|  |
| --- |
| KERJA PRAKTEK  di  PT. Adiputro Wirasejati |
| LOGO ISTTS TRANS PUTIH |
|  |
| Oleh:  Felis Gosal  219310436  Charles Ciputra Wibisono  219310433 |
| PROGRAM SARJANA  PROGRAM STUDI TEKNIK INFORMATIKA Prog. Prof.  FAKULTAS SAINS DAN TEKNOLOGI  INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA  SURABAYA  2022 |

KERJA PRAKTEK

di

PT. Adiputro Wirasejati

Menyetujui

Pembimbing Lapangan

(Elfan Hermanto)

SURABAYA

OKTOBER 2022

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Sutjipto Susanto

Fakultas/ Prodi : Sains dan Teknologi/ Teknik Elektro

NRP : 197102187

dengan ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir/Tesis dengan judul:

TEMPLATE PENULISAN JUDUL

TUGAS AKHIR/TESIS

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 01-01-2019

Yang Membuat Pernyataan,

Sutjipto Susanto

197102187

ABSTRAK

PT X adalah sebuah perusahaan karoseri bus asal Malang, Jawa Timur. PT X didirikan oleh tiga bersaudara yang bernama Andreas Jethrokusumo, Jesse Jethrokusumo dan David Jethrokusumo pada tahun 1975 di Kota Malang, Jawa Timur yang hingga kini masih dijadikan kantor pusatnya. PT X telah membuat banyak bodi bus yang digunakan oleh banyak perusahaan-perusahaan otobus ternama di Indonesia seperti Lorena, Pahala Kencana, ALS, dan lainnya.

PT X bergerak pada bidang jasa yang membuat bodi bus, elf, dan kendaraan lainnya. Proses utama dalam pengerjaan karoseri tersebut adalah PT X menunggu mesin dan chasis dari mobil pesanan datang, lalu mesin tersebut akan menuju pusat produksi. Pada proses pengerjaan karoseri, PT X menggunakan Surat Perintah Kerja (SPK) untuk memindahkan seluruh kebutuhan bahan dari gudang ke bagian produksi. Saat ini PT X masih menggunakan sistem manual yaitu SPK dicetak terlebih dahulu lalu ditempelkan pada mesin mobil. Lalu staff produksi akan melihat item apa saja yang dibutuhkan, setelah itu staff produksi akan melakukan permintaan ke staff gudang. Barulah setelah itu, komponen - komponen yang dibutuhkan untuk membuat karoseri akan diserahkan ke bagian produksi. Sistem kerja yang seperti ini dirasa belum maksimal, karena komponen - komponen yang diperlukan akan datang apabila mobil yang dipesan sudah datang. Sistem pembuatan Surat Pengantar Kerja juga hanya masih menggunakan macro excel yang tidak terintegrasi dengan database. Maka dari itu, PT X ingin memperbaiki sistem produksinya, sehingga sebelum mobil datang, komponen - komponen yang diperlukan untuk membuat karoseri sudah siap di bagian produksi. Maka dari itu PT X menginginkan sebuah website dimana saat adrnin memasukkan nomor Surat Perintah Kerja, maka sistem akan mengirirnkan list item yang diperlukan ke bagian gudang, dan bagian gudang akan mengirirnkan item - item yang dibutuhkan ke bagian produksi

KATA PENGANTAR

Segala puji dan syukur dipanjatkan kepada Tuhan Yang Maha Esa atas kemurahan dan kemudahan yang telah diberikanNya, sehingga penyusunan buku yang berjudul inidapat diselesaikan dengan baik dan tepat waktu. Tanpa rahmat dan berkat-Nya, buku ini tidak dapat disusun dengan baik. Penulis berharap buku ini dapat bermanfaat bagi semua pembaca. Buku ini disusun untuk memenuhi syarat lulus mata kuliah Kerja Praktek.

Pada kesempatan ini patut kiranya penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada dosen pembimbing, Ibu Audrey Ayu Dianaris, S.SI. atas saran dan bantuan yang diberikan dalam penyusunan buku ini.Dalam penyusunan buku ini masih banyak kekurangan dan jauh dari kata sempurna. Atas segala kekurangan dari buku ini, Penulis berharap atas kritik dan saran sebagai masukan untuk penyusun kedepannya. Atas perhatian pembaca, terima kasih yang sebesar-besarnya kepada Anda.

Surabaya, Oktober 2022

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i

HALAMAN PENGESAHAN ii

SURAT PERNYATAAN KEASLIAN iii

ABSTRAK iv

ABSTRACT v

KATA PENGANTAR vi

DAFTAR ISI vii

DAFTAR GAMBAR viii

DAFTAR TABEL ix

DAFTAR ALGORITMA x

DAFTAR SEGMEN PROGRAM xi

DAFTAR RUMUS xii

BAB I PENDAHULUAN 1

1.1 Tujuan 1

1.2 Ruang Lingkup 1

1.3 Sistematika Pembahasan 2

BAB II TEORI PENUNJANG 3

2.1 HTML5 3

2.1.1 Canvas 3

2.1.2 3D 5

2.2 Kinect II 7

DAFTAR PUSTAKA 9

RIWAYAT HIDUP 11

LAMPIRAN A KUESIONER A-1

LAMPIRAN B TAMPILAN PROGRAM B-1

DAFTAR GAMBAR

Gambar Halaman

2.1 Pola Ukuran 6

2.2 Paragraf 8

2.3 Contoh Penulisan 10

2.4 Kutipan Satu Paragraf 12

DAFTAR TABEL

Tabel Halaman

2.1 Pola Ukuran 6

2.2 Paragraf 8

2.3 Contoh Penulisan 10

2.4 Kutipan Satu Paragraf 12

DAFTAR ALGORITMA

Algoritma Halaman

2.1 Pola Ukuran 6

2.2 Paragraf 8

2.3 Contoh Penulisan 10

2.4 Kutipan Satu Paragraf 12

DAFTAR SEGMEN PROGRAM

Segmen Program Halaman

2.1 Pola Ukuran 6

2.2 Paragraf 8

2.3 Contoh Penulisan 10

2.4 Kutipan Satu Paragraf 12

DAFTAR RUMUS

Rumus Halaman

2.1 Pola Ukuran 6

2.2 Paragraf 8

2.3 Contoh Penulisan 10

2.4 Kutipan Satu Paragraf 12

BAB I  
PENDAHULUAN

Bab ini merupakan bagian pembuka dari keseluruhan penulisan buku ini. Disini akan menjelaskan gambaran umum dan alasan perubahan sistem yang sudah ada pada PT. Adiputro Wirasejati. Saat ini perusahaan menggunakan sistem semi manual dibantu dengan program excel. Namun pada pelaksanaannya hal tersebut tidak mampu memberikan kebutuhan yang diinginkan oleh pengguna. Oleh karena itu dibangunkan sebuah sistem jaringan lokal yang mampu melakukan konversi Surat Perintah Kerja (SPK) yang mampu saling terintegrasi dengan database yang sudah ada saat ini. Di bab ini juga akan dijelaskan lebih dalam lagi terkait latar belakang sistem dibuat, tujuan sistem dibuat, ruang lingkup sistem yang akan dibangun, dan sistematika pembahasan

1. Latar Belakang

Peranan sistem sudah menjadi hal yang lumrah pada dunia saat ini. Sudah banyak perusahaan yang dalam kesehariannya tidak mampu terlepas dari peranan sebuah sistem. Baik itu adalah sistem yang berbasis manual maupun yang sudah terotomatisasi oleh teknologi. Saat ini banyak perusahaan sedang berlomba – lomba untuk mengoptimalkan sistem yang sudah mereka miliki. Tidak sedikit juga terjadi peralihan sistem yang sebelumnya bersifat manual menjadi otomatisasi.

Awal mula mengapa sistem ini mau dibuat dikarenakan karena sejumlah permasalahan yang sering dijumpai pada sistem yang saat ini sedang berjalan di perusahaan. Walaupun sistem yang berjalan sudah menggunakan bantuan komputer dengan program excel permasalahan tersebut tetap terjadi. Padahal pada perusahaan yang bergerak di bidang produksi mengharuskan proses produksi tetap berjalan optimal. Salah satu hambatan yang paling dirasakan pada sistem saat ini adalah proses keterlambatan pemenuhan pesanan komponen divisi perakitan setelah pihak admin melakukan identifikasi pada spesifikasi yang tercantum pada Surat Perintah Kerja (SPK).

Dengan demikian, diharapkan sistem pada divisi perakitan pada PT. Adiputro Wirasejati yang masih dilakukan secara manual dalam mengidentifikasi Surat Perintah Kerja yang diperlukan pada masing – masing departemen dapat diproses lebih cepat dengan sistem yang akan dibangun. Saat ini SPK masih dilakukan dengan print fisik dan diharapkan dapat tergantikan dengan SPK yang berbentuk elektrik yang dapat diakses langsung oleh beberapa departemen pada divisi perakitan. Dan tentunya tujuan yang sangat diharapkan dengan pembuatan aplikasi ini adalah mampu menggantikan sistem yang sudah ada.

1. Tujuan

Tujuan dari pembuatan aplikasi ini adalah unuk memenuhi permintaan klien. Secara garis besar aplikasi yang dibuat diharapkan mampu membaca dan mendeteksi parameter – parameter yang tercantum dalam sebuah Surat Perintah Kerja (SPK). Kemudian dari informasi tersebut akan digunakan sebagai acuan dalam mendeteksi komponen – komponen yang sekiranya diperlukan dalam satu SPK ini. Adapun tujuan lain yang didapatkan dari pembuatan aplikasi website konversi Surat Perintah Kerja adalah :

* Mempermudah proses input komponen – komponen yang dibutuhkan sesuai dengan Surat Perintah Kerja.
* Mempermudah proses input data – data yang diperlukan untuk melakukan produksi karoseri.
* Mempermudah proses identifikasi komponen – komponen yang diperlukan dalam pembuatan karoseri.
* Mempermudah proses melakukan pemesanan komponen di gudang.
* Mempercepat proses penyediaan komponen di divisi produksi.

1. Ruang Lingkup

Adapun pekerjaan yang akan dilakukan pada Kerja Praktek ini adalah membuat aplikasi konversi Surat Perintah Kerja berbasis website. Website ini nantinya kana digunakan untuk melakukan konversi spesifikasi yang terdapat pada SPK PT. Adiputro Wirasejati kedalam bentuk komponen. Untuk memastikan pekerjaan yang dilakukan tidak melebihi dari apa yang dikerjakan, maka diperlukanlah sebuah ruang lingkup untuk membatasi hal yang akan dikerjakan. Berikut adalah batasan – batasan ruang lingkup dalam pembangunan aplikasi nantinya:

* Sistem yang dibangun akan memfokuskan pada permasalahan yang berada pada permasalahan seputar konversi SPK
* Sistem yang dibangun akan menggunakan framework berbasis laravel.
* Sumber informasi klien akan berasal dari divisi perakitan
* Sistem yang dibangun tidak menerima permintaan dari divisi lain apalagi yang tidak berhubungan dengan sistem konversi SPK
* Menggunakan sistem login dan logout yang membagi usernya ke dalam beberapa role
* Pembagian database akan disesuaikan dengan departemen usernya
* Database akan menggunakan MongoDB
* Membuat koneksi antar database perusahaan (Microsoft SQL Server 2019)
* Tidak mengurus security terkait priveleges database pada MongoDB, sepenuhnya diserahkan kepada perusahaan.
  + 1. Fitur - fitur

Pembuatan aplikasi ini akan menerapkan pembagian user dalam beberapa level user. Di sub bab ini akan dijelaskan mengenai fitur – fitur yang akan dibuat dalam website ini dan dikategorikan berdasarkan level dari masing – masinglevel user. Pembagian level user dibagi berdasarkan role dari usernya. Berikut merupakan fitur – fitur yang akan dibuat yang dibedakan berdasarkan role:

1. **Super Admin**

User yang memiliki kemampuan untuk melakukan manajemen kepada sistem utama dan mampu melakukan semua hal dalam sistemnya.. User ini tidak dapat dihapus. Hanya diberikan kepada orang sistem. Adapun fitur yang diberikan kepada super admin antara lain :

Manajemen user

Registrasi user

Manajemen Departemen

Mengedit Database Utama

Tarik data SPK

Order komponen

Mengedit history SPK

1. **Admin**

Level user yang diberikan kepada kepala masing masing departemen pada divisi perakitan. Tidak mampu mengakses akses yang tidak disebutkan. Secara default user ini tidak tersedia dan mengharuskan super admin untuk membuatnya terlebih dahulu. Adapun fitur yang ditawarkan pada level ini antara lain :

Mengedit Database Utama

Tarik data SPK

Order komponen

Melihat History SPK

1. **Staff**

Level user yang paling lemah. User ini memiliki akses yang sangat terbatas. Seringkali memerlukan bantuan dari role level atas kalau terjadi kesalahan saat penginputan data. Adapun fitur yang ditawarkan pada level ini antara lain :

Tarik data SPK

Order komponen

Melihat history SPK.

1. Metodologi

Dalam pengembangan aplikasi website ini, tentunya kita menggunakan metodologi pengembangan perangkat lunak (Software Development Methodologies). Metodologi yang digunakan adalah metodologi Waterfall. Waterfall Development Methods adalah salah satu metodologi pengembangan perangkat lunak yang sangat sederhana dan cukup populer dipakai. Metodologi ini memanfaatkan proses berurutan dan sistematis dalam setiap fase pengerjaan softwarenya. Model pengembangannya dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah.

