

# SISTEM PAKAR PREDIKSI HARGA MOBIL BEKAS MENGGUNAKAN DECISION TREE BERBASIS WEB

<sup>1</sup> Muhammad Fikri Sofyan

<sup>2</sup>Mufid Nilmada

<sup>1</sup>Universitas Gunadarma, kikyfikri31@student.gunadarma.ac.id

<sup>2</sup>Universitas Gunadarma, mufidnilmada@staff.gunadarma.ac.id

## ABSTRAK

*Penjualan mobil bekas merupakan bisnis yang terus berkembang dan menjanjikan di Indonesia. Namun, banyak faktor yang mempengaruhi harga mobil bekas, sehingga sulit bagi pembeli atau penjual untuk menentukan harga yang tepat. Oleh karena itu, dibutuhkan sebuah sistem pakar yang dapat membantu pengguna dalam memprediksi harga mobil bekas. Pada penelitian ini, dikembangkan sebuah sistem pakar prediksi harga mobil bekas menggunakan decision tree berbasis web. Metode yang digunakan adalah pengumpulan data harga mobil bekas dari berbagai sumber, kemudian dilakukan preprocessing data dan pembuatan model decision tree menggunakan algoritma ID3. Hasil dari penelitian ini adalah sebuah sistem pakar yang dapat membantu pengguna dalam memprediksi harga mobil bekas dengan akurasi yang tinggi. Penghitungan harga berdasarkan spesifikasi mobil yang diinginkan oleh pengguna. Fitur sistem pakar ini dipasang pada web dengan tautan sebagai berikut <https://lzroto-pemasaran.vercel.app/>.*

**Kata kunci:** sistem pakar, prediksi harga mobil bekas, decision tree, web

## PENDAHULUAN

Mobil bekas selalu menjadi pilihan yang menarik bagi sebagian besar masyarakat Indonesia karena harganya yang lebih terjangkau dibandingkan mobil baru. Namun, harga mobil bekas seringkali menjadi masalah karena sulit ditentukan secara tepat. Ada banyak faktor yang memengaruhi harga mobil bekas seperti usia kendaraan, merk kendaraan, dan kondisi kendaraan yang berbeda-beda dari satu mobil ke mobil yang lainnya. Untuk membantu mengatasi masalah ini, sistem pakar hadir sebagai solusi yang tepat (Hasan & Asep, 2014). Sistem pakar merupakan sistem yang didesain untuk menyelesaikan masalah-masalah yang kompleks dengan menggunakan pengetahuan dari para ahli. Sistem ini memungkinkan pengguna untuk membuat keputusan atau prediksi dengan akurasi yang tinggi. Salah satu metode yang dapat digunakan dalam pembuatan sistem

pakar adalah decision tree (DAN. Aman, 2019).

Metode decision tree memungkinkan pembuatan model prediksi dengan membagi data menjadi beberapa bagian yang lebih kecil berdasarkan beberapa variabel atau atribut. Setiap variabel atau atribut memiliki tingkat kepentingan yang berbeda-beda dalam pembuatan keputusan. Pada penelitian ini, algoritma ID3 digunakan untuk membuat model decision tree yang dapat membantu pengguna dalam menentukan harga mobil bekas secara akurat (L. Hakim, 2004).

Penelitian ini bertujuan untuk mengembangkan sebuah sistem pakar prediksi harga mobil bekas menggunakan decision tree berbasis web. Dengan demikian, sistem ini dapat diakses secara online melalui browser web sehingga pengguna dapat dengan mudah menggunakannya di mana saja dan kapan saja. Dalam pengembangan

sistem ini, kami mempertimbangkan beberapa faktor yang memengaruhi harga mobil bekas seperti usia kendaraan, merk kendaraan, dan kondisi kendaraan. Sistem pakar ini mampu memberikan prediksi harga mobil bekas dengan akurasi yang tinggi berdasarkan data yang diinputkan oleh pengguna.

Dengan adanya sistem pakar ini, diharapkan pengguna akan lebih mudah dan cepat menentukan harga mobil bekas dengan akurasi yang tinggi. Selain itu, sistem ini juga dapat membantu pengguna dalam membuat keputusan pembelian mobil bekas yang tepat sehingga dapat menghemat waktu dan uang.

