

BAB II

TEORI PENUNJANG

Pada bagian ini akan dijelaskan mengenai teori-teori yang akan menunjang dalam pengerjaan pembuatan Website Manajemen Bisnis Penjualan Ban PT. Goldfinger Wheels Indonesia dengan menggunakan Framework Laravel sebagai hasil dari tugas akhir. Teori-teori penunjang ini akan dijelaskan dengan rinci secara individual dalam bentuk poin-poin dalam bab ini.

2.1 Teori Penjualan Ban

Perusahaan penjualan ban adalah entitas bisnis yang secara khusus berfokus pada penjualan dan distribusi ban kendaraan kepada konsumen. Mereka berperan sebagai perantara antara produsen ban dan pengguna akhir, seperti pemilik kendaraan pribadi, perusahaan transportasi, atau pemilik armada kendaraan komersial. Dalam menjalankan peran ini, perusahaan penjualan ban harus berusaha memenuhi kebutuhan dan preferensi konsumen dengan menyediakan berbagai jenis ban, termasuk ban untuk kendaraan penumpang, truk, bus, dan sepeda motor.

Ban untuk kendaraan selalu memiliki kode yang tertulis pada sisi samping ban. Dari kode tersebut tertera informasi mengenai ban tersebut. Cara membaca kode ban untuk kendaraan truk berbeda dengan ban untuk kendaraan selain truk. Kode ban untuk kendaraan truk akan dimulai dari ukuran lebar ban dalam satuan inci, lalu tipe ban, lalu ukuran diameter dari rim atau velg ban, lalu *ply rating* atau nilai ketebalan ban yang bila semakin tinggi maka semakin tebal, lalu beban yang dapat ditopang oleh ban, dimana pada kode bagian ini akan menyertakan informasi beban maksimal untuk ban tunggal diikuti dengan ban ganda. Dan yang terakhir adalah kode indeks kecepatan ban, setiap manufaktur ban memiliki kode dan kecepatan ban masing-masing.

Seperti contoh pada ban dengan kode 7.50R16 14PR 123/121 L. Dari kode tersebut dapat kita simpulkan bahwa ban tersebut memiliki lebar ban 7.5 inci, R artinya ban tersebut bertipe *radial*, angka 16 menunjukkan diameter rim atau velg

ban, kode 14PR menunjukkan *ply rating* dari ban adalah 14. Pada kode angka 123/121, artinya indeks beban tiap ban pada truk ban tunggal adalah 123, sedangkan indeks beban tiap ban pada truk ban ganda adalah 121, indeks beban pada kendaraan ban ganda lebih kecil agar kendaraan dapat beroperasi ketika salah satu ban pecah. Dan yang terakhir adalah kode L, yang menunjukkan indeks kecepatan ban. Nilai kecepatan ban harus merujuk pada tabel indeks kecepatan manufaktur ban.

Dalam melakukan proses jual beli ban terdapat beberapa hal yang menjadi faktor penting dan sorotan yang menjadi pertimbangan saat proses jual beli akan terjadi. Mulai dari ukuran dan ketebalan ban yang harus sesuai dengan kendaraan calon pelanggan, kode kecepatan pada ban yang menentukan kecepatan maksimal ban saat berkendara, pola dari tapak ban yang akan disesuaikan dengan medan yang akan dilalui, dan kode produksi ban yang menunjukkan kapan tanggal ban dibuat, sebab ban dengan usia produksi lebih muda cenderung punya kualitas dan ketahanan yang lebih baik daripada ban dengan usia produksi lebih tua. Faktor diatas sering ditentukan preferensi masing-masing calon pelanggan sebab setiap pelanggan punya kebutuhannya masing-masing.