  
Gambar 1.1  
Metodologi Waterfall

Intisari dari Waterfall Development adalah memberikan gambaran alur pengerjaan yang jelas dan terukur. Terdapat alokasi waktu yang pasti agar setiap tim dapat menyelesaikan pekerjaan masing - masing tepat waktu. Selain dari itu, Waterfall juga menghemat biaya kepada klien dikarenakan tidak perlu membayar biaya tambahan pada saat ingin memberikan masukan atau feedback ke tim pengembang seperti pada metodologi Agile. Pembuatan aplikasi bersakala besar juga cocok menggunakan metodologi Waterfall. Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing, deployment, dan maintenance. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya tanpa menyelesaikan tahapan sebelumnya) dan berjalan secara berurutan, oleh karena itu di sebut waterfall (Air Terjun). Berikut penjelasan lebih detail terkait rencana implementasi metodologi:

1. Requirements

Pada tahapan dilakukan sesi tanya jawab dengan pemilik perusahaan mengenai kebutuhan dalam perusahaannya. Orang yang bertanggung jawab dari pihak perusahaan akan ditunjuk dan mengawasi kebutuhan pihak programmer. Sekiranya orang yang ditunjuk tersebut dapat ditanya dan paham terkait sistem yang saat ini sedang berjalan di PT. Adiputro Wirasejati. Mampu memberikan feedback dan mampu memberikan gambaran terkait sekiranya alur sistem yang ingin dibuat. Sekaligus menjelaskan permasalahan yang timbul pada sistem saat ini.

1. Design

Pada tahapan ini akan dilakukan perancangan sistem yang akan dibangun. Melihat dari analisis perangkat yang user butuhkan. Mendiskusikan dengan pihak yang bertanggung jawab. Di tahapan ini akan banyak desain gambar alur sistem yang akan diajukan ke divisi perakitan.

1. Development

Tahap Pengembangan aplikasi, didasarkan pada hasil dari desain di tahap nomor 2. Pada tahapan ini proses *coding* akan banyak dilakukan. Semua fitur akan dibuat pada tahapan ini. Tidak luput juga proses penginstallan perangkat lunak dan library yang diperlukan.

1. Testing

Testing dilakukan dengan mengetes setiap fitur yang ada. Ketika kesalahan pada fitur ditemukan maka code akan langsung diperbaiki. Hal ini ditujukan agar program yang siap di tanam di server PT. Adiputro Wirasejati dapat langsung dipakai tanpa ada kendala. Semua untuk memastikan semua fitur dari program yang telah dibangun dapat berjalan dengan sempurna.

1. Deployment

Deployment adalah tahapan penanaman program pada server perusahaan. Disini koneksi – koneksi jaringan akan diatur. Koneksi antar database juga akan diatur pada tahapan ini. Memastikan bahwa program yang dibuat dapat melakukan koneksi dengan database yang ada di perusahaan saat ini. Nantinya setelah ditanam, user akan diminta untuk malkukan uji coba pada program yang telah dibuat.

1. Maintenance

Maintenance adalah tahapan setelah melakukan deployment. Disini lebih berfokus pada penyelesaian hal – hal yang dianggap kurang oleh user. Sekaligus membenarkan beberapa *bug* atau kesalahan yang terjadi (opsional). Tahapan ini bisa dikatakan tahapan akhir juga. Sifat dari tahapan maintenance ini akan berkelanjutan.

Pertimbangan utama pemilihan metode ini didasarkan pada kebutuhan perusahaan yang sudah jelas dan pasti. Di awal pembuatan program ini sudah diberikan kejelasan bahwa program yang dibangun harus memenuhi kebutuhan perusahaan untuk mampu membaca, mengidentifikasi dan melakukan *cross check* dengan komponen – komponen yang diperlukan. *Cross check* dilakukan dengan menggunakan list yang ada pada database perusahaan saat ini. Selain itu kurun waktu yang singkat menjadikan pemilihan model ini cocok untuk diterapkan. Mengingat program yang dibangun hanya dibuat dalam kurun waktu 3 bulan yang termasuk ke dalam kategori pengerjaan yang singkat.

1. Sistematika Pembahasan

Dalam penulisan suatu laporan diperlukan sistematika pembahasan yang jelas dan runtut. Hal ini ditujukan untuk menggiring pembaca dalam memahami runtutan sistem yang dibangun. Sistematika pembahasan ini berisi ringkasan atas pembahasan yang dibagi menjadi beberapa bab. Sistematika pembahasan akan dimulai dari bab yang pertama yaitu pendahuluan, teori penunjang,

* BAB I : PENDAHULUAN

Dalam bab ini akan dijelaskan mengenai latar belakang, tujuan, ruang lingkup, metodologi, dan sistematika pembahasan dalam pembuatan aplikasi berbasis website PT. Adiputro Wirasejati.

* BAB II : TEORI PENUNJANG

Dalam bab ini akan dijelaskan mengenai seluruh kajian dan referensi yang menjadi landasan teori dalam mendukung pembuatan aplikasi berbasis website PT. Adiputro Wirasejati. Teori bersifat pengetahuan umum yang menjelaskan Surat Perintah Kerja,Front end, Back end, Database dan Arsitektur sistem.

* BAB III : ANALISIS SISTEM

Dalam bab ini akan dijelaskan mengenai analisis sistem yang sedang berjalan, alur kerja sistem yang sedang berjalan, kelemahan sistem lama dan penyelesaiannya, hasil analisis, analisis sistem baru, dan kebutuhan perangkat.

* BAB IV : SISTEM DESAIN

Dalam bab ini akan dijelaskan mengenai seluruh desain sistem dari aplikasi website PT. Adiputro Wirasejati. Terdapat desain arsitektur dari setiap role yang ada, desain database, struktur tabel, desain interface, dan nama dari setiap interface yang ada.

* BAB V : IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai setiap code yang ada aplikasi website PT. Adiputro Wirasejati. Segmen code akan banyak ditemukan pada bab ini. Segmen program nantinya akan dijelaskan baris per baris. Segmen yang dijelaskan adalah code yang memiliki fungsi penting terhadap keberlangsungan fitur – fitur yang ada.

* BAB VI : Kesimpulan Dan Saran

Pada bab ini akan menjelaskan kesimpulan akhir dalam pengerjaan web di PT. Adiputro Wirasejati. Pada bab ini akan terdapat kesimpulan dan saran.

BAB II

TEORI PENUNJANG

Dalam bab ini akan dijelaskan mengenai seluruh kajian dan referensi yang menjadi landasan dalam mendukung pembuatan aplikasi berbasis website PT. Adiputro Wirasejati. Hal ini diperlukan sebagai penunjang dan landasan teori dalam pembangunan web. Oleh karena itu pada bab ini akan berisi banyak mengenai informasi berupa pengetahuan dasar. Informasi yang akan dibahas adalah mengenai Surat Perintah Kerja, FrontEnd, BackEnd, Database, dan Arsitektur Sistem dari aplikasi berbasis website PT. Adiputro Wirasejati.

1. Surat Perintah Kerja

Pengertian dari Surat Perintah Kerja adalah surat resmi yang digunakan untuk memberikan perintah pada seseorang untuk melakukan suatu pekerjaan khusus. Surat ini biasanya memuat hal – hal apa saja yang harus dilakukan oleh pekerja, mulai dari instruksi untuk memulai pekerjaan hingga kepastian waktu yang harus dilaksanakan oleh pekerja. Surat Perintah Kerja pada umumnya meliputi siapa yang memberikan perintah, kepada siapa perintah tersebut ditujukan, kepastian waktu pekerjaan dimulai dan selesainya, biaya pekerjaan yang dilaksanakan dan sanksi bila ada keterlambatan.

Surat Perintah Kerja yang dimiliki oleh PT. Adiputro Wirasejati berisikan detail dari komponen – komponen yang digunakan dalam pembuatan karoseri dari sebuah kendaraan muat. Pada Surat Perintah Kerja PT. Adiputro Wirasejati terdapat header yang berisi tanggal surat perintah kerja dibuat, nama customer, alamat customer, nomor surat perintah kerja, tanggal penerimaan, dan status. Lalu pada bagian tengah teradapat detail dari kendaraan yang akan dibuatkan karoserinya. Detail dari kendaraan tersebut meliputi merk, model, tahun, nomor seri, nomor rangka, dan nomor mesin.

Lalu pada bagian terakhir dari Surat Perintah Kerja PT. Adiputro terdapat berbagai komponen yang dipesan untuk kendaraan tersebut. Berbagai komponen tersebut dicatat dengan pengkategorian sesuai dengan masing – masing bagian dari mobil. Kategori pertama adalah Body yang berisi bagasi toolkit, model body, lampu belakang, model trap tangga, dan variasi lampu (bila ada). Kategori selanjutnya ada Kaca yang berisi kaca samping. Untuk kategori ketiga adalah Eksterior yang berisi tinggi body. Untuk kategori keempat yaitu Interior berisikan detail dari bangku depan, warna bangku, deck samping, ducting, interior, lampu interior, karpet, dan rel bangku. Kategori selanjutnya merupakan Warna Cat dan Warna Body. Permintaan tambahan dari customer dituliskan pada bagian bawah dari Surat Perintah Kerja.

  
Gambar 2.1  
Contoh Surat Perintah Kerja PT. Adiputro Wirasejati

Seluruh data komponen yang ada pada Surat Perintah Kerja nantinya akan disimpan pada database sistem PT. Adiputro Wirasejati. Kemudian data – data dari komponen tersebut akan di teruskan ke divisi gudang dari PT. Adiputro Wirasejati. Divisi gudang nantinya akan mengirimkan komponen – komponen yang dibutuhkan untuk membuat karoseri ke divisi produksi. Namun saat ini proses identifikasi komponen dilakukan secara manual.

1. FrontEnd (Vue JS)

FrontEnd adalah salah satu bagian dari website yang menampilkan tampilan pada pengguna. Bagian ini dibuat dengan menggunakan Vue.Js, Cascading Style Sheets, dan juga template desain lainnya. Sehingga, suatu website bisa bekerja dengan baik. Vue.Js adalah salah satu framework atau library dari JavaScript yang digunakan untuk membuat tampilan (interface) pada website agar tampak lebih interaktif. Fungsi lain dari Vue.Js adalah membuat SPA (Single Page Application). Apabila digunakan pada arsitektur MVC (Model-View-Controller), maka Vue.Js menempati pada posisi View yang berjalan di sisi frontend.

Sehingga tugas utama dari framework ini adalah mengirim dan menerima data, kemudian membuat tampilan UI (User Interface) yang menarik. Framework ini juga sangat mudah untuk diintegrasikan dengan library yang lain. Hal ini dikarenakan pendekatan kerangka dari framework ini yang memfokuskan pada pembangunan sebuah komponen. Jika diimplementasikan pada komponen HTML, maka Vue,Js menggunakan ID, class, atau name untuk menginiliasiasikannya.

Framework ini mengalami perkembangan yang sangat pesat, bahkan beberapa perusahaan besar menggunakannya, sebagai contoh Google dan Adobe. Terlepas dari itu, Vue.Js pertama kali dikembangkan pada tahun 2013 oleh Evan You yang sebelumnya bekerja di Google dengan menggunakan AngularJs. Hal itu yang melatar-belakangi seorang Evan You mengembangkan sebuah library yang lebih ringan daripada AngularJs.

1. Backend (Laravel)

BackEnd merupakan bagian dari website yang bertanggung jawab untuk menyediakan kebutuhan yang tak terlihat oleh pengguna (tidak berinteraksi langsung dengan pengguna), seoerti bagaimana data disimpan, diolah, serta ditransaksikan secara aman. Itu semua bertujuan untuk mendukung bagian FrontEnd bekerja sesuai dengan fungsinya. Bagian BackEnd pada website ini dikerjakan dan dihubungkan dengan menggunakan suatu framework. Framework yang digunakan dalam website konversi kode Surat Perintah Kerja ini adalah Laravel.

Laravel merupakan framework yang dapat membantu pengembangan website dalam memaksimalkan penggunaan PHP dalam proses pengembangan website. Seperti diketahui, PHP sendiri merupakan bahasa pemograman yang cukup dinamis. Dimana kehadiran Laravel kemudian membuat PHP menjadi lebih powerful, cepat, aman, dan simple. Terlebih lagi, framework Laravel selalu memunculkan teknologi terbarunya diantara framework PHP lain.

Framework ini mengikuti struktur MVC atau Model-View-Controller. Dimana MVC adalah metode aplikasi yang memisahkan data dari tampilan berdasarkan komponen – komponen aplikasi, seperti manipulasi data, controller, dan user interface. Penggunaan struktur MVC ini membuat Laravel mudah untuk dipelajari dan mampu mempercepat proses pembuatan prototipe aplikasi web. Sebagaimana ia juga menyediakan fitur bawaan seperti otentikasi, email, perutean, sesi, dan daftar berjalan.

Laravel juga memiliki kelebihan terhadap fokusnya pada end-user, yang artinya hanya memfokuskan perhatian pada kejelasan dan kesederhanaan, baik dari penulisan hingga tampilan. Laravel juga bisa menghasilkan fungsional aplikasi web yang berjalan dengan semestinya. Hal ini menjadikan laravel menjadi salah satu pilihan framework dalam pengembangan suatu aplikasi dimulai dari skala kecil hingga besar. Selain itu laravel juga memiliki beberapa bantuan fitur yang diunggulkan seperti template, routing dan modularity yang sangat memberikan kemudahan dalam proses pengembangan websitenya.

1. Database

Database atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis disebut sistem manajemen basis data (database management system system, DBMS).

* + 1. MongoDB

Database yang digunakan dalam website konversi Surat Perintah Kerja adalah MongoDB. MongoDB adalah salah satu jenis database yang cukup populer digunakan dalam pengembangan website. Berbeda dengan database jenis [SQL](https://www.niagahoster.co.id/blog/apa-itu-query-sql/) yang menyimpan data menggunakan relasi tabel, MongoDB menggunakan dokumen dengan format [JSON](https://www.niagahoster.co.id/blog/json-adalah/).  Hal inilah yang dianggap membuat pengelolaan data menggunakan MongoDB lebih baik. Apalagi MongoDB memiliki fleksibilitas yang tinggi menjadikannya pilihan yang cocok untuk pengerjaan proyek yang kebanyakan bersifat dinamis. Sistem database ini menggunakan beberapa komponen penting, yaitu :

1. Database – merupakan wadah dengan struktur penyimpanan yang disebut collection.
2. Collection – merupakan tempat kumpulan informasi data yang berbentuk dokumen.
3. Document – merupakan satuan unit terkecil dalam MongoDB.