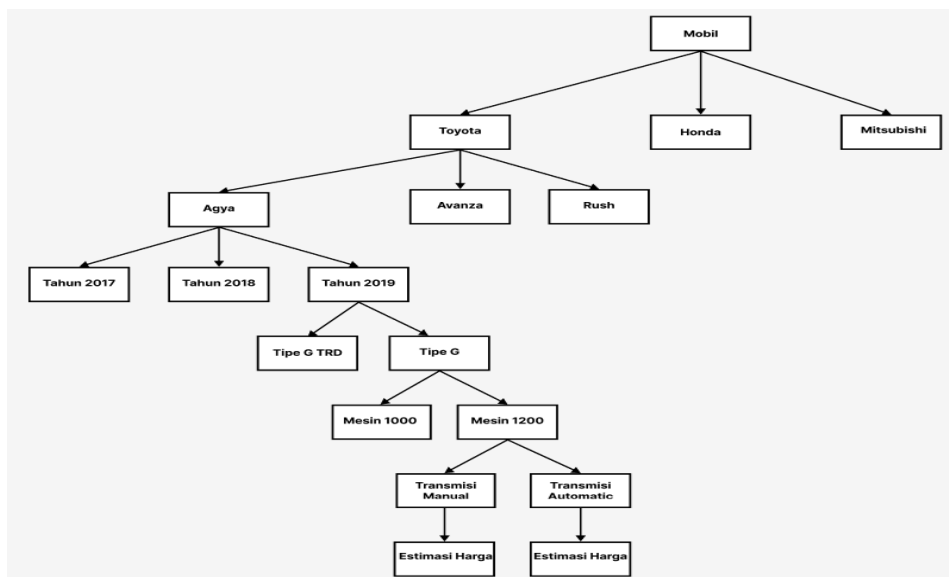
### METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah pengumpulan data harga mobil bekas dari berbagai sumber online, seperti situs jual beli mobil bekas dan forum otomotif. Data yang diperoleh kemudian diolah menggunakan metode preprocessing data untuk membersihkan dan mempersiapkan data

sebelum dibuat model decision tree. Metode preprocessing yang digunakan antara lain penghapusan data yang tidak relevan, imputasi data yang hilang, dan encoding data kategorikal. Selanjutnya, model decision tree dibuat dengan menggunakan algoritma ID3 dan diimplementasikan ke dalam sistem pakar berbasis web. Pembuatan website menggunakan metode System Development Live Cycle (SDLC). Uji coba dilakukan dengan memasukkan beberapa data mobil bekas dan membandingkan hasil prediksi harga dari sistem pakar dengan harga sebenarnya. Hasil dari penelitian ini diharapkan dapat membantu pengguna dalam menentukan harga mobil bekas secara akurat.

### HASIL DAN PEMBAHASAN

Berdasarkan pengumpulan data harga mobil bekas dari berbagai merk kendaraan, tahun pembuatan serta kondisinya, maka dibuatkan decision tree dari sebagian data yang tergambar dalam diagram berikut ini.



**Gambar 1.** Alur penelusuran decision tree

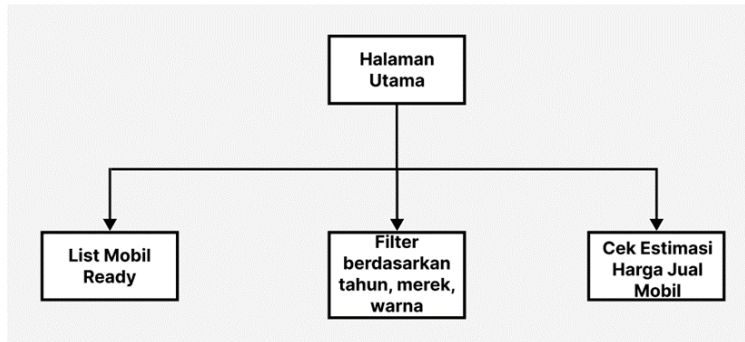
Pada gambar 1 diatas memperlihatkan alur penelusuran

decision tree mulai dari jenis pabrikan mobil hingga sampai pada harga

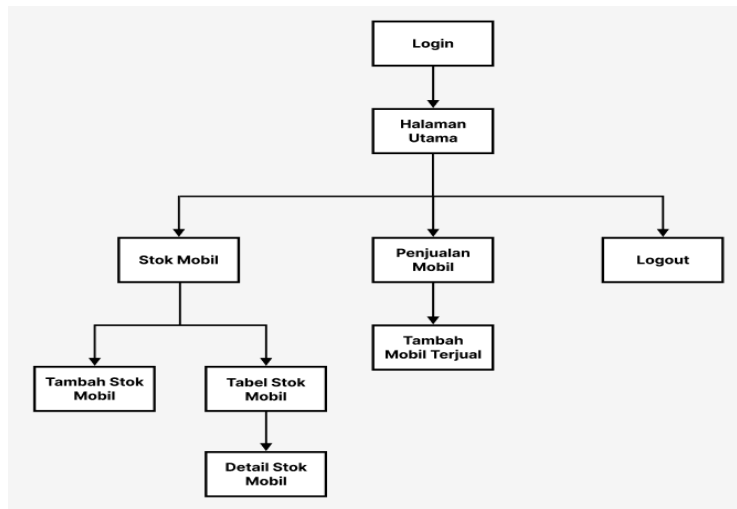
estimasi mobil bekas. Sistem pakar ini menggunakan data enam pabrikan mobil yaitu Toyota, Daihatsu, Mazda, Nissan, Honda dan Mitsubishi. Gambar 1 diatas hanya menampilkan sebagian dari alur proses penelusuran decision tree.