Manajemen penjualan ban adalah rangkaian kegiatan yang dilakukan untuk memastikan proses bisnis penjualan ban yang optimal. Aspek-aspek dalam manajemen penjualan ban meliputi pemilihan jenis, merk, dan kualitas ban yang tepat, teknik penyimpanan dan pemasangan ban yang tepat. Selain itu, perusahaan juga harus mempertimbangkan strategi pemasaran yang efektif untuk menarik pelanggan dan membangun hubungan jangka panjang dengan mereka. Ini melibatkan periklanan, promosi penjualan, dan layanan pelanggan yang baik.

2.2 Vulkanisir

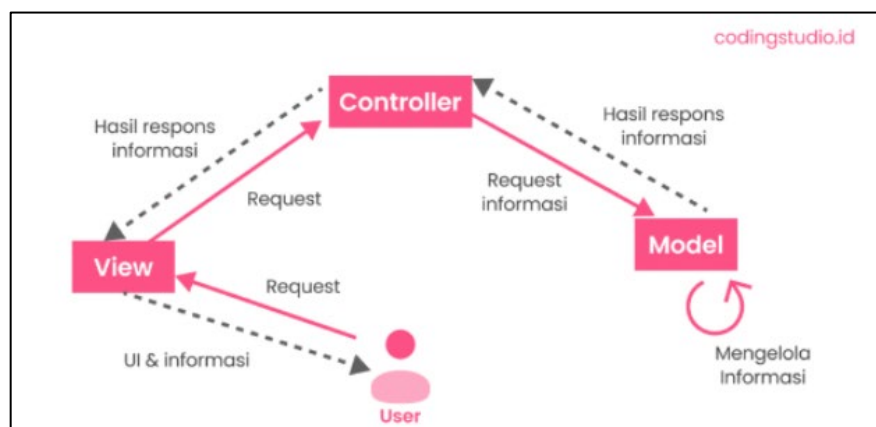
Vulkanisir merupakan teknik yang digunakan untuk memperbaiki ban yang sudah aus dengan menerapkan lapisan karet baru pada permukaan luar ban. Proses ini melibatkan beberapa tahapan penting, termasuk pembersihan menyeluruh ban untuk menghilangkan kotoran dan residu, pemotongan bagian yang rusak untuk memastikan permukaan yang bersih dan rata, dan penerapan lapisan karet baru yang disiapkan secara khusus agar sesuai dengan spesifikasi ban yang sedang diperbaiki.

Proses vulkanisir juga melibatkan penggunaan panas dan tekanan. Lapisan karet baru ditempatkan pada permukaan ban yang sudah disiapkan, kemudian dipanaskan dan ditekan agar menempel dengan kuat. Proses ini menciptakan ikatan yang kuat antara lapisan karet baru dan ban yang sudah ada, sehingga menciptakan permukaan yang halus dan aman.

Umumnya, vulkanisir diterapkan pada ban kendaraan besar seperti truk dan bus guna memperpanjang umur pakai ban serta mengurangi biaya penggantian ban secara keseluruhan. Selain itu, metode vulkanisir juga memberikan manfaat ekonomis yang signifikan bagi perusahaan transportasi dengan mengurangi pengeluaran terkait pemeliharaan dan penggantian ban. Dengan menggunakan metode vulkanisir, ban yang sudah aus dapat dipulihkan sehingga memiliki performa yang hampir serupa dengan ban baru, sambil juga membantu mengurangi dampak lingkungan melalui pengurangan limbah ban yang dibuang.

2.3 Model View Controller (MVC)

Arsitektur model-view-controller (MVC) merupakan konsep yang menggunakan tiga pembagian utama, yaitu model, view, dan controller. Model berfungsi sebagai pengatur hal-hal yang berkaitan dengan basis data dan logika bisnis aplikasi. Model bertanggung jawab untuk mengambil, menyimpan, dan memanipulasi data yang diperlukan oleh aplikasi. Dengan memisahkan model dari view dan controller, pengembangan aplikasi menjadi lebih terstruktur dan memungkinkan pemeliharaan kode yang lebih mudah.



