kategori ini memungkinkan pengguna:

- Menjelajahi mobil berdasarkan tipe kendaraan, seperti MPV, SEDAN, SPORTS, dan SUV, sehingga pengguna dapat langsung menemukan jenis mobil sesuai kebutuhan mereka.
- Mempercepat proses pencarian, karena pengguna tidak perlu menelusuri semua produk satu per satu cukup memilih kategori yang sesuai di bagian kiri halaman.
- Disusun dengan ikon dan teks yang intuitif, membuat navigasi menjadi lebih mudah, cepat, dan nyaman digunakan bahkan untuk pengguna baru.

10. Fitur show produk



Fitur Show Produk ini memungkinkan pengguna:

- Melihat daftar produk mobil yang tersedia secara lengkap, mulai dari tipe, gambar,

hingga harga awal setiap mobil, seperti Honda Mobilio, New Honda City, Honda Civic Type R, dan Honda HR-V.

- Menampilkan informasi produk secara visual dan informatif, sehingga pengguna dapat langsung membandingkan mobil berdasarkan tampilan dan harga.
- Disertai tombol “Add to Wishlist”, yang memudahkan pengguna menyimpan produk favorit mereka untuk dibuka kembali nanti.
- Didesain dengan tata letak yang rapi dan elegan, memudahkan pengguna menjelajahi berbagai pilihan tanpa merasa bingung atau kewalahan.

11. Fitur CRUD

VC

VEVO TAXI

Dashboard

Daftar Produk

Daftar Pesanan

Seller Center

Logout

Daftar Produk

TAMBAH PRODUK

NO	PRODUK	PENJUALAN	HARGA	STOK	AKSI
1	 Honda Civic Type R 1 unit	Rp 1.455.500.000	2	<div><div></div></div> <div><div></div></div>	
2	 Honda Civic Type R 1 unit	Rp 1.455.500.000	2	<div><div></div></div> <div><div></div></div>	
3	 Honda Civic Type R 1 unit	Rp 1.455.500.000	1	<div><div></div></div> <div><div></div></div>	
4	 Honda HR-V 1 unit	Rp 392.100.000	1	<div><div></div></div> <div><div></div></div>	
5	 Honda HR-V 1 unit	Rp 392.100.000	3	<div><div></div></div> <div><div></div></div>	
6	 Honda HR-V 1 unit	Rp 392.100.000	4	<div><div></div></div> <div><div></div></div>	

Fitur ini memungkinkan penjual untuk mengelola daftar produk secara menyeluruh, mencakup empat aksi utama CRUD:

Create, Read, Update, dan Delete.

- Create (Tambah Produk)

Penjual dapat menambahkan produk baru ke dalam daftar menggunakan tombol "TAMBAH PRODUK".

- Read (Lihat Daftar Produk)

Menampilkan semua produk yang dijual, lengkap dengan informasi:

Gambar produk

Nama produk

Jumlah penjualan

Harga

Stok tersedia



- Update (Edit Produk)

Tombol "EDIT" memungkinkan penjual untuk mengubah detail produk seperti nama, harga, atau stok.

- Delete (Hapus Produk)

Tombol "DELETE" digunakan untuk menghapus produk dari daftar.

12. Fitur menampilkan semua pesanan

No	Produk	Total Pesanan	Status	Aksi
1	 Honda Mobilio	Rp 243,300,000	Dikemas	Print
2	 New Honda City	Rp 402,000,000	Dikemas	Print

Fitur ini menampilkan semua pesanan yang masuk ke penjual, sehingga penjual bisa memantau, memproses, dan mengelola setiap transaksi secara efisien.