Dalam merancang sebuah aplikasi perlu adanya penggambaran

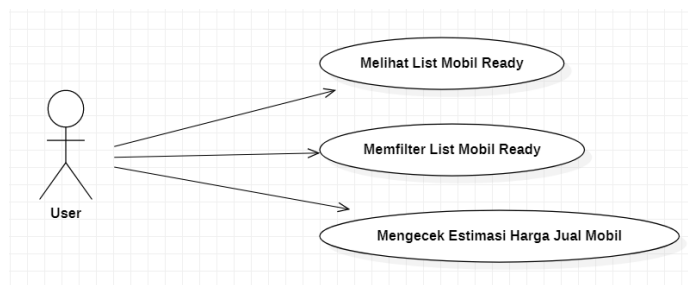
dari sistem yang dibangun. Penggambaran tersebut dapat dituliskan ke dalam bentuk struktur navigasi. Di bawah ini merupakan struktur navigasi yang digunakan untuk merancang aplikasi. Tipe struktur navigasi yang digunakan untuk web admin panel adalah hirarki, sedangkan untuk web pemasaran adalah linier.



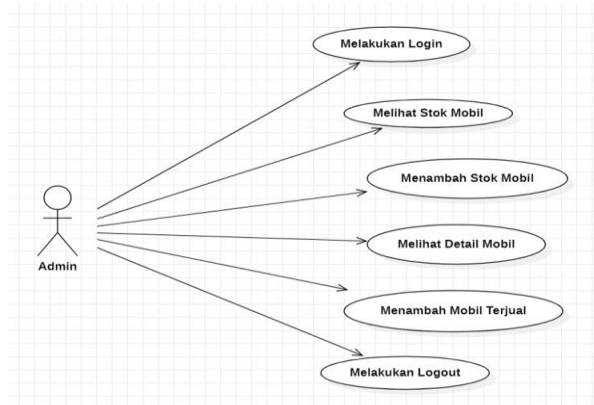
**Gambar 2. Struktur navigasi pengguna**



**Gambar 3. Struktur navigasi admin**



**Gambar 4. Diagram Use case untuk pengguna**



**Gambar 5. Diagram use case untuk admin**

Gambar 2 diatas menjelaskan peta yang dapat ditelusuri oleh pengguna ketika menggunakan web penjualan mobil yang ditambahkan fitur khusus untuk estimasi harga mobil bekas.

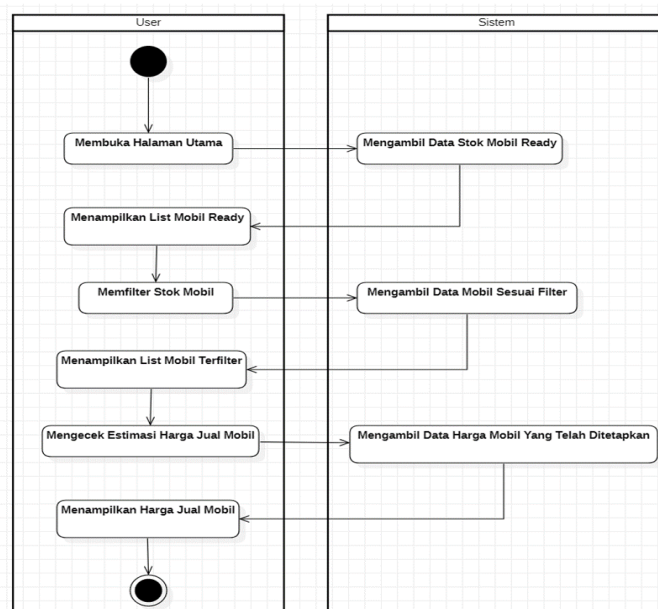
Gambar 3 diatas menjelaskan laman-laman yang dapat ditelusuri oleh admin ketika menggunakan web penjualan mobil ini.

Diagram use case menggambarkan secara ringkas aktor yang menggunakan sistem dan apa saja yang dapat dilakukannya. Pada diagram ini tidak menggambarkan secara detail tentang penggunaan use case, namun hanya memberi gambaran singkat hubungan antara use case, aktor, dan sistem.