Sebagai satuan terkecil, dokumen akan berisi baris – baris data tanpa skema tertentu, tapi berupa struktur pasangan key-value. Key digunakan untuk melacak objek dengan (value) nilai yang bervariasi, seperti data angka, string, atau objek kompleks lainnya. Dengan format dokumen tersebut, MongoDB mampu menampung data yang lebih bervariasi, seperti data angka, string, atau objek kompleks lainnya. Selain itu, bisa juga dilakukan scale-out database untuk meningkatkan kapasitas data tanpa mengganggu performa server.

* + 1. Microsoft SQL Server

Saat ini perusahaan menggunakan Microsoft SQL Server. Microsoft SQL Server adalah salah satu software jenis Relational Database Management System (RDBMS) yang cukup sering digunakan. Sesuai namanya, software ini dikembangkan oleh perusahaan besar Microsoft dan cukup scalable. Maksudnya adalah untuk pemakaiannya Microsoft SQL Server bisa dimana – mana dari laptop manapun, ke jaringan server cloud dan lain – lain. Namun istilah scalable ini tetap saja harus memperhatikan persyaratan hardware maupun software. Sejak rilis pertama kali pada tahun 1989, Microsoft SQL Server sudah menjadi pilihan dari para pengguna database dan hingga saat ini masih banyak dipakai. Perkembangannya pun cukup menjanjikan mulai dari versi SQL Server 1.0 yang pertama dirilis dan hingga saat ini. Fungsinya pun sekarang tidak hanya sebagai RDBMS (Relational Database Management System) saja tapi bisa lebih dari itu, termasuk alat built-in intelijen bisnis, serta bisa juga digunakan sebagai analisis dan pelaporan.

Sebagai salah satu produk dari perusahaan besar Microsoft, tentunya keberadaan Microsoft SQL Server ini memiliki fungsi yang vital dalam dunia management database. Sesuai dengan namanya software ini melakukan pengelolaan database dengan menggunakan query atau perintah SQL. SQL atau Structures Query Language merupakan suatu bahasa komputer yang diakui dunia dan digunakan untuk mengakses data – data yang tersimpan dalam suatu database.

1. Arsitektur Sistem

Arsitektur Sistem adalah suatu pendekatan terhadap desain dan perencanaan situs yang, seperti arsitektur itu sendiri, melibatkan teknis, kriteria estetis dan fungsional. Seperti dalam arsitektur tradisional, fokusnya adalah benar pada pengguna dan kebutuhan pengguna. Hal ini memerlukan perhatian khusus pada konten website, rencana bisnis, kegunaan, desain interaksi, informasi dan desain arsitektur website. Untuk optimasi mesin pencari yang efektif perlu memiliki apresiasi tentang bagaimana sebuah situs Web terkait dengan World Wide Web.

Arsitektur Sistem yang digunakan dalam proses pengembangan website konversi Surat Perintah Kerja PT. Adiputro Wirasejati adalah Model – View – Controller (MVC). Model View Controller atau yang dapat disingkat MVC adalah sebuah pola arsitektur dalam membuat sebuah aplikasi website dengan cara memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Dalam implementasinya kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. Berikut adalah penjelasan dari MVC:

* Model

Bagian yang berfungsi untuk menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada pada database. Model juga berfungsi sebagai pengelola perilaku dan data pada domain aplikasi, melakukan tanggapan terhadap permintaan informasi dan merespons instruksi untuk merubah suatu kondisi (state). Model biasanya merepresentasikan sekiranya data apa yang harus ada di pada sebuah aplikasi. Kumpulan dari data tersebut yang kita sebut model.

* View

Adalah bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI) sehingga enak untuk diliat. Pada website ini, view akan dipadukan dengan Vue.JS. View juga berfungsi untuk menerjemahkan informasi yang berasal dari model ke dalam sebuah bentuk yang sesuai untuk berinteraksi dengan pengguna. Biasanya berupa satu atau lebih elemen antar muka pengguna.

* Controller

Adalah bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung. Controller menerima masukan dari user dan memicu respons dengan membuat pemanggilan ke objek – objek model. Controller biasanya menampung logic – logic yang akan dipanggil saat inputan dilakukan. Biasanya controller akan melakukan aksi terhadap model dan view.

  
Gambar 2.2  
Alur Proses MVC

Alur kerja sistem MVC tergolong sederhana seperti yang ditunjukkan pada gambar diatas. Untuk memudahkan pemahaman terkait cara kerja MVC, analogikan MVC sebagai sebuah restoran, dimana customer adalah view, controller adalah pelayan dan chef adalah model. Ketika customer memesan menu, pelayan akan melakukan pencatatan pesanan dan menginformasikannya ke chef. Chef yang sudah mengerti pesanan customer akan segera mencarikan bahannya dan segera memasakkan pesanannya. Alur model dari arsitektur kurang lebih mirip seperti analogi diatas, namun untuk penjelasan lebih teknis dapat dilihat pada proses dibawah ini:

1. Proses pertama adalah view akan meminta data untuk ditampilkan dalam bentuk grafis kepada pengguna.
2. Permintaan tersebut diterima oleh controller dan diteruskan ke model untuk diproses.
3. Model akan mencari dan mengolah data yang diminta di dalam database.
4. Setelah data ditemukan dan diolah, model akan mengirimkan data tersebut kepada controller untuk ditampilkan di view.
5. Controller akan mengambil data hasil pengolahan model dan mengaturnya di bagian view untuk ditampikan kepada pengguna.

BAB III  
ANALISIS SISTEM

Pada bab ini akan dijelaskan mengenai analisis sitem yang sedang berjalan, kelemahan sistem lama, hasil analisis, analisis sistem baru, keunggulan sistem baru, dan kebutuhan perangkat lunak selama proses pengembangan aplikasi berbasis website dari PT. Adiputro Wirasejati ini. Kegiatan Analisa sangat penting untuk dilakukan karena dengan menganalisa pembuat dapat memahami seluruh informasi yang terdapat pada suatu kasus, dan juga isu apa yang sedang terjadi. Analisa bertujuan untuk mencari solusi yang tepat untuk masalah yang ada. Semua ini ditujukan untuk memastikan program yang dibangun tepat sasaran tujuan.

1. Analisis Sistem yang Sedang Berjalan

Pada sub bab ini akan dijelaskan mengenai sistem yang sedang berjalan pada perusahaan saat ini. Saat ini proses pemesanan komponen masih banyak dilakukan secara manual. Saat Surat Perintah Kerja (SPK) diterima pada masing – masing divisi, maka divisi akan melakukan proses identifikasi untuk komponen yang diperlukan pada SPK tersebut. Sistem bantu untuk mencatat komponen masih sangat sederhana, salah satu alat bantu yang digunakan disini adalah program excel yang dibantu dengan fungsi macro. Namun sistem tersebut tidak resmi dan pada masing – masing divisi belum ada keseragaman terkait program excel. Adapun beberapa kelemahan yang timbul dari sistem saat ini antara lain sebagai berikut:

Proses kesiapan komponen yang belum siap pada divisi perakitan dikarenakan keterlambatan proses pemesanan komponen – komponen yang tercantum di SPK.

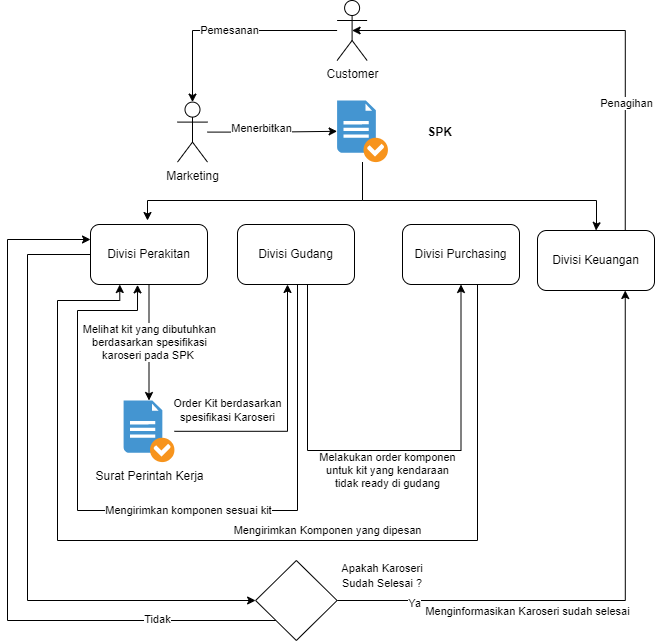
Memungkinkan terjadinya kesalahan pemesanan komponen karena proses mengidentifikasi komponen dari SPK masih dilakukan secara manual.’

Terjadinya keterlambatan pada proses kelanjutan setelah pemesanan komponen karena proses pemesanan komponen yang lambat.

Tidak bisa melakukan antisipasi lebih awal untuk komponen – komponen yang memiliki jumlah persediaan yang jumlahnya sedikit atau hampir habis di gudang.

1. Alur Kerja Sistem yang Sedang Berjalan

Pada bagian ini akan menjelaskan alur dari sistem yang sedang berjalan di PT.Adiputro Wirasejati. Alur dari sistem yang ada sebenarnya tidak terlalu rumit dan tergolong sederhana. Sistem yang ada saat ini terutama yang terkait dengan proses pemesanan barang komponen ke gudang masih cenderung ke sisi manual walaupun memang dalam kenyataannya dibantu dengan komputer. Dari alur kerja nanti inilah yang akan dipakai sebagai bahan pertimbangan untuk mengembangkan web pembantu sistem yang ada saat ini. Berikut adalah gambar dari alur sistem yang sedang berjalan di PT. Adiputro.

  
Gambar 3.1   
Alur sistem yang sedang berjalan

Sistem yang saat ini sedang berlanjut pada PT. Adiputro Wirasejati dimulai dari customer melakukan pemesanan karoseri. Di PT. Adiputro customer bebas memilih model karoseri, dan interior setelah diajak berkeliling dengan pihak marketing. Jika customer setuju sudah menentukan pilihannya, barulah proses negosiasi dimulai. Negosiasi yang berhasil akan ditandai dengan divisi marketing menerbitkan SPK yang berisikan data – data dimulai dari tanggal SPK dibuat, nomor SPK, tanggal penerimaan pemesanan, status SPK, nama dari customer, dan alamat dari customer, merk mobil, tipe mobil, nomor mesin, nomor rangka, tahun mobil. Pastinya dalam SPK tadi akan dicantumkan spesifikasi karoseri yang diinginkan oleh customer dimulai dari body, kaca, warna cat, interior dan tambahan – tambahan.

SPK kemudian akan diteruskan divisi perakitan dan divisi keuangan. Setelah divisi gudang menerima SPK, maka pihak gudang akan mengidentifikasi kit – kit yang diperlukan berdasarkan spesifikasi karoseri. Kit adalah sebutan di PT. Adiputro yang memuat beberapa komponen. Selanjutnya divisi perakitan akan melakukan proses pemesanan berdasarkan nama dan jumlah kit yang diperlukan ke divisi gudang.

Divisi gudang yang mendapatkan pesanan kit selanjutnya akan mencarikan komponen - komponen yang diperlukan pada masing – masing kit. Divisi gudang yang sudah selesai mendapatkan semua komponen pada kit yang diperlukan akan segera mengirimkan kembali pada divisi perakitan. Adakalanya komponen yang diperlukan tidak ada sehinngga divisi gudang perlu melakukan permintaan pembelian komponen ke divisi purchasing. Setelah kit yang dipesan sampai, maka kit akan diteruskan ke divisi perakitan. Sehingga proses perakitan karoseri dapat segera dikerjakan.

Proses perakitan kendaraan bisa dilaksanakan apabila seluruh kit yang dibutuhkan dan sudah berada di divisi perakitan. Proses order dari divisi perakitan ke divisi gudang membutuhkan waktu. Untuk itu proses order menjadi hal yang sangat penting. Divisi perakitan diharuskan untuk melakukan order dengan cepat ke divisi gudang sehingga proses penyelesaian karoseri dapat berjalan dengan efektif dan efisien. Maka dari itu diharapkan dengan sistem yang baru nantinya dapat membantu dalam mengatasi permasalahan tersebut.

1. Kelemahan Sistem lama dan penyelesaiannya

Pada penerapan sistem yang saat ini sedang berjalan, PT. Adiputro seringkali menghadapi permasalahan. Permasalahan utama yang dihadapi berada pada divisi perakitan. Keterlambatan dalam proses perakitan karoseri adalah hal utama dan fatal yang menjadi permasalahan disini. Berikut adalah ringkasan tabel permasalahan dan sekiranya penyelesaiannya.

Tabel 3.1

Kelemahan Sistem Lama

|  |  |  |
| --- | --- | --- |
| No. | Permasalahan | Penyelesaian |
| 1 | Identifikasi kit yang diperlukan masih dilakukan secara manual untuk beberapa departemen pada divisi perakitan | Membuat sebuah database master utama yang mampu mengidentifikasi kit – kit yang diperlukan pada spesifikasi tertentu |
| 2 | Belum ada kesegaraman sistem antar masing – masing departemen di divisi perakitan | Membuat sebuah sistem yang mampu membagi usernya berdasarkan departemen |
| 3 | Data update SPK belum realtime | Membuat jadwal penarikan data dan membuat sebuah tombol trigger untuk penarikan data SPK terbaru. |
| 4 | Human error karena kesalahan manusia saat mengidentifikasi kit | Membuat logic yang mampu mengidentifikasi kit yang diperlukan |
| 5 | Keterlambatan perakitan karoseri karena komponen yang tidak siap tepat waktu di lapangan | Pembuatan fitur print saat input spk supaya divisi perakitan dapat segera memesan barang di gudang |

Pada sistem yang lama, perusahaan masih menggunakan sistem manual, yang mana dalam pengerjaan atau proses pengolahan data sehari hari masih menggunakan sistem manual. Pemilik perusahaan atau yang mengolah data gaji dan absensi akan mencatat satu persatu setiap harinya , hal ini sangat membuang waktu dan rawan dalam salah input data atau ***human error***.