Fungsi dan Komponen Utama:

- Daftar pesanan ditampilkan dalam bentuk tabel yang memuat informasi penting:

Nomor urut

Gambar dan nama produk

Total harga pesanan

Status pesanan (misalnya: Dikemas)

Tombol aksi (contohnya: Print)

- Filter Status Pesanan

Terdapat tombol filter di bagian atas untuk memudahkan penjual menyaring pesanan berdasarkan status:

Semua

Konfirmasi

Dikemas

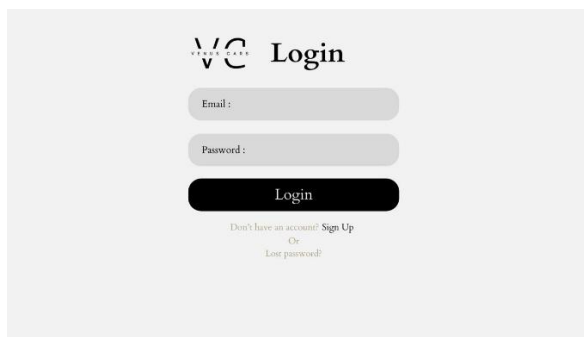
Dikirim

Selesai

- Tombol "Print"

Memungkinkan penjual mencetak data pesanan untuk keperluan pengemasan atau pengiriman.

13. Fitur Login



Fitur ini menampilkan halaman login dari sistem Venus Cars, yang digunakan untuk mengautentikasi pengguna sebelum mereka bisa mengakses fitur-fitur utama dalam aplikasi.

- Logo & Judul

Terdapat logo VC (Venus Cars) dan teks "Login" sebagai penanda halaman login.

Input Form:

- Email:

Field untuk memasukkan alamat email pengguna.

- Password:

Field untuk memasukkan kata sandi yang terkait dengan akun.

- Tombol Login:

Tombol besar berwarna hitam bertuliskan "Login" digunakan untuk mengirim data email dan password ke sistem untuk proses autentikasi.

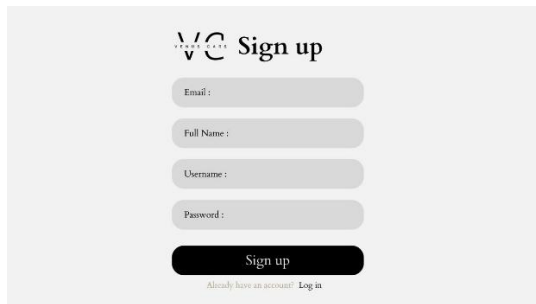
- Navigasi Akun:

Tautan ke Sign Up bagi pengguna baru yang belum memiliki akun.

- Tautan Lost password?

untuk membantu pengguna memulihkan akun mereka jika lupa password.

14. Fitur register



Fitur ini menampilkan fitur registrasi (Sign Up) pada aplikasi Venus Cars. Di halaman ini, pengguna yang belum memiliki akun bisa mendaftar dengan mengisi beberapa data yang diperlukan, yaitu:

- Email – untuk keperluan verifikasi dan login
- Full Name – nama lengkap pengguna
- Username – sebagai identitas unik pengguna dalam system
- Password – untuk keamanan akun saat login

Setelah semua data diisi, pengguna bisa klik tombol Sign Up untuk membuat akun baru. Setelah berhasil registrasi, pengguna bisa langsung login dan mengakses fitur-fitur yang tersedia, seperti menambah produk, melihat pesanan, dan lainnya.

- Buat spesifikasi kebutuhan *hardware* dan *software* aplikasi Anda

3. Kebutuhan Perangkat Keras (Hardware)

a. Untuk Pengembang :

Komponen	Spesifikasi Minimum
Prosesor	Intel Core i5
RAM	20 GB
Hard Disk	512 GB SSD

Monitor	Resolusi 1920 x 1080
Perangkat Tambahan	Mouse

b. Untuk Pengguna :

Komponen	Spesifikasi Minimum
Perangkat	Laptop / PC / Smartphone
Browser	Chrome, Firefox, Microsoft Edge, Safari
RAM	2 GB
Koneksi Internet	Stabil (minimal 1 Mbps)

4. Kebutuhan Perangkat Lunak (Software)

c. Untuk Pengembang :