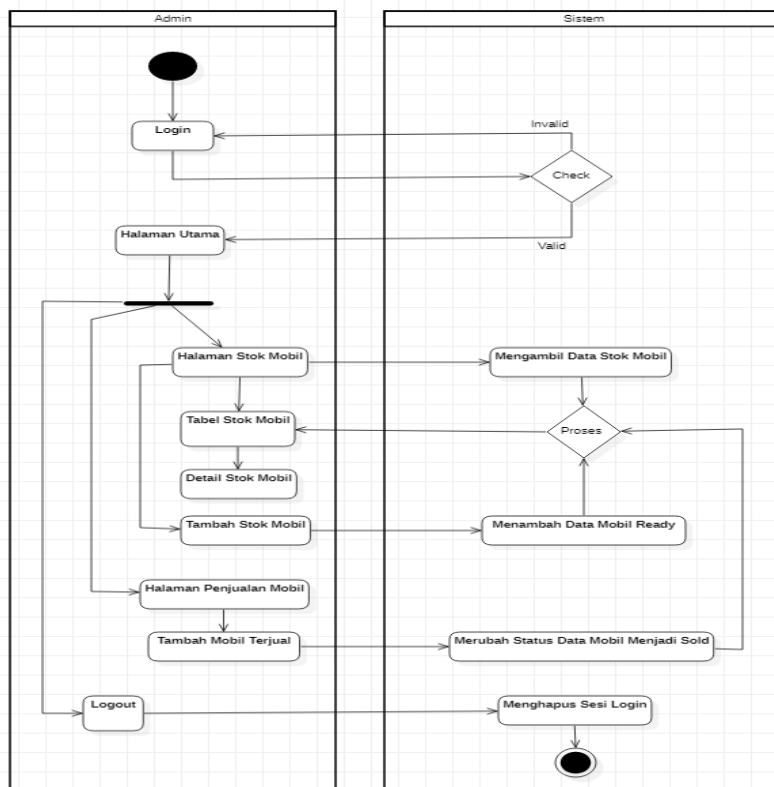
Diagram usecase pada gambar 4 menjelaskan aktor pengguna dapat melihat daftar mobil, menyaring daftar mobil yang ingin dilihat dan melakukan estimasi harga jual mobil bekas.

Selain diagram usecase untuk pengguna, dicantumkan juga diagram usecase untuk admin seperti yang terlihat pada gambar 5 diatas. Diagram tersebut menjelaskan apa saja yang dapat dilakukan oleh admin. Admin dapat melakukan beberapa hal mulai dari login hingga melakukan logout.

Diagram aktivitas digunakan untuk menggambarkan workflow (alur kerja) atau aktifitas dari sebuah sistem, berikut merupakan diagram aktivitas dari website yang dibangun.



**Gambar 6. Diagram aktivitas dari pengguna**



**Gambar 7. Diagram Aktivitas Admin**

**Tabel 1.  
Struktur Tabel Akun**

Field	Tipe Data	Keterangan
id	varchar(50)	Primary Key
role_id	varchar(10)	Foreign Key Related Tabel role id
name	varchar(50)	NOT NULL
phone	varchar(15)	NOT NULL
email	varchar(50)	NOT NULL
password	varchar(50)	NOT NULL
created_at	datetime	NOT NULL
updated_at	datetime	NOT NULL

Diagram aktivitas dari pengguna memperlihatkan alur kerja dari sistem yang dapat dilakukan oleh pengguna website yang dibangun. Alur kerja dibuat sederhana agar pengguna tidak merasa kesulitan dalam menelusurinya. Aktivitas admin juga dapat dilihat berdasarkan diagram aktivitas di bawah ini.

Diagram aktivitas admin memperlihatkan alur aktivitas yang dapat dilakukan admin dalam aplikasi

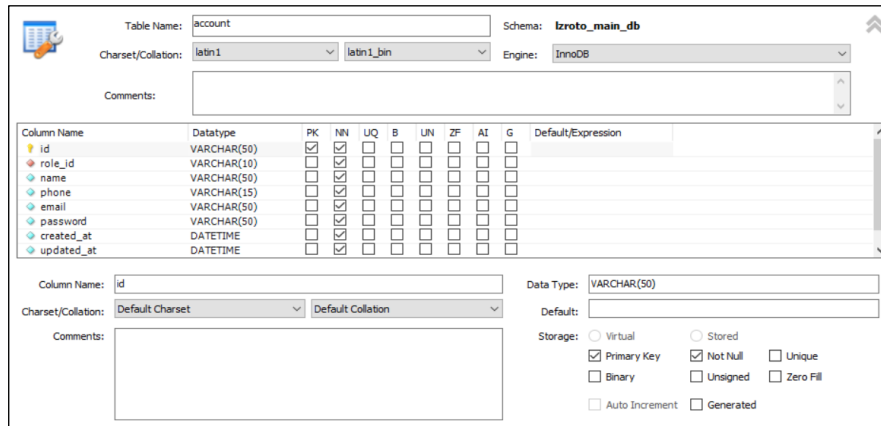
ini. Admin dapat melakukan mulai dari login hingga mengakhirinya dengan aktivitas logout.

Pada tahap perancangan database dimulai dengan membuat beberapa tabel. Pembuatan database dengan menggunakan MySQL dengan nama database yaitu lzroto\_main\_db. Di dalam database ini terdapat dibuat lima buah tabel, yaitu tabel akun, tabel otentifikasi, tabel penjualan mobil, tabel role dan tabel stok mobil (Arifin &