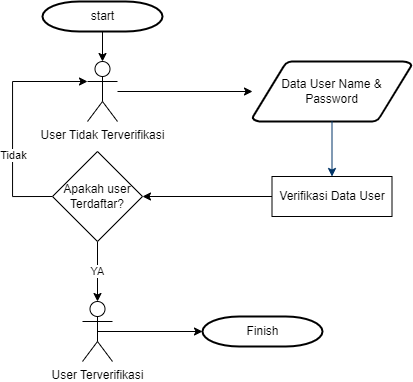
1. Hasil Analisis

Dari sistem yang sedang berjalan saat ini pada PT. Adiputro Wirasejati, proses pengerjaan kendaraan menjadi tidak efisien dikarenakan pemasalahan – permalahan pada sub bab 3.3. untuk meminimalisir sektor bisnis terpenting yaitu pada divisi perakitan. Maka dibangunlah sebuah sistem yang mampu membantu divisi tersebut. Selain itu pihak dari divisi perakitan meminta agar sistem yang dibangun dikunci dengan sistem login yang membagi usernya ke dalam beberapa level user dan akses dari database dibagi berdasarkan departemen.

* + 1. Alur Kerja Sistem yang Baru

Alur kerja sistem yang baru bertugas untuk memberikan gambaran terkait alur atau gambaran dari cara kerja sistem. Disini fokus utama yang menjadi perhatian dari alur kerja sistem yang baru terletak pada konversi spesifikasi SPK ke komponen – komponen. Namun perlu diketahui ada beberapa tahapan yang harus dilalui sebelum memulai proses konversi di sistem yang akan dibangun. Berikut adalah tahapan dari alur kerja sistem.

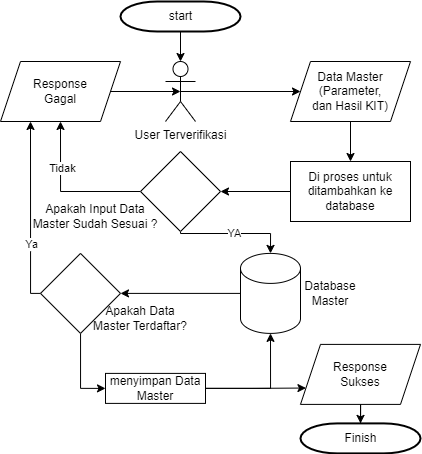
1. **Proses Login**



Gambar 3.2   
Flow Login

Proses Login Sangatlah sederhana. User hanya perlu memasukan Data berupa username dan Password. Kemudian sistem akan melakukan verifikasi terhadap data yang diinput user. Jika ternyata username dan password ditemukan kesamaan pada database, maka user dapat login dan usernya akan mendapati role sesuai dengan data username dan password yang diinput oleh user. Sebaliknya jika tidak terdapat kesalahan data maka user akan tetap tidak terverifikasi dan tidak bisa masuk ke sistem. User yang belum login tidak dapat melakukan apa – apa di sistem.

1. **Input Database Master**



Gambar 3.3  
Flow Input Master

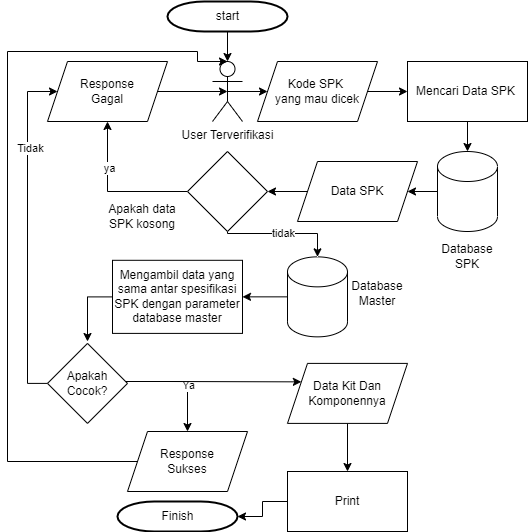
Proses Input Master Dimulai dengan user yang terverifikasi melakukan input data master pada form input data master. Data Master Berisikan 2 hal utama yaitu Parameter dan Hasil Kit. Parameter adalah faktor penentu yang digunakan sebagai data yang akan di cross check dengan data spesifikasi yang ada pada SPK perusahaan. Parameter berisikan tipe mobil, model mobil, tinggi mobil, departemen, stall, stock dan new parameter. Khusus untuk new parameter, data parameter yang diisi adalah data selain parameter yang telah disebutkan. Hal ini dikhususkan untuk mengatasi permasalahan spesifikasi SPK yang sangat beragam. Kit nantinya akan diisi dengan kode kit, dan saat begitu diinputkan akan memperlihatkan sekiranya komponen apa saja yang diperlukan berserta jumlahnya. Nanti kit ini adalah hasil saat spesifikasi pada data SPK perusahaan *matching* dengan data parameter pada database master Mongo.

Sebelum diinput ke database master, data yang diinput akan dicek terlebih dahulu. Apakah input data master sudah sesuai prosedur ?. Jika tidak akan memberikan response gagal ke user. Jika iya maka data akan dicek lagi ke database master yang ada sekarang untuk memastikan apakah data master yang diinput user sudah pernah terdaftar sebelumnya?. Jika terdapat kesamaan data maka sistem akan mengirimkan response gagal juga ke user. Namun jika tidak ditemukan kesamaan, maka sistem akan melakukan proses menyimpan ke database master dan sistem akan mengeluarkan response sukses ke user. Hal ini akan berlaku kurang lebih sama dengan cara mengupdate database master yang sudah ada sebelumnya.

1. **Input SPK untuk konversi ke komponen**

Proses Input SPK untuk dikonversi Dimulai dengan user menginputkan data SPK yang mau di konversikan. Proses input SPK sangat sederhana user hanya perlu memasukkan kode SPK dan stall. Selanjutnya SPK akan diambilkan dari database SPK perusahaan saat ini. Jika tidak ditemukan maka sistem akan mengirimkan response gagal.

Data SPK yang telah ditemukan selanjutnya akan di cross check dengan database master. Data SPK yang diambil adalah spesifikasi kemudian dicocokkan dengan data parameter yang ada di master. Jika tidak ditemukan kesamaan maka akan diberikan response gagal, sebaliknya jika ditemukan maka sistem akan mengirimkan data kit dan komponennya. Datanya ini nanti siap di print dan user dapat dengan segera melakukan proses pemesanan di gudang. Dibawah ini adalah alur dari proses konversi komponen.



Gambar 3.3  
Flow Input dan konversi

* + 1. Analisis Sistem Baru

Pada sub-bab ini akan menjelaskan sistem baru yang akan diterapkan di PT. Adiputro Wirasejati. Sistem yang baru akan berbasis aplikasi web yang diharapkan mampu mengganti sistem manual yang lama. Aplikasi website ini akan menjadi solusi dari permasalahan yang terjadi pada alur sistem yang lama. Dengan adanya aplikasi website ini, diharapkan proses pemesanan komponen yang dilakukan pada divisi perakitan dapat menjadi lebih baik.

Sistem yang baru akan menggunakan sistem satu pintu yang mengharuskan user melakukan login terlebih dahulu. User yang belum login hanya mampu melihat halaman home dan tidak bisa melakukan apa – apa. Sebaliknya user yang telah login dapat melakukan banyak hal. Pada sistem yang baru ini user sistem akan dibagi ke dalam 3 level dengan kemampuan sebagai berikut:

* + - 1. Super Admin

Super Admin merupakan role dengan kemampuan tertinggi. Artinya, role super admin bisa memegang kendali atas kedua role lainnya. Role ini berada di level paling tinggi jika diliat berdasarkan hirarki sebuah sistem. Role Super Admin juga yang akan mengatur role lain. Berikut merupakan fitur – fitur yang ada pada role super admin:

* Manajemen user

Manajemen user merupakan fitur utama dari role super admin. Fitur ini memberikan user dengan role super admin untuk melakukan manajemen akun pada aplikasi. User lain yang memiliki role selain super admin tidak dapat mengakses fitur ini. Hal ini dimaksudkan untuk memberikan kemudahan dalam mengorganisir akun – akun dalam website. Fitur ini memungkinkan seorang Super Admin untuk:

Menghapus user yang sudah ada

Mengubah detail dari user yang sudah ada

Mengubah role dari user yang sudah ada

Registrasi user baru

Fitur lain dari role Super Admin adalah melakukan registrasi user baru. Dimana fitur ini berfungsi untuk membuat user baru yang dapat digunakan dalam akses login nantinya. Disaat yang bersamaan user akan diberikan role sesuai dengan kebutuhan pekerjaan. Selain itu pastinya user tadi akan ditempatkan dalam sebuah departemen yang bertujuan untuk membatasi akses akun supaya tidak mengakses data pada departemen lain.

* Manajemen departemen

Fitur ketiga dari role Super Admin adalah melakukan manajemen departemen. Manajemen departemen dilakukan untuk mengatur pekerjaan dari setiap user yang ada. Departemen disini bertujuan untuk mengelompokkan akun nantinya sekaligus mengatur akses database pada masing – masing departemen. Disini super admin mampu menambah, mengedit departemen yang sudah ada dan menghapus departemen (dengan beberapa ketentuan khusus) serta mengatur akses database pada masing - masing departemen.

* Input data ke database utama

Fitur keempat dari Role Super Admin adalah melakukan input data ke database utama dari PT. Adiputro Wirasejati. Fitur ini memungkinkan seorang Super Admin untuk melakukan input data – data master menuju database utama. Data – data master berisi seluruh data pemesanan dan komponen – komponen berupa kode SPK yang diperlukan dalam proses produksi.

* Tarik data dari Kode Surat Perintah Kerja

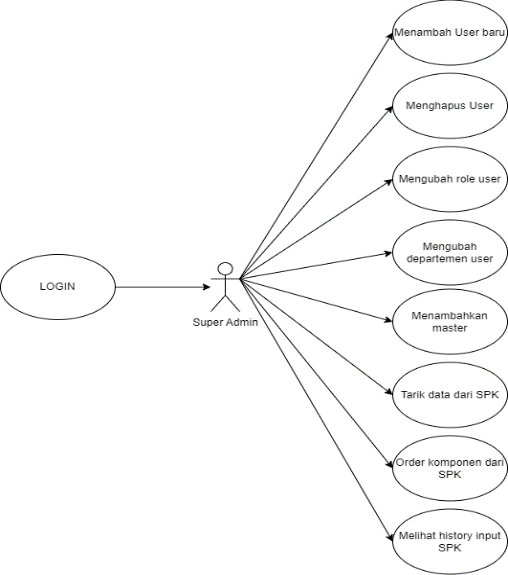
Fitur tarik data dari Kode Surat Perintah Kerja merupakan fitur kelima dari role Super Admin. Fitur tarik data dari Surat Perintah Kerja bertujuan untuk menampilkan seluruh data yang tersimpan dalam Surat Perintah Kerja. Saat nomor Surat Perintah Kerja sudah di masukkan, kolom – kolom komponen akan terisi secara otomatis. Super Admin bisa menghapus, mengganti, dan menambahkan komponen yang sudah ada.

* Order komponen dari Surat Perintah Kerja

Fitur ini merupakan fitur yang sangat penting pada aplikasi website ini. Setelah melakukan penarikan data dari Surat Perintah Kerja, Super Admin dapat melakukan order agar komponen, kit, atau mesin kendaraan bisa dikirimkan dari divisi gudang ke divisi perakitan. Fitur ini akan berjalan secara otomatis saat tombol print list komponen ditekan. Hasil print list komponen tadi akan dipakai sebagai dasar order memesan barang.

* History input kode Surat Perintah Kerja

Fitur terakhir dari user Super Admin adalah melihat history input kode Surat Perintah Kerja. Fitur ini akan memberikan daftar histori dari seluruh user admin yang telah melakukan input kode Surat Perintah Kerja. Fitur ini menjadi penting bagi karena Super Admin perlu melakukan pengawasan terhadap seluruh admin dalam melakukan input kode Surat Perintah Kerja. Disini super admin memiliki kemampuan untuk mengecek history input kode SPK yang salah dan menghapusnya.

  
Gambar 3.4  
Use Case Role Super Admin

* + - 1. Admin

Admin merupakan role kedua yang ada di aplikasi website PT. Adiputro Wirasejati. Role Admin memiliki beberapa fitur yang penting dalam proses produksi. Role Admin berada satu tingkat dibawah role Super Admin. Berikut merupakan fitur – fitur yang ada pada role admin:

* Input data ke database utama

Fitur pertama dari role Admin adalah melakukan input data ke database utama dari PT. Adiputro Wirasejati. Fitur ini memungkinkan seorang Admin untuk melakukan input data – data master menuju database utama. Data – data master berisi seluruh data pemesanan dan komponen – komponen berupa kode SPK yang diperlukan dalam proses produksi.

* Tarik data dari Kode Surat Perintah Kerja

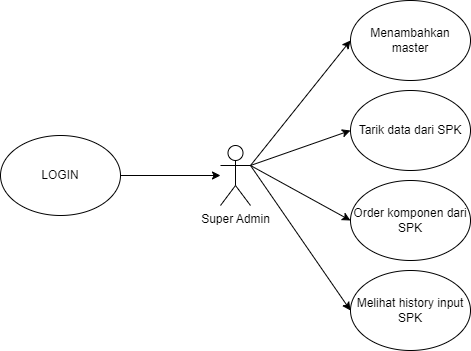
Fitur tarik data dari Kode Surat Perintah Kerja merupakan fitur kedua dari role Admin. Fitur tarik data dari Surat Perintah Kerja bertujuan untuk menampilkan seluruh data yang tersimpan dalam Surat Perintah Kerja. Saat nomor Surat Perintah Kerja sudah di masukkan, kolom – kolom komponen akan terisi secara otomatis. Super Admin bisa menghapus, mengganti, dan menambahkan komponen yang sudah ada.