Gambar 2.1 **Sistem Pada MVC**

View, di sisi lain, berperan sebagai pengatur tampilan halaman atau antarmuka pengguna. View menampilkan data yang dihasilkan oleh model kepada pengguna dengan cara yang estetik dan mudah dipahami. View juga memungkinkan pengguna untuk berinteraksi dengan aplikasi dan mengirimkan input kembali ke controller. Dengan demikian, view berfungsi sebagai jembatan antara aplikasi dan pengguna akhir.

Controller adalah penghubung antara model dan view, mengimplementasikan logika-logika penulisan kode yang mengatur aliran data dan interaksi antara model dan view. Controller menerima input dari pengguna melalui view dan meneruskan permintaan ke model untuk diproses. Setelah model mengolah data dan menghasilkan output, controller kemudian mengirimkan hasil tersebut ke view untuk ditampilkan kepada pengguna. Controller juga mengelola tindakan pengguna, seperti mengklik tombol atau mengisi formulir, dan mengarahkan respons yang sesuai berdasarkan logika bisnis aplikasi.

Dengan menerapkan arsitektur MVC, pengembang dapat memisahkan tanggung jawab dan fokus pada aspek yang berbeda dari aplikasi secara mandiri. Ini membantu meningkatkan keterbacaan, efisiensi, dan skalabilitas kode. Selain itu, arsitektur MVC juga memfasilitasi pengembangan tim yang lebih kolaboratif, di mana anggota tim dapat bekerja pada komponen yang berbeda secara paralel, tanpa mengganggu bagian lain dari aplikasi. Secara keseluruhan, konsep ini telah menjadi pondasi penting dalam pengembangan aplikasi web modern.

2.4 Laravel

Laravel merupakan salah satu framework PHP yang dikembangkan oleh Taylor Otwell pada tahun 2011. Framework ini dirilis di bawah lisensi MIT dan menggunakan konsep MVC (model-view-controller). Laravel bersifat open source dan dikembangkan dengan tujuan untuk memudahkan pengembangan aplikasi web dengan menyediakan lingkungan dan alat yang kuat serta efisien bagi para pengembang, sehingga mengurangi kerumitan dalam mengembangkan aplikasi

web dan memungkinkan para pengembang untuk lebih fokus pada logika bisnis dan fitur aplikasi daripada pada tugas-tugas teknis yang repetitif.

Selain menerapkan konsep MVC, Laravel juga menawarkan fitur-fitur canggih lainnya yang membantu pengembang dalam mengembangkan aplikasi web dengan lebih cepat dan efisien. Salah satunya adalah sistem routing yang fleksibel, yang memungkinkan pengembang untuk dengan mudah menentukan rute-rute yang sesuai dengan kebutuhan aplikasi mereka. Framework ini juga menyediakan fitur Eloquent ORM (Object-Relational Mapping) yang memudahkan interaksi dengan basis data, sehingga pengembang dapat bekerja dengan data dalam aplikasi mereka dengan lebih intuitif dan efisien.

Komunitas pengguna Laravel yang besar dan aktif juga berperan dalam menjadikan framework ini pilihan populer di kalangan pengembang web. Dukungan komunitas ini memastikan ketersediaan berbagai tutorial, paket ekstensi, dan sumber daya lainnya yang dapat membantu pengembang dalam memanfaatkan fitur-fitur Laravel dengan maksimal. Dengan seluruh kelebihan tersebut, Laravel menjadi framework pilihan bagi banyak pengembang dalam membangun aplikasi web modern yang efisien, aman, dan terukur.

2.5 MySQL

MySQL adalah sebuah sistem manajemen basis data relasional (RDBMS) yang berbasis open-source. Pertama kali dirilis pada tahun 1995 oleh perusahaan Swedia, MySQL AB, yang dipimpin oleh Michael "Monty" Widenius, David Axmark, dan Allan Larsson. Nama "MySQL" sendiri diambil dari nama putri Monty yang bernama My. MySQL awalnya dikembangkan sebagai alternatif open-source yang lebih terjangkau untuk sistem manajemen basis data komersial yang mahal pada saat itu.