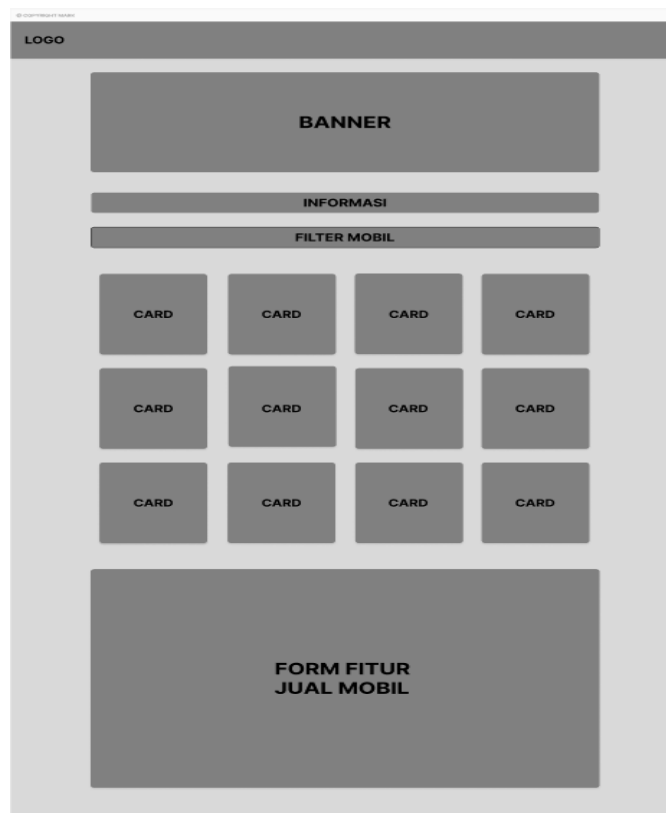
Krisnadita, 2017). Salah satunya adalah tabel akun seperti terlihat pada tabel 1 di bawah ini.

Tabel 1 memperlihatkan struktur akun dari pengguna yang akan menggunakan aplikasi. Field yang dibutuhkan adalah id, role\_id, nama,

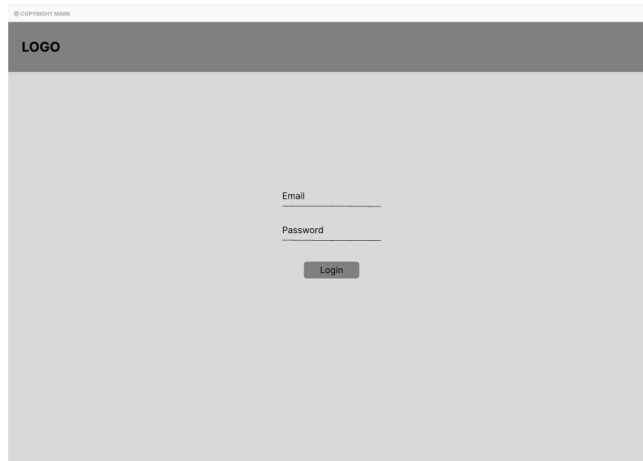
nomor telepon, alamat email serta password yang nanti akan digunakan oleh pengguna. Berdasarkan tabel ini maka dibuatkan tabel pada aplikasi dengan proses seperti terlihat pada gambar di bawah ini.



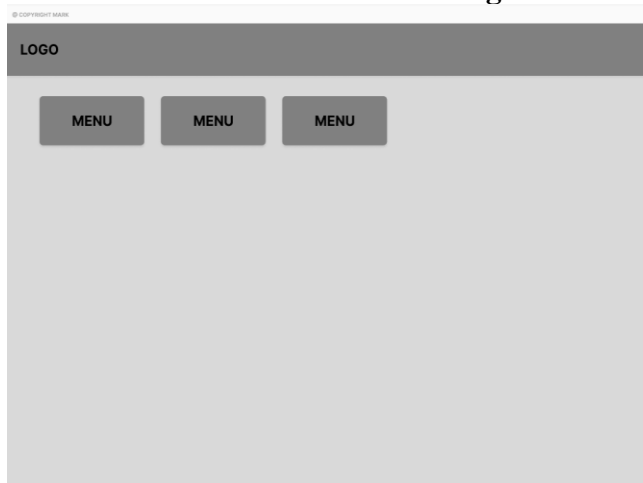
**Gambar 8. Proses pembuatan Tabel Akun**



**Gambar 9. Rancangan tampilan halaman utama**



**Gambar 10. Halaman Login**



**Gambar 11. Halaman Utama Admin Panel**

Pada gambar 8 terlihat proses pembuatan tabel akun yang berisi data admin dengan kolom id, role\_id, name, phone, email, password, created\_at, dan updated\_at. Field id adalah sebuah primary key dan role\_id adalah foreign key.

Dalam pembuatan halaman website, diperlukan membuat rancangan tampilan terlebih dahulu sebelum melakukan proses coding. Berikut ini gambar rancangan tampilan halaman utama.

Pada gambar 9 terlihat rancangan halaman awal ketika user membuka website pemasaran. Pada halaman ini terdapat logo copyright oleh LZR oto. Pada bagian bawahnya juga terdapat navbar yang berisikan nama perusahaan, yaitu LZR OTO. Terdapat filter dan daftar persediaan

mobil berstatus ready disertai gambar, keterangan mobil, dan harga mobil serta terdapat form untuk mengecek estimasi harga jual mobil.