* Order komponen dari Surat Perintah Kerja

Fitur ini merupakan fitur yang sangat penting pada aplikasi website ini. Setelah melakukan penarikan data dari Surat Perintah Kerja, Admin dapat melakukan order agar komponen, kit, atau mesin kendaraan bisa dikirimkan dari divisi gudang ke divisi perakitan. Fitur ini akan berjalan secara otomatis saat tombol print list komponen ditekan. Master yang dibuat pada database utama harus ada dan sesuai dengan parameter – parameter yang terdapat pada surat perintah kerja untuk membuat fitur ini berfungsi dengan baik.

* History input kode Surat Perintah Kerja

Fitur terakhir dari user Admin adalah melihat history input kode Surat Perintah Kerja. Fitur ini akan memberikan daftar histori dari seluruh user admin yang telah melakukan input kode Surat Perintah Kerja. Fitur ini menjadi penting bagi karena Admin perlu melakukan pengawasan terhadap seluruh admin dalam melakukan input kode Surat Perintah Kerja. Di role admin, hostory yang salah input tidak dapat dihapus.

  
Gambar 3.5  
Use Case Role Admin

* + - 1. Staff

Staff merupakan role ketiga yang ada di aplikasi website PT. Adiputro Wirasejati. Role Staff memiliki fitur paling sedikit bila dibandingkan dengan dua role lainnya. Role staff biasanya akan digunakan untuk karyawan lapangan dan tentu saja memiliki akses level yang secukupnya. Berikut merupakan fitur – fitur yang ada pada role staff :

* Tarik data dari Kode Surat Perintah Kerja

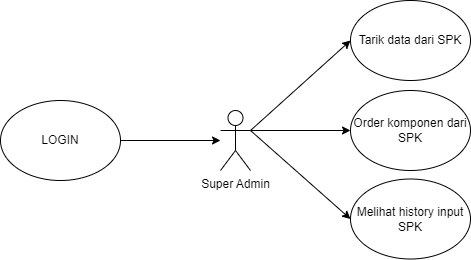
Fitur tarik data dari Kode Surat Perintah Kerja merupakan fitur pertama dari role Staff. Fitur tarik data dari Surat Perintah Kerja (SPK) bertujuan untuk menampilkan seluruh data yang tersimpan dalam SPK. Saat nomor Surat Perintah Kerja sudah di masukkan, kolom – kolom komponen akan terisi secara otomatis. Super Admin bisa menghapus, mengganti, dan menambahkan komponen yang sudah ada.

* Order komponen dari Surat Perintah Kerja

Fitur ini merupakan fitur yang sangat penting pada aplikasi website ini. Setelah melakukan penarikan data dari Surat Perintah Kerja, Staff dapat melakukan order agar komponen, kit, atau mesin kendaraan bisa dikirimkan dari divisi gudang ke divisi perakitan. Fitur ini akan berjalan secara otomatis saat tombol print list komponen ditekan. Perlu diingat agar fitur ini berjalan dengan semestinya, master data juga harus sesuai.

* History input kode Surat Perintah Kerja

Fitur terakhir dari user Staff adalah melihat history input kode Surat Perintah Kerja. Fitur ini akan memberikan daftar histori dari seluruh user admin yang telah melakukan input kode Surat Perintah Kerja. Fitur ini menjadi penting bagi karena Admin perlu melakukan pengawasan terhadap seluruh admin dalam melakukan input kode Surat Perintah Kerja. Di role staff , sama halnya dengan role admin, history yang salah input tidak dapat dihapus.

  
Gambar 3.6   
Use Case Role Staff

* + 1. Pembagian Database

Di PT. Adiputro terdapat beberapa departemen pada divisi perakitan. Nantinya pembagian databasenya akan mengacu pada departemen dari akun user itu sendiri. Departemen diberikan akses database minimal 1 database dan bisa lebih dari itu. Saat ini di PT. Adiputro database dibagi berdasarkan tipe SPKnya yaitu SPK Bus dan SPK Mini Bus. Adapun pembagian manajemen database sebagai berikut:

1. SPK Bus

SPK Bus adalah tipe SPK yang sekiranya modelnya dikatakan bus. Spesifikasinya sendiri bisa dilihat dari jumlah kursinya kadang bisa 60-80 kursi. Panjang mobilnya sendiri 12 meter atau lebih. Departemen pada divisi yang memegang akses SPK bus adalah dibawah ini.

Departemen Rangka Bus

Departemen Paneling

Departemen Putty Bus

Departemen Painting Bus

Departemen Trimming Bus

Departemen Finishing Bus

Departemen Sub Assy Bus

Departemen Other

1. SPK Mini Bus

SPK Bus adalah tipe SPK yang sekiranya modelnya dikatakan bus. Spesifikasinya sendiri bisa dilihat dari jumlah kursinya kadang bisa 25-35 kursi. Panjang mobilnya sendiri 7 meter sampai 8,5 meter. Departemen pada divisi yang memegang akses SPK mini bus adalah dibawah ini.

Departemen Body Welding

Departemen Rangka Minibus

Departemen Putty Minibus

Departemen Painting Minibus

Departemen Trimming Minibus

Departemen Finishing Minibus

Departemen Sub Assy Minibus

* + 1. Keunggulan Sistem Baru

Pembuatan sebuah sistem yang baru dipastikan untuk memberikan nilai tambah yang lebih baik dibandingkan pada sistem yang lama. Sesuai dengan permasalahan yang telah dijabarkan diatas, pembangunan aplikasi web ini akan memfokuskan pada divisi perakitan. Pastinya banyak manfaat yang akan didapatkan setelah menerapkan sistem yang baru. Beberapa keunggulan yang ditawarkan antara lain:

* Mampu menyajikan data SPK secara real time

Dengan penerapan sistem yang baru, data SPK dapat dengan mudah diupdate. Ada beberapa pola bantuan yang ditawarkan yaitu dengan menekan tombol untuk melakukan proses update data atau menunggu data melakukan proses update berdasarkan jadwal jam yang telah ditentukan. Di sistem lama terkadang pihak staff yang melakukan proses input harus melakukan proses konfirmasi pada atasannya untuk mengecek SPK yang baru. Dengan adanya sistem yang baru hal tersebut dapat dieliminasi.

* Proses penyediaan komponen – komponen di divisi perakitan tepat waktu

Memberikan efisiensi dalam proses pemesanan komponen – komponen di gudang berdasarkan jumlah dan nama kit. Diharapkan proses perakitan dapat segera dikerjakan saat mobil tiba pada departemen yang bersangkutan. Karena seringkali di sistem yang lama, perakitan tidak dapat segera dilakukan karena keterlambatan kedatangan komponen di departemen yang bersangkutan. Sudah terlihat jelas bahwa waktu penyediaan komponen menjadi hal yang sangat penting dan perlu diperhatikan.

* Standarisasi sistem yang berjalan

Proses kerapian dalam sebuah sistem menjadi salah satu hal yang penting untuk diperhatikan. Di beberapa perusahaan besar dan sudah terorganisir Standar Operasional Prosedur (SOP) itu ada. Data menjadi terpusat dan tidak ada perbedaan struktur data yang dapat menyebabkan kesalahan tafsir antar departemen dan divisi. Selain itu standarisasi juga memberikan kemudahan bagi semua usernya dan meminimalisir terjadinya sebuah kesalahan saat proses input data ke sistem.

* Mengurangi kesalahan saat mengidentifikasi kebutuhan komponen

Dengan adanya sistem yang memiliki database master yang terpusat, kesalahan mengidentifikasi yang dulunya dilakukan secara manual dapat dikurangi. Waktu Proses mengidentifikasi kebutuhan menjadi lebih cepat dan mengurangi proses penggunaan daya ingat manusia lagi. Ingatlah bahwa manusia tidak luput dari kesalahan, bisa dikarenakan salah ingat, atau salah tafsir nama komponen dikarenakan bahasa antara orang gudang dengan orang lapangan perakitan berbeda. Semakin cepat divisi perakitan mengidentifikasi komponen yang diperlukan maka proses pemesanan komponen di gudang juga akan semakin cepat.

* + 1. Kebutuhan Perangkat

Kebutuhan perangkat menjadi salah satu hal yang diperlukan di dalam pembangunan sebuah sistem. Seperti yang diketahui, setiap aplikasi memiliki spesifikasi keperluan yang berbeda dan bahkan memiliki beberapa hal yang harus diinstal terlebih dahulu. Pastinya hal ini juga berlaku pada pengembangan yang dilakukan, agar sistem yang dibangun dapat berjalan dengan optimal. Beberapa hal yang harus diperhatikan dalam pembangunan sistem adalah *support* antar perangkat dimulai dari *hardware*, *software*, dan *library* yang diperlukan. Banyak perangkat diluar sana namun jika tidak dipelajari terlebih dahulu, terkadang antar perangkat tidak mendukung satu sama lain. Akibatnya programmer diberikan sebuah pekerjaan tambahan untuk melakukan analisa dan dan menyelesaikan permasalahan dukungan. Terkadang hal ini dapat sangat sulit untuk dilakukan karena harus masuk ke dalam core bahasa program itu sendiri apalagi jika software atau library yang digunakan tidak memiliki dokumentasi yang baik. Untuk itu sangat disarankan pemilihan hardware, software, dan library yang tepat. berikut adalah spesifikasi yang diperlukan dalam pembangunan sistem ini.

1. **Kebutuhan *Hardware* (perangkat Keras)**

Dalam membangun sebuah sistem pastinya diperlukan sebuah alat atau komponen. Hardware merupakan teknologi yang dapat dilihat wujudnya. Pada pembangunan sistem berbasis web komputer menjadi komponen yang paling utama. Hardware disini adalah unit yang nantinya akan paling banyak menerima interaksi fisik dari user. Namun tentu saja agar pembuatan sistem yang akan dibangun dapat berjalan dengan baik dibutuhkan spesifikasi yang memadai dan sesuai. Pemilihan *Hardware* sebaiknya dipilih secara hati – hati. Ingatlah bahwa biaya teknologi itu sendiri tidak murah, dan pemilihan hardware yang melebihi spesifikasi terkadang menjadikan hal yang boros. Dalam pembuatan web diperlukan minimal sebuah komputer yang nantinya dapat bertindak sebagai server dalam melayani kebutuhan website itu sendiri. Spesifikasi minimal yang diperlukan adalah sebagai berikut:

1. CPU (Central Processing Unit) sebagai Server Utama
2. Processor : 3.2 GHz.
3. *Memory* : 32 GB.
4. *Harddisk* : 2 TB.
5. CPU untuk komputer kantor
6. Processor : 2.4 GHz.
7. *Memory* : 8 GB.
8. *Harddisk* : 256 GB.
9. Infrastruktur jaringan antar departemen
10. Kabel
11. *Routers*
12. *Switches*
13. **Kebutuhan *Software* (Perangkat Lunak)**

Perangkat lunak bisa diartikan sebagai sebuah atau beberapa kumpulan yang ditugaskan untuk memberikan dari perintah kepada perangkat keras untuk melakukan suatu fungsi atau tugas tertentu. Tanpa kehadiran dari perangkat lunak, perangkat keras tidak akan mampu bekerja. Sebuah sistem yang dibangun tentu saja memerlukan perangkat lunak, baik itu terdiri dari sebuah *software* maupun lebih dari satu *software*. Adapun kebutuhan Software yang diperlukan untuk merancang aplikasi ini adalah sebagai berikut:

* Sistem Operasi : *Windows 10*
* PHP 8.1
* Mongo DB Community Edition & Mongo DB Compass
* Microsoft SQL Server Management Studio 19
* Laravel 9.x
* Vue 2.x
* Task Scheduler
* Node JS versi 16.x

1. **Kebutuhan *Library***

Tentunya dalam pembangunan program ini, tidak semua dibuat dari awal. Saat ini sudah banyak kumpulan *code* yang beredar di internet, yang sudah pernah ditulis oleh orang lain. Hal ini tentu saja dapat dimanfaatkan pada pembangunan program. *Library* adalah kumpulan beberapa *code* yang dapat memiliki kemiripan fungsi dan biasanya fungsi – fungsinya bisa lebih dari satu. Pada pembuatan program terdapat beberapa library pendukung yang digunakan antara lain :

* Font Awesome
* Datatables.net
* JQuery
* Express
* Axios
* Pinia
* Vue Html to Paper
* Vue-router
* SweetAlert2
* Webpack

VuetifyBAB V  
IMPLEMENTASI

Implementasi akan merelealisasikan desain sistem yang telah kita buat ke dalam wujud nyata. Pada bab ini akan banyak menjelaskan mengenai segmen program yang dijalankan pada masing – masing fitur. Selain itu beberapa segmen program yang mungkin tidak masuk ke dalam fitur karena sifatnya yang terlalu umum namun penting akan dijelaskan. Bagian ini akan menjelaskan bagaimana sekiranya code yang telah dibuat di dalam sistem bekerja. Semua detail terkait program yang dibuat akan dijelaskan.

1. General

General akan menjelaskan beberapa hal yang sekiranya perlu diatur terlebih dahulu. Disini banyak hal terkait settingan program ditulis. Program ini akan berjalan pada hampir di setiap halaman program. Namun perlu diingat program ini tidak akan jalan ketika *code* dari program yang dibuat tidak dipanggil.

* + 1. Koneksi Database

Koneksi diperlukan untuk membuat program yang dibuat dalam melakukan akses ke database. Hal ini harus dilakukan dikarenakan program database dengan yang dibuat terpisah. Ketika kita sudah membuat koneksi, beberapa segmen yang lainnya dapat dengan mudah melakukan *insert*,*update*,dan delete data ke database. Bisa dikatakan koneksi database ini adalah sebuah helper yang membantu programmer dalam memudahkan hal – hal yang berhubungan dengan database.