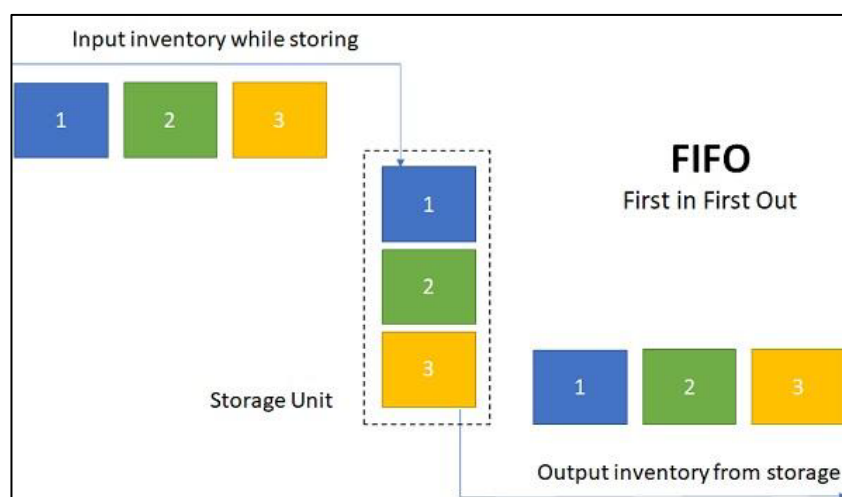
Dengan struktur client-server, MySQL memungkinkan banyak pengguna untuk mengakses dan mengelola data secara bersamaan melalui jaringan. Ini menjadikannya pilihan yang populer untuk berbagai aplikasi, mulai dari situs web kecil hingga aplikasi korporat yang besar.

MySQL menggunakan bahasa kueri SQL (Structured Query Language) untuk mengelola data. Pengguna dapat menggunakan perintah SQL untuk melakukan berbagai operasi, termasuk pencarian, penyisipan, pembaruan, dan penghapusan data dalam basis data.

Hingga saat ini, MySQL terus menjadi salah satu RDBMS yang paling banyak digunakan di dunia. Pengembangannya terus berlanjut dengan rilis-versi baru yang menyertakan peningkatan kinerja, fitur baru, dan perbaikan bug. MySQL memiliki pengaruh yang besar dalam dunia pengembangan perangkat lunak dan terus menjadi pilihan utama bagi banyak pengembang dan organisasi di seluruh dunia.

2.6 FIFO (First In First Out)

FIFO, singkatan dari First In First Out, adalah sebuah metode atau prinsip pengelolaan stok yang mengacu pada urutan keluar barang atau item dari gudang atau persediaan. Dalam metode FIFO, barang yang pertama kali masuk ke persediaan (masuk pertama) akan menjadi barang yang pertama kali diambil atau dijual (keluar pertama). Hal ini membantu memastikan bahwa stok yang lebih lama keluar terlebih dahulu, mengurangi risiko kedaluwarsa atau kerusakan barang.



Gambar 2.2
Sistem FIFO

Dalam konteks manajemen persediaan, FIFO digunakan untuk mengatur bagaimana barang disimpan dan diambil dari gudang atau stok. Misalnya, jika Anda memiliki gudang penyimpanan untuk produk-produk tertentu, FIFO akan memastikan bahwa produk-produk yang masuk terlebih dahulu ke dalam stok juga akan dikeluarkan terlebih dahulu untuk dijual atau digunakan. Dengan menerapkan prinsip FIFO, risiko barang menjadi kadaluwarsa atau usang dapat diminimalkan karena barang yang sudah ada dalam stok lebih lama akan dijual atau digunakan lebih dulu.