Untuk pembuatan halaman login menggunakan Reactjs dengan bantuan library Material-UI, dan pengambilan data dari server side menggunakan fetch API dengan method POST untuk melakukan request ke database (Kusrini, 2009). Source code yang dibuat menggambarkan sebuah aplikasi web yang memiliki beberapa halaman terdaftar, di antaranya halaman "Stok Mobil", "Penjualan Mobil", dan "Logout". Setiap halaman memiliki route yang sesuai, yaitu "/stok", "/penjualan", dan "/logout". Selain itu, pada kode tersebut terdapat deklarasi variabel "cookies" yang menggunakan fungsi "Cookie()", yang digunakan

untuk memanipulasi cookie pada aplikasi web tersebut. Selanjutnya, pada komponen "Index" terdapat fungsi "navigate" yang menggunakan hook "useNavigate()" dari React Router untuk melakukan navigasi antar halaman. Fungsi "onClickeditem" digunakan untuk menangani klik pada elemen halaman dan memeriksa apakah halaman yang diakses adalah halaman logout. Jika iya, maka akan memanggil fungsi "logout()" dan mengarahkan pengguna kembali ke halaman awal ("/").

Untuk pembuatan halaman utama menggunakan Reactjs dengan bantuan library react-router-dom untuk berpindah ke halaman yang dituju dan cookie-universal untuk menyimpan sesi login (token) dari admin yang sedang

login (Salim, 2021). Halaman login merupakan halaman awal ketika website dibuka. mekanisme untuk masuk terdiri dari email, password, dan juga button login. Seperti yang bisa dilihat pada Gambar 10 di bawah ini.

Halaman login merupakan halaman awal ketika website dibuka. Pada halaman ini terdapat logo copyright oleh Lzr oto tersemat dibagian atas, serta identifikasi yang dibutuhkan untuk dapat login. Selanjutnya halaman utama merupakan halaman yang akan diakses setelah berhasil login. Pada halaman ini terdapat logo copyright oleh Lzr oto tahun 2022, seperti yang dapat dilihat pada Gambar 11 di bawah ini.

<input type="checkbox"/>	Merik Mobil <sup>1</sup>	Tahun <sup>2</sup>	Warna	Pajak <sup>3</sup>	Status	Action
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Terjual	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail
<input type="checkbox"/>	Cell	2018	LIWA	Chip	Ready	Lihat detail

**Gambar 12. Halaman Persediaan Mobil**

**Gambar 13. Halaman Pengisian identitas mobil**



**Gambar 14. Halaman input detail mobil**

**Gambar 15. Halaman penjualan website admin panel**

Pada gambar 11 diatas dapat dilihat, terdapat navbar yang berisikan nama perusahaan yaitu LZR OTO. Lalu di bawahnya terdapat button yang berisikan nama dari fitur-fitur utama yang ada pada website admin panel LZR OTO ini.

Pada halaman stok mobil juga terdapat logo copyright oleh LZR oto tahun 2022. Di Bawahnya juga terdapat navbar yang berisikan nama perusahaan yaitu LZR OTO. Lalu terdapat button Tambah Stok Mobil yang akan mengarahkan ke halaman Tambah Stok Mobil. Dan di bawahnya terdapat tabel yang memiliki kolom merk mobil, tahun, warna, pajak, status, dan action, seperti yang dapat dilihat pada Gambar 12 di bawah.

Pada gambar 12, terlihat pada bagian kanan bawah terdapat selector

page untuk user dapat memilih jumlah berapa data yang akan ditampilkan, dan arrow untuk pergi ke halaman sebelum atau selanjutnya. Pada halaman tambah stok mobil terdapat label, input, dan button untuk menyimpan data ke tabel pada halaman stok mobil, seperti yang bisa dilihat pada Gambar 13 di bawah ini.

Pada halaman detail mobil terdapat logo copyright oleh LZR oto tahun 2022. Di Bawahnya juga terdapat navbar yang berisikan nama perusahaan yaitu LZR OTO. Lalu terdapat label dan input yang akan menampilkan informasi detail dari mobil tersebut. Seperti yang dapat dilihat pada Gambar 14 di bawah ini.

Pada halaman penjualan mobil terdapat logo copyright oleh LZR oto tahun 2022. Di Bawahnya juga terdapat

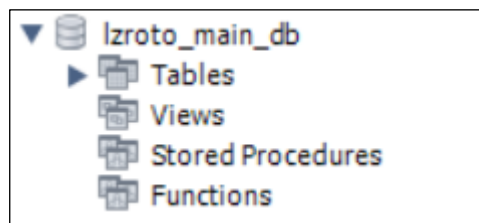
navbar yang berisikan nama perusahaan yaitu LZR OTO. Lalu terdapat label, input, dan button untuk menyimpan data mobil yang telah terjual. Seperti

yang bisa dilihat pada Gambar 15 dibawah ini.