Segmen Program 5.1 Koneksi

1. 'connections' => [
2. 'mongodb' => [
3. 'driver' => 'mongodb',
4. 'host' => env('DB\_HOST', '127.0.0.1'),
5. 'port' => env('DB\_PORT', 27017),
6. 'database' => env('DB\_DATABASE', 'homestead'),
7. 'username' => env('DB\_USERNAME', 'homestead'),
8. 'password' => env('DB\_PASSWORD', 'secret'),
9. 'options' => [
10. 'database'=>env('DB\_AUTHENTICATION\_DATABASE' ,'admin'),
11. ],
12. ],
13. ],

Line 1 hingga ke 13 adalah segmen code untuk sebuah koneksi. Untuk mendefinisikan sebuah koneksi line 2 hinnga 12 diperlukan. Nantinya koneksi ke dua akan diletakkan di dalam kurung ‘*connection*’. Line 2 adalah nama koneksinya. Line 3 adalah driver yang digunakan bisa mysql, sqlserver, atau mongodb dalam hal ini. Line 4 adalah untuk memberikan koneksi ini di host di mana, jika tidak di setting di *environment* maka defaultnya adalah nembak ke *localhost*. Line 5 untuk memberikan port yang digunakan. Line 6 adalah nama database yang digunakan. Line 7 hingga 8 adalah untuk memberikan username dan password saat mau mengakses database. Line 10 adalah untuk autentikasi databasenya.

* + 1. Authentikasi dan persist data login

Autentikasi dipastikan digunakan untuk memastikan user yang login saat ini. Data user yang berhasil login harus disimpan ke dalam sebuah cookie sehingga user ketika berpindah halaman atau tidak sengaja menutup browsernya tetap login. Hal ini digunakan untuk memberikan *user experience* yang lebih baik kepada user. Untuk memastikan tidak terjadi perulangan ke menu login. Penyimpanan pada program ini menggunakan library Pinia, untuk melakukan hal yang telah disebutkan diatas.

Segmen Program 5.2 Auth Store Pinia

1. export const useAuth = defineStore("auth", {
2. state: () => ({
3. authuser: null,
4. authErrors: []
5. }),
6. getters: {
7. user: (state) => state.authuser,
8. errors: (state) => state.authErrors
9. },
10. actions: {
11. async getUser() {
12. this.authuser = this.authuser
13. },
14. async handleLogin(data) {
15. this.authErrors = [];
16. try {
17. await axios.post('/api/login', data).then((response) => {
18. if (response.data[0] == null) {
19. this.authuser = null
20. }
21. else {
22. this.authuser = response.data[0].user
23. }
24. });
25. } catch (error) {}
26. },
27. async handleLogout() {
28. try {
29. this.authuser = null;
30. } catch (error) {
31. }
32. },
33. },
34. persist: true
35. })

Line 1 adalah nama storenya. Line 2 hingga line 5 menjelaskan model data dari store yang dibuat. Line 6 hingga 8 adalah *getter* seperti pada java yang ditujukan untuk mengambil data. Line 11 hingga 33 adalah kumpulan aksi atau perintah yang dapat dilakukan pada store ini. Line 11 hingga 13 berfungsi untuk mendapatkan user sekarang. Line 14 hingga 26 berfungsi adalah fungsi untuk melakukan login ke sistem. Line 27 hingga 33 berfungsi untuk proses logout user. Line 34 adalah untuk membuat data user yang login sekarang tetap tersimpan selama user belum menekan tombol logout.

* + 1. Tarik Data Database Perusahaan (SQL Server)

Perbedaan database mongodb dengan yang perusahaan miliki sekarang, mengharuskan dilakukan penarikan data. Saat ini perusahaan menggunakan database berbasis SQL yang bernama Microsoft SQL Server. Oleh karena itu penarikan data ini tidak bisa langsung semerta merta dipakai. Pada tahapan ini terjadi beberapa tahapan yaitu proses menarik semua data menggunakan SQL query kemudian melakukan konversi ke json untuk disimpan ke mongodb.

Segmen Program 5.3 Tarik Data SPK

1. $datas=DB::connection('sqlsrv')-> table('SURATPERINTAHKERJA')
2. ->join('SPECIFICATION', 'SPECIFICATION.SPK Number', '=', 'SURATPERINTAHKERJA.SPK Number')
3. ->get();
4. foreach ($datas as $data) {
5. $newdata = SPK::create([
6. "NOSPK" => trim($data->{'SPK Number'}),
7. "Tipe"=>trim($data->{'SPK Type'}),
8. "AirSuspensi"=>trim($data->{'Air Suspensi'}),
9. "Semi\_Monocoque"=>trim($data->{'Semi Monocoque'}),
10. "No\_Rangka"=>trim($data->{'No Rangka'}),
11. "No\_Mesin"=>trim($data->{'No Mesin'}),
12. "parameter" => [
13. "ModelMobil" => trim($data->{'Merk'}),
14. "TipeMobil" => trim($data->{'Type'}),
15. "TinggiMobil" => "",
16. "newparameter" => [
17. [
18. "Newparam" => trim($data->{'User Defined'}),
19. "Component" => [trim($data->{'User Defined Desc'})],
20. ],
21. ]
22. ]
23. ]);
24. }

Line 1 hingga 3 adalah perintah untuk mendapatkan semua data SPK milik perusahaan. Baris 4 hingga 24 adalah untuk proses melakukan looping baris data yang didapat. Setelah didatpatkan dibuatlah Model dalam hal ini SPK untuk proses insert data dimulai dari line 5 hingga 23. Segmen program ini memiliki cara yang sama dengan pengecekkan data yang sudah ada di database master milik mongodb. Perbedaannya hanya terletak pada jumlah banyak if untuk melakukan proses pengecekkan data.

* + 1. Routing

Routing adalah kumpulan rute pada website yang mana akan menjadi tempat path website didefinisikan. Karena penggunaan vue sebagai frontendnya maka route juga harus menggunakan cara vue. Disini diperlukan sebuah library bernama vue-router untuk melakukan routing. Pada bagian ini juga middleware akan didefinisikan untuk membatasi akses web user supaya tidak dapat mengakses path yang bukan haknya.

Segmen Program 5.4 Routing

1. import Vue from 'vue';
2. import VueRouter from 'vue-router'
3. import { useAuth } from '../../Stores/Auth'
4. Vue.use(VueRouter)
5. const routes = [{
6. name: 'Home',
7. path: '/',
8. component:()=> import("../components/Pages/Home/Home.vue"),
9. meta: {
10. guestPageAccess: true,
11. }
12. },
13. {
14. name: 'Register', // register user baru
15. path: '/user/create',
16. component:()=>import("../components/Pages/Register/Register.vue"),
17. meta: {
18. guestPageAccess: false,
19. levelAccess: 'Super Admin Role'
20. }
21. },
22. ]

Line 1 hingga 4 adalah inisialisasi untuk mengimpor library. Line 5 hingga 22 adalah tempat *route* ditaruh. Line 5 hingga 12 adalah contoh pembuatan sebuah route begitu juga line 13 hingga 21. Baris 6 adalah untuk memberikan alias path sehingga dapat dengan mudah dipanggil. Baris 7 adalah *pathnya*. Baris 8 adalah tempat untuk mengimpor komponen yang akan diload pada halaman *path*.baris 11 hingga 11 adalah akses level yang akan berhubungan dengan middleware pada segmen program 5.5. Meta adalah fungsi yang sifatnya opsional ketika tidak di declare maka semua user yang belum login bisa menagkses path yang bersangkutan.

Segmen Program 5.5 Middleware

1. router.beforeEach((to, from, next) => {
2. const store = useAuth();
3. if (!to.meta.guestPageAccess) {
4. store.getUser();
5. const routes = [
6. if (store.user) {
7. if (store.user.account\_privileges.title == "Super Admin Role") {
8. next()
9. }
10. else if (store.user.account\_privileges.title == "Admin Role") {
11. if (to.meta.levelAccess == "Admin Role" || to.meta.levelAccess == "Staff Role") {
12. next()
13. }
14. }
15. else if (store.user.account\_privileges.title == "Staff Role") {
16. if (to.meta.levelAccess == "Staff Role") {
17. next()
18. }
19. }else{
20. next({
21. name: 'NotFound'
22. })
23. }
24. }
25. else {
26. next({
27. name: 'Login'
28. })}
29. }
30. else {
31. next()
32. }
33. })

Line 1 adalah mendefinisikan fungsi middleware. Line 2 adalah untuk mengakses store yang ada di pinia. Line 3 hingga 33 adalah untuk mendeteksi apakah user sedang login atau tidak. Line 4 berfungsi untuk mendapatkan role dari user. Line 5 berfungsi untuk memberikan rute yang dapat dipilih oleh user berdasarkan rolenya. Line 6 hingga 9 adalah fungsi untuk mengatur *path* user dengan role super admin. Line 10 sampai 14 adalah fungsi untuk mengatur *path* user yang memiliki role admin. Line 15 hingga 24 adalah untuk mengatur path user dengan role staff. Line 25 hingga 29 adalah untuk fungsi untuk melakukan *redirect* ke path login ketika user mengakses path yang bukan rolenya. Dan line 30 hingga 32 adalah untuk mengatur path user yang belum login.

* + 1. Helper Konversi Waktu

Konversi waktu diperlukan untuk membuat format tanggal dan waktu sesuai dengan keinginan. Di mongodb secara default penyimpanan yang berhubungan dengan datetime ditulis dalam bentuk model data UTC miliseconds sejak epoch. Untuk menghindari proses melakukan penulisan code yang berulang dalam melakukan konversi datetime dibuatlah sebuah fungsi global yang membantu.

Segmen Program 5.6 Konversi Format Waktu

1. import { format, parseISO } from "date-fns"
2. import { id } from 'date-fns/locale';
3. export default {
4. methods: {
5. converttime(date) {
6. const str = format(
7. new Date(date),
8. 'dd-MMMM-yyyy HH:mm:ss', { locale: id }
9. );
10. return str;
11. },
12. }
13. };

Line 1 dan 2 adalah proses mengimpor library yang diperlukan. Line 3 adalah proses mendaklarasikan agar segmen program ini dapat dipakai secara global. Selanjutnya line 4 adalah tempat meletakkan methodnya. Line 5 sampai 11 adalah satu kesatuan fungsi. Dimana pada line 6 hingga 9 dilakukan proses konversi data ke datetime local indonesia.

1. Master

Untuk melakukan fitur konversi nantinya, fitur master harus dilakukan terlebih dahulu. Disini master sesuai yang telah di sebutkan pada pembagian role sebelumnya, pengirisan master hanya dapat dilakukan oleh dua role yaitu Super admin dan admin. Proses penambahan master disini memiliki 2 macam form yang terbagi di sebelah kiri dan kanan untuk itu dalam melakukan proses input kedua form harus diisi. Database master dapat di tambah , edit, dan dihapus (*soft delete*) seperti data pada umumnya. Untuk edit dan hapus tidak dibahas pada segmen program nantinya dikarenakan fungsi edit memiliki kesamaan yang cukup banyak dengan fungsi tambah, perbedaannya hanya terletak pada prosedur langkah awal, yang mana data yang mau di update harus dicari terlebih dahulu kemudian di tampilkan. Lalu data akan diubah seperti pengecekkan validasi master sebelum memasukkan data. Untuk fungsi delete juga tidak dibahas karena programnya yang sangat singkat, yaitu mencari id dari data yang mau dihapus kemudian menulis sebaris program untuk delete

Segmen Program 5.7 Format Input

1. $param = $request->dataparam;
2. $kit = $request->datakit;
3. $i = 0;
4. foreach ($param['TipeMobil'] as $item) {
5. $param['TipeMobil'][$i] = ucwords($item);
6. $i++;
7. }
8. $i = 0;
9. foreach ($param['ModelMobil'] as $item) {
10. $param['ModelMobil'][$i] = ucwords($item);
11. $i++;
12. }
13. $i = 0;
14. foreach ($param['TinggiMobil'] as $item) {
15. $param['TinggiMobil'][$i] = ucwords($item);
16. $i++;
17. }

Segmen ini berfungsi untuk mengubah format input yang dilakukan oleh user. Fungsinya adalah membuat setiap kata pada huruf pertama selalu berhuruf besar diikuti oleh huruf kecil pada kata tersebut.pada baris 1 dan 2 berfungsi untuk mengambil inputan user. Pada baris 3 hingga 7 adalah proses melakukan looping untuk mengganti setiap field inputan tipe mobil. Baris 8 hingga 12 berperan untuk mengganti format di field model mobil secara berulang . dan pada baris 13 hingga 17 berperan untuk megganti format pada field tinggi mobil.

Segmen Program 5.8 Cek Inputan Kosong

1. function FungsicekKosong(array $cek)
2. {
3. if (count($cek) == 0) {
4. return true;
5. }
6. }
7. $paramkosong = false;
8. if (!$paramkosong) {
9. $paramkosong = FungsicekKosong($param['TipeMobil']);
10. }
11. if (!$paramkosong) {
12. $paramkosong = FungsicekKosong($param['ModelMobil']);
13. }
14. if (!$paramkosong) {
15. $paramkosong = FungsicekKosong($param['TinggiMobil']);
16. }
17. if (!$paramkosong) {
18. $paramkosong = FungsicekKosong($param['Departemen']);
19. }
20. if (!$paramkosong) {
21. $paramkosong = FungsicekKosong($param['Stall']);
22. }
23. if ($paramkosong) {
24. return response()->json([
25. "success" => true,
26. "statuscode" => 401,
27. ]);
28. }
29. $paramtambahankosong = false;
30. foreach ($param['NewParameter'] as $newparam) {
31. if ($newparam['Newparam'] == "" || $newparam['Newparam'] == null) {
32. $paramtambahankosong = true;
33. break;
34. }
35. foreach ($newparam['Component'] as $komponen) {
36. if ($komponen == "" || $komponen == null) {
37. $paramtambahankosong = true;
38. break;
39. }
40. }
41. if ($paramtambahankosong) {
42. break;
43. }
44. }
45. if ($paramtambahankosong) {
46. return response()->json([
47. "success" => true,
48. "statuscode" => 403,
49. ]);

Line 1 hingga 6 berfungsi untuk membuat fungsi kecil dalam melakukan proses untuk mengecek apakah terdapat inputan yang kosong. Nantinya hal ini akan digunakan secara berulang untuk mengecek beberapa parameter. Line 18 adalah penanda untuk memberikan tanda apakah nantinya inputan yang diberikan oleh user ada yang kosong atau terisi semua. Line 19 hingga 21 mengecek parameter tipe mobil terisi lengkap atau tidak. Line 22 hingga 24 mengecek parameter model mobil terisi lengkap atau tidak. Line 25 hingga 27 mengecek parameter tinggi mobil terisi lengkap atau tidak. Line 28 hingga 30 mengecek parameter departemen terisi lengkap atau tidak. Line 31 hingga 33 mengecek parameter stall terisi lengkap atau tidak. Line 34 hingga 39 berfungsi untuk memberikan response gagal jika ternyata terdapat parameter yang kosong.