Software	Keterangan
OS	Windows 11
Web Server	Laragon (Apache, MySQL, PHP)
Framework	Laravel (PHP Framework)
Editor Code	Visual Studio Code
Browser	Google Chrome, Microsoft Edge
DBMS	MySQL / phpMyAdmin

b. Untuk Pengguna :

Software	Keterangan
Web Browser	Chrome / Firefox / Microsoft Edge / Safari
OS	Android / Windows / iOS

▪ **Perancangan Data**

Fitur utama aplikasi meliputi :

- Pengelolaan data mobil (jual/beli)
- Pengguna (pembeli & penjual)
- Transaksi
- Pembayaran
- Riwayat pesanan

-
- Kategori mobil, merek dan warna mobil.

1. Entitas Utama (Tabel Database)

1.1. Users (Pengguna)

Kolom	Tipe Data	Keterangan
id	bigint (PK)	ID pengguna
name	string	Nama pengguna
email	string	Email unik
password	string	Hash password
phone	string	Nomor telepon
role	enum	'buyer', 'seller', 'admin'
created_at	timestamp	
updated_at	timestamp	

1.2. Cars (Mobil)

Kolom	Tipe Data	Keterangan
id	bigint (PK)	ID mobil
user_id	bigint (FK)	ID pengguna (penjual)
brand_id	bigint (FK)	ID merek mobil
model	string	Nama mobil (contoh: Avanza G)
color	string	Warna mobil (contoh: Hitam)
specification	text	Spesifikasi teknis
description	text	Deskripsi umum atau kondisi mobil
price	decimal	Harga jual
status	enum	'available', 'sold'
created_at	timestamp	

Kolom	Tipe Data	Keterangan
updated_at	timestamp	

1.3. Brands (Merek Mobil)

Kolom	Tipe Data	Keterangan
id	bigint	PK
name	string	Nama merek

1.4. Categories (Kategori Mobil)

Kolom	Tipe Data	Keterangan
id	bigint	PK
name	string	Contoh: SUV, Sedan

1.5. Transactions (Transaksi)

Kolom	Tipe Data	Keterangan
id	bigint (PK)	
car_id	bigint (FK)	Mobil yang dibeli
buyer_id	bigint (FK)	Pembeli
price	decimal	Harga transaksi
status	enum	'pending', 'paid', 'cancelled'
created_at	timestamp	

1.6. Payments (Pembayaran)

Kolom	Tipe Data	Keterangan
id	bigint	
transaction_id	bigint (FK)	
method	string	Transfer, COD, dll

Kolom	Tipe Data	Keterangan
paid_at	timestamp	
status	enum	'success', 'failed'

2. Relasi antar tabel

- Users memiliki banyak Cars dan Transactions
- Cars dimiliki oleh satu User
- Cars masuk dalam satu Brand dan satu Category
- Transactions berhubungan dengan Cars, Users (buyers), dan satu Payment
- Payments terkait satu Transaction
- Reviews dari Users terhadap Cars

▪ Perancangan Algoritma

Algoritma yang dirancang akan mencakup alur utama dari sistem seperti:

- Registrasi & login
- Menambah mobil
- Menampilkan daftar mobil
- Membeli mobil
- Proses transaksi

1. Algoritma Registrasi Pengguna

Input: nama, email, password, no telepon, role

Output: akun baru dibuat / error

```
function registerUser(name, email, password, phone, role):
    if email sudah terdaftar:
        return "Email sudah digunakan"
    hashPassword = hash(password)
    simpan data user ke database
    return "Registrasi berhasil"
```

2. Algoritma Login

Input: email, password

Output: sesi login aktif / gagal

```
function login(email, password):  
    cari user berdasarkan email  
    if tidak ditemukan:  
        return "Email tidak ditemukan"  
    if password tidak cocok:  
        return "Password salah"  
    buat sesi login  
    return "Login berhasil"
```

3. Algoritma Menambahkan Mobil oleh Penjual

Input: data mobil (model, tahun, harga, deskripsi, dll)