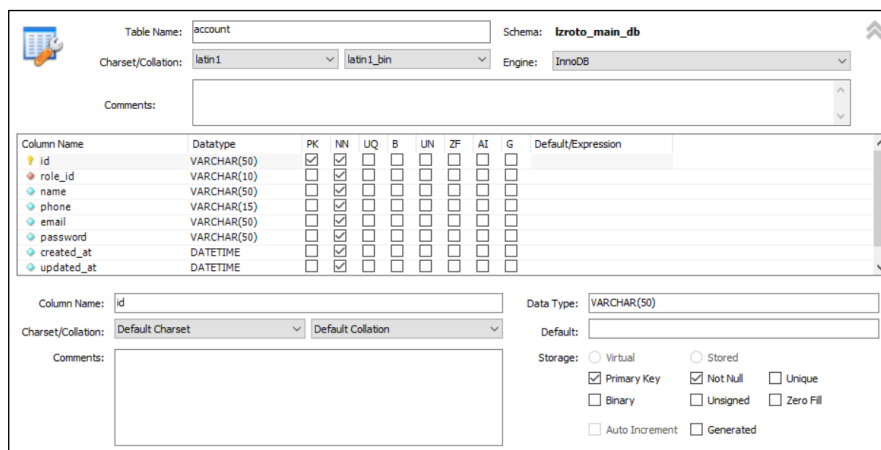
```
C:\Users\asus>node -v
v16.16.0

C:\Users\asus>npm -v
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.15.1
```

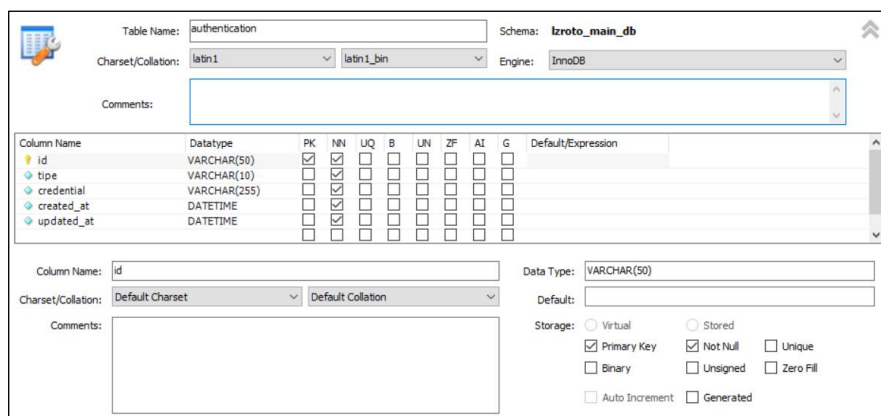
**Gambar 16. Informasi Nodejs telah terinstall**



**Gambar 17. Proses pembuatan database**



**Gambar 18. Pembuatan Tabel Account**



**Gambar 19. Struktur tabel Authentication**

Setelah tahap perancangan selesai, selanjutnya masuk pada tahap

pembuatan website. Pada tahap ini mulai proses pembuatan website LZR

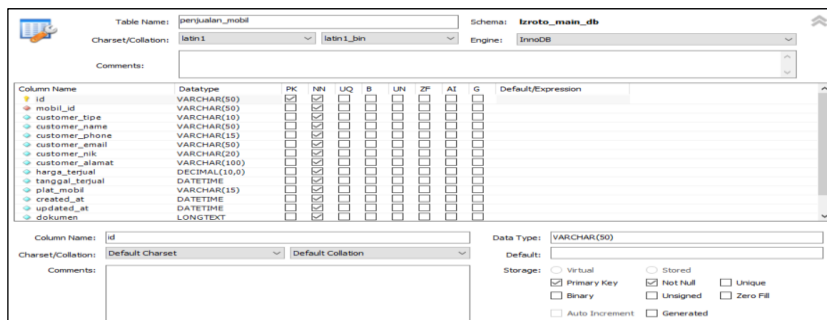
OTO berdasarkan analisis dan perancangan yang telah dilakukan sebelumnya. Tahap pertama yang dilakukan adalah menyiapkan perangkat lunak yang dibutuhkan yaitu Visual Studio Code, GIT, Database MySQL, Reactjs, Nodejs, dan bahasa pemrograman yang dipakai adalah JavaScript. Untuk instalasi cukup mengunduh installer yang telah disediakan di website resminya, yakni <https://nodejs.org/en/download/>. Kemudian setelah ter install cek di CMD dengan mengetikan `node-v` dan `npm-v` seperti Gambar 16 dibawah.

Terlihat pada gambar 16 informasi aplikasi nodejs telah terpasang dan siap untuk digunakan dalam proses pembuatan website ini. Hal lain yang dilakukan selanjutnya dalam pembuatan website adalah pembuatan sebuah database yang nantinya akan menampung seluruh data. Database yang dibuat dengan nama `lzroto_main_db` menggunakan MySQL seperti Gambar 17 dibawah ini.