FIFO juga berguna untuk produk dengan harga yang fluktuatif, seperti komoditas, karena menjual barang yang lebih tua sebelum harga berubah dapat menjaga margin keuntungan yang lebih stabil. Selain itu, penerapan FIFO mendukung pencatatan keuangan yang akurat karena mencerminkan biaya aktual untuk membeli barang. Meskipun metode ini memiliki banyak manfaat, penggunaannya harus disesuaikan dengan jenis bisnis dan karakteristik produk yang dikelola.

Secara keseluruhan, FIFO merupakan konsep yang sederhana namun penting dalam mengatur urutan pemrosesan, pengeluaran, atau pelayanan barang, layanan, atau informasi. Dengan menerapkan prinsip FIFO, sistem dapat menjaga keteraturan, keadilan, dan efisiensi dalam pengelolaan barang atau layanan, serta memberikan pengalaman yang lebih baik bagi pengguna atau pelanggan.

2.7 Waterfall

Metodologi waterfall adalah salah satu model pengembangan perangkat lunak yang telah ada sejak lama. Dalam pendekatan ini, pengembangan perangkat lunak mengikuti urutan linear dan terstruktur yang melibatkan serangkaian tahapan yang berurutan dan tidak dapat diputar mundur. Dengan kata lain, setiap tahap dalam proses pengembangan harus diselesaikan sepenuhnya sebelum memulai tahap berikutnya, mirip dengan air yang mengalir dari satu tahap ke tahap berikutnya seperti air terjun (waterfall).

Proses pengembangan dalam metodologi waterfall dimulai dengan tahap analisis dan perencanaan. Tahap ini sangat penting karena di sinilah kebutuhan bisnis dan persyaratan pengguna ditentukan dengan jelas. Tim pengembangan

mengidentifikasi tujuan proyek, menetapkan lingkup, dan merencanakan strategi implementasi. Selanjutnya, hasil dari tahap analisis dan perencanaan ini didokumentasikan secara rinci untuk menjadi panduan bagi tahapan-tahapan selanjutnya.

Setelah tahap analisis dan perencanaan selesai, proses berlanjut ke tahap desain. Di sini, arsitektur perangkat lunak direncanakan secara rinci. Ini termasuk merancang struktur sistem, antarmuka pengguna, serta menentukan algoritma dan teknologi yang akan digunakan. Dokumentasi desain yang lengkap diperlukan untuk memastikan bahwa semua anggota tim memiliki pemahaman yang sama tentang bagaimana sistem akan dibangun.

Selanjutnya adalah tahap implementasi, di mana kode program sebenarnya ditulis berdasarkan desain yang telah disepakati. Tim pengembangan bekerja untuk mengubah konsep desain menjadi produk perangkat lunak yang berfungsi. Dalam tahap ini, perhatian besar diberikan pada kualitas kode, dokumentasi, dan kompatibilitas dengan lingkungan yang dituju.

Setelah implementasi selesai, tahap pengujian dilakukan untuk memverifikasi bahwa perangkat lunak berfungsi seperti yang diharapkan dan memenuhi semua persyaratan yang telah ditetapkan. Ini termasuk pengujian fungsional, pengujian performa, dan pengujian integrasi untuk memastikan bahwa setiap komponen sistem berinteraksi dengan baik.

Terakhir, setelah semua tahap selesai dan perangkat lunak telah lolos pengujian, dilakukan tahap pemeliharaan. Tahap ini melibatkan pemeliharaan rutin, perbaikan bug, dan penyempurnaan fitur sesuai dengan umpan balik pengguna. Pemeliharaan ini penting untuk memastikan bahwa perangkat lunak tetap berjalan dengan baik dan dapat mengakomodasi perubahan kebutuhan bisnis yang terus berkembang.

Pendekatan waterfall menekankan pentingnya perencanaan yang teliti dan dokumentasi yang kuat untuk memastikan setiap tahap berjalan lancar dan hasilnya dapat diprediksi dengan baik.