Untuk pengisian pada parameter baru pengecekkan dilakukan dimulai dari line 40 hingga 60. Line 42 berfungsi untuk melakukan looping pada area inoputan new parameter. Line 42 hingga 45 adalah mengecek apakah nama dari parameter yang baru kosong atau tidak diisi. Line 46 hingga 49 adalah fungsi tambahan untuk pengecekkan isi dari parameter yang baru. Line 52 hingga 54 berfungsi untuk menghentikan proses looping ketika ternyata inputan dari parameter baru ada yang kosong. Line 56 hingga 60 berfungsi untuk mengembalikan response gagal.

Segmen Program 5.9 Fungsi Inputan kembar

1. function fungsiceksama(array $cek)
2. {
3. if (count($cek) !== count(array\_unique($cek))) {
4. return true;
5. }
6. }
7. $paramsama = false;
8. if (!$paramsama) {
9. $paramsama = fungsiceksama($param['TipeMobil']);
10. }
11. if (!$paramsama) {
12. $paramsama = fungsiceksama($param['ModelMobil']);
13. }
14. if (!$paramsama) {
15. $paramsama = fungsiceksama($param['TinggiMobil']);
16. }
17. if (!$paramsama) {
18. $paramsama = fungsiceksama($param['Departemen']);
19. }
20. if (!$paramsama) {
21. $paramsama = fungsiceksama($param['Stall']);
22. }
23. if (!$paramsama) {
24. $paramsama = fungsiceksama($param['Stock']);
25. }
26. if ($paramsama) {
27. return response()->json([
28. "success" => true,
29. "statuscode" => 402,
30. ]);
31. }
32. $paramtambahansama = false;
33. $judulparamtambahankembar = array();
34. foreach ($param['NewParameter'] as $newparam) {
35. array\_push($judulparamtambahankembar, $newparam['Newparam']);
36. if(count($newparam['Component'])!==count(array\_unique( $newparam['Component']))) {
37. $paramtambahansama = true;
38. break;
39. }
40. }
41. if(count($judulparamtambahankembar)!== count(array\_unique($judulparamtambahankembar))) {
42. $paramtambahansama = true;
43. }
44. if ($paramtambahansama) {
45. return response()->json([
46. "success" => true,
47. "statuscode" => 404,
48. ]);
49. }
50. function fungsicekparameterterdaftar(array $array1, array $array2){
51. $jumlahkesamaan = 0;
52. foreach ($array2 as $isiarray2) {
53. foreach ($array1 as $isiarray1) {
54. if(strtoupper($isiarray1)==strtoupper($isiarray2)){
55. $jumlahkesamaan++;
56. break;
57. }
58. }
59. }
60. if ($jumlahkesamaan == count($array2)) {
61. return true;
62. }
63. }

Line 1 sampai 6 berfungsi untuk membuat sebuah fungsi untuk mengecek apakah terdapat inputan yang kembar. Line 7 untuk memberikan tanda nantinya apakah terdapat inputan yang kembar. Line 8 hingga 10 berfungsi untuk mengecek inputan kembar pada parameter tipe mobil. Line 11 sampai 13 mengecek inputan kembar pada parameter model mobil. Line 14 sampai 16 mengecek inputan kembar pada parameter tinggi mobil. Line 17 sampai 19 mengecek inputan kembar pada parameter departemen. Line 20 sampai 22 mengecek inputan kembar pada parameter stall. Line 23 sampai 25 mengecek inputan kembar pada stock. Line 26 sampai 31 berfungsi untuk mengembalikan response ke user.

Line 32 berfungsi untuk penanda apakah terdapat inputan kembar pada parameter baru. Line 33 merupakan inisialisasi awal untuk membuat array kosong. Line 34 sampai 63 berfungsi untuk melakukan looping, mengecek setiap inputan pada parameter baru apakah ada inputan yang kembar. Pengecekkan kembar yang dilakukan adalah nama parameter yang baru dan isi dari parameter yang baru. dan line 44 hingga 49 apabila ditemukan kesamaan response gagal langsung diberikan.

Segmen Program 5.10 Pengecekkan Data Input Dengan Data Master

1. $allmaster = Master::all();
2. foreach ($allmaster as $master) {
3. $saved = $master->Parameter;
4. $cekTipeMobil = false;
5. $cekModelMobil = false;
6. $cekTinggiMobil = false;
7. $cekDepartemen = false;
8. $cekstall = false;
9. $cekStock = false;
10. $cekAdditionaParameter = false;
11. $cekTipeMobil=fungsicekparameterterdaftar($saved['TipeMobil'], $param['TipeMobil']);
12. $cekModelMobil=fungsicekparameterterdaftar($saved['ModelMobil'], $param['ModelMobil']);
13. $cekTinggiMobil=fungsicekparameterterdaftar($saved['TinggiMobil'], $param['TinggiMobil']);
14. $cekDepartemen=fungsicekparameterterdaftar($saved['Departemen'], $param['Departemen']);
15. $cekstall=fungsicekparameterterdaftar($saved['Stall'],$param['Stall']);
16. if(count($param['Stock'])==0&&count($saved['Stock']) == 0) {
17. $cekStock = true;
18. }else{
19. $cekStock=fungsicekparameterterdaftar($saved['Stock'], $param['Stock']);
20. }
21. if(count($param['NewParameter'])==0&&count($saved['NewParameter']) == 0){
22. $cekAdditionaParameter = true;
23. }
24. if ($cekTipeMobil && $cekModelMobil && $cekTinggiMobil && $cekDepartemen&& $cekStock && $cekstall && $cekAdditionaParameter) {
25. return response()->json([
26. "success" => true,
27. "statuscode" => 406,
28. ]);
29. }}

Line 1 bertujuan untuk mengambil semua data yang ada pada master. Kemudian pada line 2 akan dilakukan proses looping untuk mengecek apakah data yang akan diinput sudah terdaftar apa belum. Baris 3 adalah proses mengambil parameter pada database master. Line 4 sampai 10 berfungsi sebagai penanda apakah parameter yang mau diinputkan sudah ada atau belum. Kemudian pada line 11 hingga 23 proses pengecekkan parameter secara berurutan mulai dilakukan. Dimulau dari Tipe Mobil, Model Mobil, Tinggi Mobil, Departemen, Stock, dan new parameter. Lalu pada Line 24 hingga 29 berfungsi untuk mengembalikan response gagal jika memang terdapat parameter yang mau diinput sudah terdaftar.

Segmen Program 5.11 Pembuatan Hasil Result

1. $listmasterkit = Masterkit::all();
2. $masterkit = new Masterkit();
3. foreach ($listmasterkit as $kit) {
4. if(strtoupper($kit->kode\_kit) ==strtoupper($request->param)){
5. $masterkit = $kit;
6. return response()->json([
7. "success" => true,
8. "statuscode" => 201,
9. "data" =>  strtoupper($request->param),
10. "message" => $masterkit,
11. "result" => $masterkit,
12. "hasil" => $kitsudahada
13. ]);
14. }
15. }
16. if (response.data.statuscode == 201) {
17. let datanamakit = response.data.result.nama\_kit
18. let dataKodeKit = response.data.result.kode\_kit
19. let dataisikit = response.data.result.komponen
20. let kembar = false
21. this.Result.forEach(element => {
22. if(element.NamaKit.toUpperCase()==datanamakit.toUpperCase()) {
23. kembar = true
24. }
25. });
26. }
27. if (kembar) {
28. this.$swal({
29. title: 'Kode Kit ' + this.InputKodeKit + ' sudah ada',
30. icon: 'error'
31. });
32. }

Line 1 adalah proses menarik semua data kit yang ada. Line 2 berfungsi untuk membuat sebuah object kosong. Kemudian pada line 3 hingga 14 adalah proses looping yang mana mencari kecocokkan kode kit antara yang ada pada data kit dengan kode yang diinput oleh user. Ketika terjadi kecocokkan nama kode barulah server mengembalikkan response sukses.

Line 16 kebawah adalah bahasa javascript. Line 16 berfungsi untuk melakukan proses cek status code dari response. Ketika cocok maka fungsi dibawahnya akan dijalankan. Selanjutnya pada line 17 sampai 25 adalah proses melakukan cek apakah kode kit yang dipakai sudah pernah dipakai. Jika ternyata kembar maka akan muncul peringatan yang menunjjukan error bahwa pemakaian kode kit ini sudah pernah dipakai line 27 sampai 32.

Segmen Program 5.12 Pengecekkan result

1. $kosongkit = false;
2. if (count($kit) > 0) {
3. foreach ($kit as $subkit) {
4. $subkit["NamaKit"] = ucwords($subkit["NamaKit"]);
5. if (count($subkit['IsiKit']) > 0) {
6. $j = 0;
7. foreach ($subkit['IsiKit'] as $komponen) {
8. $komponen[$j]['nama\_komponen']=ucwords($komponen['nama\_komponen']);
9. $j++;
10. }
11. }
12. foreach ($subkit['IsiKit'] as $komponen) {
13. if ($komponen['nama\_komponen'] == null || $komponen['qty'] == null) {
14. $kosongkit = true;
15. break;
16. }
17. }
18. }
19. } else {
20. return response()->json([
21. "success" => true,
22. "statuscode" => 405,
23. ]);
24. }
25. if ($kosongkit) {
26. return response()->json([
27. "success" => true,
28. "statuscode" => 408,
29. ]);
30. }

Line 1 berfungsi sebagai penanda apakah kit ada atau tidak. Selanjutnya dilakukan proses pengecekkan apakah kit ada atau kosong termasuk isi dari kit tersebut Line 2 sampai 18. Line 19 sampai 24 adalah pengembalian response gagal jika ternyata ada isi dari komponen yang kosong. Line 25 sampai 30 adalah pengembalian response gagal untuk kit yang kosong sama sekali.

Segmen Program 5.13 Input Data Master

1. $Newmaster = Master::create([
2. "Kit" => $kit,
3. "Parameter" => $param,
4. ]);
5. return response()->json([
6. "success" => true,
7. "statuscode" => 200,
8. "kit" => $kit
9. ]);

Segmen ini berfungsi untuk melakukan penambahan data pada database master setelah berhasil melalui semua pengecekkan. Line 1 sampai 4 adalah proses penambahan data baru. Line 5 sampai 9 adalah bentuk pengembalian response berhasil. Line 7 status code menunjukkan kode keberhasilan.

1. Konversi

Konversi adalah fitur yang mencocokkan data parameter yang berada pada SPK dengan data parameter yang telah diisikan di master. Nantinya SPK yang berhasil dalam proses kecocokkan akan menampilkan data komponen – komponen yang diperlukan. Konversi yang tidak berhasil akan menampilkan data komponen kosong dan sekiranya permasalahan dari alasan mengapa tidak lolos pada proses cek. Hal ini akan ditampilkan ketika user menekan tombol problem pada SPK yang tidak lolos.

Segmen Program 5.14 Validasi Input SPK

1. if (this.SPKfield == "" || this.stall == "" || this.stall == 0 || this.NamaStall == "" || this.Departemen == "") {
2. this.$swal({
3. title: 'pengisian SPK tidak Valid',
4. icon: 'error'
5. });
6. }

Segmen ini bertugas untuk mengecek apakah input SPK yang mau dicek sudah memenuhi kriteria. Pada line 1 hingga 6 adalah proses pengecekkan apakah ada field yang kosong. Di line 2 hingga 5 adalah proses memunculkan penringatan dialog box. Dan di line 3 adalah isi judul pesan error tersebut. Swal ini adalah fungsi yang berasal dari library Sweetalert.

Segmen Program 5.15 Penambahan Input SPK

1. $newdata = SavedConversionResult::create([
2. "NOSPK" => $spk->NOSPK,
3. "stall" => $stall,
4. "namastall" => $namastall,
5. "Departemen" => $Departemen,
6. "checked" => false,
7. "status" => "Pending",
8. "parameter" => $parameter,
9. "created\_at" => Carbon::now()->format('Y-m-d H:i:s'),
10. "updated\_at" => Carbon::now()->format('Y-m-d H:i:s'),
11. ]);
12. return response()->json([
13. "success" => true,
14. "status" => 200,
15. "spk" => $spk,
16. "namastall" => $namastall,
17. "Departemen" => $Departemen,
18. "newdata" => $parameter,
19. "allsaved" => $allsaved
20. ]);

Segmen ini bertugas untuk mengecek apakah input SPK yang mau dicek sudah memenuhi kriteria. Pada line 1 hingga 6 adalah proses pengecekkan apakah ada field yang kosong. Di line 2 hingga 5 adalah proses memunculkan penringatan dialog box. Dan di line 3 adalah isi judul pesan error tersebut. Swal ini adalah fungsi yang berasal dari library Sweetalert.