Output: mobil berhasil ditambahkan

```
function addCar(user_id, car_data):  
    if user role != 'seller':  
        return "Akses ditolak"  
    simpan car_data ke database dengan user_id sebagai owner  
    return "Mobil berhasil ditambahkan"
```

4. Algoritma Menampilkan Daftar Mobil

Input: filter (merek, kategori, harga)

Output: daftar mobil

```
function getCarList(filters):  
    query = "SELECT * FROM cars WHERE status = 'available'"  
    if filter merek:  
        tambahkan filter brand_id  
    if filter harga:  
        tambahkan range harga  
    if filter kategori:  
        tambahkan filter category_id  
    return hasil query
```

5. Algoritma Pembelian Mobil

Input: id mobil, id pembeli

Output: transaksi baru dibuat

```
function purchaseCar(car_id, buyer_id):  
    cek mobil tersedia  
    if mobil.status != 'available':  
        return "Mobil tidak tersedia"  
    buat transaksi dengan status 'pending'  
    ubah status mobil jadi 'sold'  
    return "Transaksi berhasil dibuat"
```

6. Algoritma Pembayaran

Input: id transaksi, metode pembayaran

Output: status pembayaran

```
function payTransaction(transaction_id, method):  
    cek transaksi valid dan pending  
    simpan data pembayaran  
    update status transaksi jadi 'paid'  
    return "Pembayaran berhasil"
```

2. CHECKPOINT

Jawab pertanyaan berikut :

a. Pilih salah satu dari Best Practices dalam Rekayasa Perangkat Lunak sebagai berikut (Setiap Kelompok harus menjawab minimal 5 pertanyaan, Dalam 1 kelompok PBL Tidak boleh menjawab pertanyaan yang sama) :

i. Ada anggapan berikut:

1. Makin banyak orang makin bagus. Jadi jika desain kita tidak bagus, kita bisa menambahkan programmer ke tim dan mengejar ketinggalan kita.

-
2. Jika project manager merasa bahwa tahap pemrograman ketinggalan jadwal, maka tambahkan lagi jumlah programmer sehingga bisa mengejar keterlambatan jadwal.

Dalam rekayasa perangkat lunak, kedua anggapan tersebut disebut sebagai konsep "Mongolian horde".

- ii. Apakah menambah programmer di tahap implementasi akan dapat menyelesaikan masalah? Jawab: Tidak selalu. Menambah programmer di tahap implementasi bisa menyebabkan penundaan alih-alih mempercepat proyek.

Programmer baru membutuhkan waktu untuk memahami sistem yang sudah berjalan, dan hal ini bisa mengganggu produktivitas tim yang sudah ada. Prinsip ini dikenal dalam *rekayasa perangkat lunak* sebagai Brooks' Law: "*adding manpower to a late software project makes it later.*"

- iii. Kemampuan utama yang dibutuhkan untuk membangun sebuah aplikasi/ sistem yang userfriendly and usable ada kemampuan menulis kode (coding). Jawab : Tidak benar sepenuhnya. Coding hanyalah salah satu bagian dari proses pengembangan sistem. Untuk menghasilkan aplikasi yang user-friendly dan usable, diperlukan kemampuan dalam memahami kebutuhan pengguna (user research), desain antarmuka (UI/UX), serta pengujian usability. Tanpa pemahaman tentang pengguna akhir, aplikasi bisa saja tidak efektif meskipun kodenya bagus.

- iv. Ekperimen dan pengujian aplikasi akan berhasil pada percobaan pertama. Jawab: Umumnya tidak. Dalam praktiknya, pengujian pertama jarang langsung berhasil. Diperlukan proses iteratif, perbaikan bug, dan validasi terus-menerus agar sistem berfungsi sesuai harapan. Oleh karena itu, fase testing dan debugging sangat penting dalam siklus hidup pengembangan perangkat lunak.

-
- v. Tidak ada yang tidak mungkin ditangani oleh sistem.