Segmen Program 5.16 Konversi SPK

1. foreach ($master as $item2) {
2. $data = SPK::where('NOSPK', $item1["NOSPK"])->first();
3. $ModelMobilterdaftar = false;
4. $TinggiMobilterdaftar = false;
5. $TipeMobilTerdaftar = false;
6. $DepartemenTerdaftar = false;
7. $StallTerdaftar = false;
8. $newparameterTerdaftar = false;
9. foreach ($item2["Parameter"]["ModelMobil"] as $subitem2) {
10. if(strtoupper($subitem2)==strtoupper($data["parameter"]["ModelMobil"])){
11. $ModelMobilterdaftar = true;
12. $errorModelMobil = false;
13. break;
14. }
15. }
16. foreach ($item2["Parameter"]["TinggiMobil"] as $subitem2) {
17. if(strtoupper($subitem2)==strtoupper($data["parameter"]["TinggiMobil"])) {
18. $TinggiMobilterdaftar = true;
19. $errorTinggiMobil = false;
20. break;
21. }
22. }
23. if($ModelMobilterdaftar && $TinggiMobilterdaftar && $TipeMobilTerdaftar && $DepartemenTerdaftar && $StallTerdaftar && $newparameterTerdaftar) {
24. array\_push($result, $item2["Kit"]);
25. $i++;
26. }
27. if ($i > 0) {
28. array\_push($results, [
29. 'kit' => $result,
30. 'NoSPK' => $item1->NOSPK,
31. ]);
32. $item1["status"] = "berhasil";
33. $item1["kit"] = $results;
34. $item1->save();
35. } else {
36. $item1["status"] = "Pending";
37. $item1->save();
38. }

Line 1 adalah proses melakukan looping data master untuk di crosscheck dengan parameter SPK.di line 2 adalah proses pengambilan data SPK. Line 3 sampai 8 adalah pemberian tanda yang nanti akan berguna untuk memberikan informasi apakah parameter SPK *matching* dengan parameter pada database master. Kemudian di baris 9 sampai 15 adalah proses pengecekkan parameter model mobil. Lalu pada baris 16 sampai 22 adalah proses pengecekkan untuk parameter tinggi mobil. Seharusnya masih ada untuk pengecekkan yang lain hanya saja segmen akan menjadi kepanjangan karena secara garis besar fungsi yang dijalankan sama.

Line 23 sampai 26 adalah proses pengelompokkan data kit yang berhasil didapatkan. Kemudian pada baris 27 hingga 31 adalah proses merapikan data sebelum dilakukan proses save pada database. Kemudian pada baris 32 sampai 34 adalah fungsi untuk menyimpan data yang telah diatur tadi. Lalu pada line 35 sampai 38 adalah untuk menyimpan perubahan status untuk data SPK yang tidak berhasil ditemukan kitnya. Nantinya data yang tidak ditemukan tadi akan membuat munculnya tobol problem pada halaman input SPK yang mana bisa dicek untuk melihat pesan errornya.

1. Datatable

Datatable adalah fitur yang diperlukan pada kebanyakan web yang banyak berhubungan dengan laporan. Disini datatable berperan dalam menampilkan data yang banyak dalam bentuk yang rapi. Selain itu beberapa fungsi datatable disini juga adalah sebagai media untuk melakukan aksi terhadap data seperti edit, insert, dan delete. Datatable juga memiliki fitur untuk melakukan filter tanpa merusak data aslinya.

Segmen Program 5.17 Inisialisasi Datatable

1. headerstable: [
2. {text: 'Nomor SPK',
3. align: 'start',
4. sortable: false,
5. value: 'NOSPK',
6. class: "title text-uppercase font-weight-black black--text light-blue lighten-5"
7. },
8. { text: 'Nama Stall', value: 'namastall', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
9. { text: 'Stall', value: 'stall', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
10. { text: 'Departemen', value: 'Departemen', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
11. { text: 'Status', value: 'status', width: '150px', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
12. { text: 'Waktu Update Terakhir', width: '250px', value: 'updated\_at', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
13. { text: 'Action', value: 'actions', width: '300px', class: "title text-uppercase font-weight-black black--text light-blue lighten-5" },
14. ],

Line 1 adalah line pembuka untuk memberikan judul tabel. Line 2 sampai 7 adalah fungsi untuk membuat sebuah kolom. Line 3 adalah untuk memberikan alignment pada tabel. Line 4 berfungsi untuk memberikan fitur pada kolom yang bersangkutan apakah bisa dilakukan sortir atau tidak. Line 5 adalah tempat isi datanya nanti. Line 6 adalah untuk memberikan css inline pada html. Sama halnya pada line 8, 9, 10, 11, 12, dan 13 masing masing line adalah untuk membuat sebuah kolom.

Segmen Program 5.18 Pengisian Datatable

1. Await axios.post('/api/getdatatable',

{Role:this.authStore.user.account\_privileges.title, Departemen:this.authStore.user.account\_privileges.account\_dept}).then((response) => {

1. this.datatable = []
2. this.datatable = response.data.reverse()
3. this.datatable.forEach(element => {
4. element["updated\_at"]
5. this.converttime(element["updated\_at"])
6. });
7. })

Line 1 berfungsi untuk melakukan request untuk mendapatkan data yang akan diisi pada datatable. Selanjutnya pada baris 16 datatable dikosongkan terlebih dahulu sebelum diisi. Line 17 berfungsi untuk membalik data dari posisi baru ke lama sekaligus melakukan pengisian pada datatable. Selanjutnya pada line 19 pada masing masing baris yang berasal dari data object yang sama ditambahkan data updated\_at. Kemudian pada baris 20 format data diubah. Data datatable akan secara otomatis melakukan update data sesuai pada value this.datatable.

1. Excel

Excel adalah kebutuhan yang sudah tidak dapat terpisahkan lagi. Excel sendiri sudah banyak dipakai di perkantoran. Pada program ini nantinya excel dipakai untuk mendownload data table yang telah menampilkan komponen – komponen SPK. Proses download akan dimulai ketika user menekan tombol download excel.

Segmen Program 5.19 Pengisian Datatable

1. import JsonExcel from "vue-json-excel"
2. json\_fields: {
3. "NO SPK": "NoSPK",
4. "Kode Kit": "kode",
5. "Nama Kit": "namakit",
6. "Nama Komponen": "namakomponen",
7. "Kebutuhan": "Qty",
8. "Siteid": "siteID",
9. "Dari Rak": "Dari",
10. "Ke Rak": "Kerak",
11. },
12. <JsonExcel class="btn btn-primary"
13. :data="datatable" :fields="json\_fields"
14. worksheet="My Worksheet" name="filename.xls" style="margin-right: 20px;">
15. Download Excel <font-awesome-icon icon="fa-solid fa-download" />
16. </JsonExcel>

Line 1 adalah fungsi untuk melakukan impor fungsi Jsonexcel dari library vue-json-excel. Line 12 bergungsi untuk melakukan menampilkan tombol download pada html. Nanti tombol bisa diberikan css inline. Di line 13 data bertugas sebagai penyedia data dan fields bertugas untuk menyiapkan format data yang akan didownloa. Lalu di line 14 worksheet bertugas untuk memberikan nama sheet pada excel, lalu name untuk nama file excelnya. Line 15 bertugas untuk memberikan icon download. Line 16 adalah tutup tag htmlnya.

1. Print

Print adalah fitur yang tidak kalah penting. Bagi perkantoran fitur ini hal yang cukup penting. Penggunaannya sangatlah sederhana hanya dengan klik tombol user akan diarahkan ke halaman untuk melakukan proses print. Tidak ada yang special dari print disini karena sudah jelas dengan fungsinya.

Segmen Program 5.20 Print

1. <div id="printMe">
2. <div id="image"></div>
3. <v-data-table dense :headers="headerstable" :items="datatable" :items-per-page="30" :search="search"
4. class="elevation-1 font-weight-bold">
5. <template v-slot:top>
6. <v-toolbar flat>
7. <v-toolbar-title>List daftar komponen</v-toolbar-title>
8. <h5 class="tanggal">tanggal {{ new Date().toLocaleString() }}</h5>
9. </v-toolbar>
10. </template>
11. </v-data-table>
12. </div>
13. const options = {
14. styles: [
15. '/css/print.css' // <- inject here
16. ]
17. }
18. async print() {
19. $("#image").append(`<img src='/images/Logo\_Adi\_Putro.svg' alt='' srcset=''>`)
20. $(".v-data-footer\_\_select").html('')
21. $(".v-data-footer\_\_pagination").html('')
22. $(".v-data-footer\_\_icons-before").html('')
23. $(".v-data-footer\_\_icons-after").html('')
24. await this.$htmlToPaper('printMe', options);
25. $("#image").html(``);
26. },

Line 1 hingga 12 adalah html untuk memasang id print pada data table. Hal ini harus dilakukan untuk menandai area pada html yang mau di print. Line 13 sampai 17 adalah fungsi untuk memberikan css pada halaman print nantinya. Ini diperuntukkan untuk memberikan kerapian. Kemudian di line 18 sampai 26 adalah fungsi untuk melakukan fitur print, nanti ada beberapa hal yang akan dihilangkan dan ada yang ditambah, seperti memberikan logo gambar perusahaan.

1. Halaman Setting

Halaman Setting berfungsi bagi super admin untuk melakukan managemen websitenya. Dimulai dari proses manajemen akun, manajemen departemen, dan manajemen stall. Namun proses codingnya dibuat dalam satu kesatuan karena memiliki kemiripan walaupun terpisahkan oleh path yang berbeda. Selain super admin halaman setting tidak dapat diakses.

Segmen Program 5.21 Ganti tampilan

1. import User from './User.vue';
2. import Departemen from './Departemen.vue';
3. import Stall from './Stall.vue';
4. <v-container fill-height fluid>
5. <v-row justify="center">
6. <v-btn-toggle >
7. <v-btn

@click="changetoggle('Account')" :class="[toggleAccount=='Account'? 'aa' : 'white']">

1. <v-icon>Account</v-icon>
2. </v-btn>
3. <v-btn @click="changetoggle('Departemen')"
4. :class="[toggleAccount=='Departemen'?'aa':'white']>
5. <v-icon>Departemen</v-icon>
6. </v-btn>
7. <v-btn @click="changetoggle('Stall')":class="[toggleAccount=='Stall'?'aa':'white']">
8. <v-icon>Stall</v-icon>
9. </v-btn>
10. </v-btn-toggle>
11. <div v-if="toggleAccount === 'Account'">
12. <User />
13. </div>
14. <div v-if="toggleAccount === 'Departemen'">
15. <Departemen />
16. </div>
17. <div v-if="toggleAccount === 'Stall'">
18. <Stall />
19. </div>
20. </v-row>
21. </v-container>

Line 1 hingga 3 adalah proses melakukan impor komponen. Line 4 adalah tag pembuka untuk mengisi. Line 5 adalah untuk membuat tampilan memiliki align justify. Line 6 sampai 17 adalah tempat untuk memberikan tombol yang nantinya akan digunakan untuk melakukan routing Line 18 sampai 28 adalah tempat untuk melakukan aktivasi routing. Jika true maka tampilkan komponen ini.

1. Insert, Update dan Delete

Insert update dan delete pastinya diperlukan dalam proses melakukan update data pada database. di sub bab ini akan menjelaskan sekiranya gambaran ringkas cara melakukan insert, update, dan delete. Hal ini bertujuan untuk memberikan gambaran singkat mengenai cara codenya. Karena segmen – segmen code ini akan sangat banyak berulang

Segmen Program 5.22 Insert Code Singkat

1. $user = Account::create([
2. "account\_username" => $request->username,
3. "account\_name" => $request->name,
4. "password" => $request->password,
5. 'account\_privileges' => [
6. 'title' => $request->role,
7. 'account\_dept' => $request->departemen,
8. ],
9. "account\_active" => false,
10. ]);

Pada Line 1 kita membuat objek kosong dan menentukan model data yang mau diinsert. Dalam hal ini account adalah model data yang kita pakai, kode create artinya kita mau melakukan insert. Selanjutnya line 2 hingga 10 adalah kode singkat terkait data apa saja yang mau dimasukkan. Menggunakan mongo db membisakan user untuk melakukan insert data dalam bentuk object lihat line 5 sampai 8.

Segmen Program 5.23 Update Code Singkat

1. $departemenupdate=Departemen::where('\_id',$request>id)>first();
2. $departemenupdate->Nama\_Departemen= $request->namadepartemen;
3. $departemenupdate->AksesTipeDatabase=$request->databaseakses;
4. $departemenupdate->save();

Objek contoh kali ini adalah untuk model data departemen. Langkah pertama yang harus dilakukan adalah mengambil atau mencari id data mana yang mau diubah, hal ini dilakukan pada line 1. Selanjutnya pada line 2 data yang kita dapatkan kita ganti valuenya. Begitu juga pada line 3 kita mengganti data akses tipe database. Selanjutnya pada line 4 kita melakukan proses save untuk mengupdate data yang telah kita update. Perlu diingat nama panah yang diisikan disesuaikan dengan area fillable pada model data. Jika tidak disebutkan maka data tidak akan masuk dan diberikan peringatan bahwa perubahan yang dilakukan tidak akan berdampak apa – apa.

Segmen Program 5.24 Delete

1. $departemenupdate=Departemen::where('\_id',$request>id)>first();
2. $hapusdepartemen->delete();

Masih objek contoh yang sama disini objek contohnya adalah data departemen. Begitu pula halnya dengan proses update, delete juga perlu menentuka id mana yang mau dilakukan penghapusan. Hal ini didapatkan dengan code pada line nomor 1. Selanjutnya user yang telah mendapatkan idnya bisa melakukan panah delete yang ada pada line 2 untuk melakukan proses penghapusan. Jika model yang kita berikan menerapkan softdelete maka data tidak dihapus melainkan diberikan atribut tambahan berupa delete\_at yang berfungsi untuk menandai bahwa data ini sudah pernah dihapus. Jangan pernah membuat atribut dengan nama itu ketika menerapkan soft delete. Sebaliknya jika model tidak menerapkan soft delete maka data langsung akan dihapus.