Jawab : Anggapan ini terlalu optimis. Sistem memiliki keterbatasan, baik dari sisi teknologi, biaya, waktu pengembangan, maupun kompleksitas masalah. Tidak semua kebutuhan dapat diotomatisasi. Oleh karena itu, penting untuk menetapkan batasan dan prioritas sejak awal agar sistem tetap realistis dan dapat diimplementasikan dengan baik.

vi. Programmer berusaha melakukan implementasi kode secepat mungkin.

Jawab : Kecepatan memang penting, tetapi **kualitas jauh lebih penting**. Menulis kode terburu-buru dapat menyebabkan **technical debt** (utang teknis), sulit dipelihara, dan rawan bug. Pendekatan yang disarankan adalah pengembangan secara bertahap, dengan memperhatikan struktur, dokumentasi, dan pengujian yang memadai.

vii. Bahasa pemrograman menentukan kualitas aplikasi atau sistem yang akan dibuat.

Jawab : Bahasa pemrograman hanyalah alat. Kualitas aplikasi lebih ditentukan oleh arsitektur sistem, praktik pengembangan, desain antarmuka, serta kemampuan tim pengembang. Bahasa yang digunakan harus dipilih berdasarkan kebutuhan proyek, bukan semata-mata karena "bahasa tertentu dianggap bagus"

viii. Bagaimana pendapat Anda mengenai anggapan tersebut?

Mengapa ya atau mengapa tidak? Jawab : Sebagian besar anggapan tersebut bersifat mitos atau penyederhanaan berlebihan. Dalam rekayasa perangkat lunak, pengembangan sistem adalah proses kompleks yang melibatkan banyak aspek, seperti manajemen proyek, analisis kebutuhan, perancangan sistem, pengujian, dan pemeliharaan. Menyederhanakan proses ini hanya ke dalam jumlah programmer atau kecepatan coding adalah keliru dan berisiko besar.

- b. Jelaskan Bagaimana usecase atau diagram perancangan lain, classdiagram, sequence diagram, activity diagram dapat membantu mengidentifikasi pembuatan design perangkat lunak (arsitektur, data, algoritma, antarmuka)

Jawab : Diagram perancangan seperti use case, class diagram, sequence diagram, dan activity diagram sangat membantu untuk memahami dan merancang sistem sebelum mulai coding:

- Use Case Diagram: Menjelaskan fungsi-fungsi utama yang dibutuhkan pengguna dan bagaimana interaksinya dengan sistem.
- Class Diagram: Menunjukkan struktur data dan hubungan antar objek, membantu dalam desain database dan logika program.
- Sequence Diagram: Menjelaskan urutan alur proses atau komunikasi antar bagian sistem.
- Activity Diagram: Mempermudah memahami alur kerja atau proses bisnis dari suatu fitur.

Dengan semua diagram ini, tim bisa memahami sistem lebih jelas, meminimalkan kesalahan, dan membagi tugas dengan lebih terstruktur.

c. Apa kesulitan yang Anda hadapi dalam mengerjakan praktikum minggu ini?

Jawab : Kesulitan yang kami hadapi dalam praktikum minggu ini adalah pada saat menyusun diagram perancangan yang lebih kompleks dan memastikan konsistensinya antara satu diagram dengan yang lainnya. Terutama dalam menghubungkan antara use case, class diagram, dan sequence diagram agar sesuai dengan fungsionalitas sistem yang sedang dikembangkan. Selain itu, tantangan lainnya adalah memahami bagaimana setiap diagram berkontribusi pada desain sistem secara keseluruhan dan memastikan bahwa alur proses bisnis atau interaksi antar komponen sudah terstruktur dengan baik. Hal ini membutuhkan pemahaman mendalam tentang bagaimana sistem bekerja secara menyeluruh, yang bisa menjadi cukup rumit ketika diterjemahkan ke dalam diagram yang jelas dan mudah dipahami.

3. SYARAT Pengerjaan Modul

Setiap mahasiswa mengerjakan modul ini secara berkelompok berdasarkan Tim PBL nya. Namun setiap orang wajib memiliki peran tersendiri dalam penyusunan hasil praktikum minggu ini.