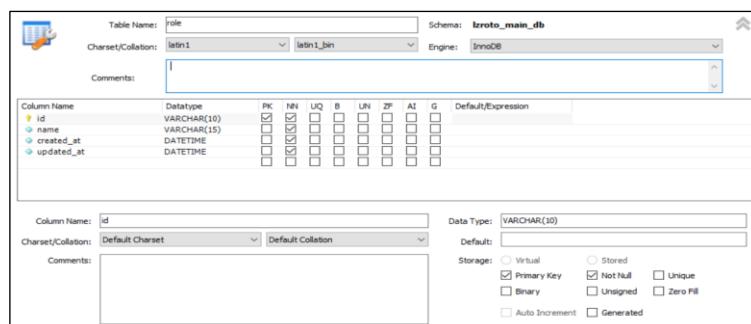
Selanjutnya pembuatan tabel `account`. Pada tabel ini akan berisi data admin dengan kolom `id`, `role_id`, `name`, `phone`, `email`, `password`, `created_at`, dan `updated_at`. Pada tabel ini, `id` adalah sebuah primary key dan `role_id` adalah foreign key seperti Gambar 18 di bawah ini.

Selanjutnya adalah pembuatan tabel `authentication`. Pada tabel ini akan berisi data log aktivitas login dari admin dengan kolom `id`, `tipe`, `credential`, `created_at`, dan `updated_at`. Pada tabel ini, `id` adalah sebuah primary key seperti Gambar 19 di bawah ini.

Selanjutnya adalah pembuatan tabel `penjualan mobil`. Pada tabel ini akan berisi data penjualan mobil dengan kolom `id`, `mobil_id`, `customer_tipe`, `customer_name`, `customer_email`, `customer_nik`, `customer_alamat`, `harga_terjual`, `tanggal_terjual`, `plat_mobil`, `created_at`, `updated_at`, dan `dokumen`. Pada tabel ini, `id` adalah sebuah primary key dan `mobil_id` adalah foreign key seperti Gambar 20 di bawah ini.



**Gambar 20. Struktur Tabel Penjualan Mobil**



**Gambar 21. Struktur tabel Role**

Column Name	Datatype	PK	NN	UQ	B	UN	ZP	AI	G	Default/Expression
id	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_merk	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_tahun	VARCHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_warna	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_pajak	VARCHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_status	VARCHAR(5)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_harga	DECIMAL(10,0)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_gambar	LONGTEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_dokumen	LONGTEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
created_at	DATETIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
updated_at	DATETIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_plat	VARCHAR(15)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_merk_detail	VARCHAR(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_sales_judul	VARCHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kendaraan_sales_deskripsi	VARCHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

**Gambar 22. Struktur tabel Stok\_mobil**

Selanjutnya adalah pembuatan struktur tabel Role. Pada tabel ini akan berisi data role dari admin dengan kolom id, name, created\_at, dan updated\_at. Pada tabel ini, id adalah sebuah primary key seperti Gambar 21 di bawah ini.

Tahap selanjutnya adalah pembuatan tabel Stok Mobil. Pada tabel ini akan berisi data persediaan mobil yang sedang dijual dan pernah dijual dengan kolom id, kendaraan\_merk, kendaraan\_tahun, kendaraan\_warna, kendaraan\_pajak, kendaraan\_status, kendaraan\_harga, kendaraan\_gambar, kendaraan\_dokumen, kendaraan\_plat, kendaraan\_merk\_detail, kendaraan\_sales\_judul, kendaraan\_sales\_deskripsi, created\_at, dan updated\_at. Pada tabel ini, id adalah sebuah primary key seperti Gambar 22 di bawah ini.

Selanjutnya pengerjaan program pembuatan website admin panel. Admin panel ini dikhususkan bagi para pegawai LZR-OTO yang ingin melihat dan menginput persediaan atau dokumen mobil. Untuk pembuatan halaman login menggunakan Reactjs dengan bantuan library Material-UI, dan pengambilan data dari server side menggunakan fetch API dengan method POST untuk melakukan request ke database. Source code di atas merupakan sebuah fungsi asynchronous

yang digunakan untuk menangani proses login pada suatu aplikasi web. Fungsi ini akan dipanggil ketika pengguna melakukan login dengan mengeklik tombol "login" atau sejenisnya. Pada awal fungsi, variabel "loading" di-set ke nilai true untuk menunjukkan bahwa proses login sedang berlangsung. Kemudian, dilakukan fetch ke endpoint login pada server yang di-hosting pada URL "https://lzroto-be.herokuapp.com/api/v1/user/user\_login". Request ini menggunakan metode POST dengan mode "cors" dan header "Content-Type" yang diatur sebagai "application/json". Body request berisi email dan password yang diambil dari elemen "emailRef" dan "passwordRef". Setelah request berhasil dilakukan, respons yang diterima di-parse sebagai JSON menggunakan fungsi "json()". Kemudian, hasil parse tersebut dicetak ke console menggunakan fungsi "console.log()". Jika status respons adalah "success", cookie dengan key "token" akan di-set dengan nilai JSON yang berisi data yang diterima dari server. Selanjutnya, pengguna akan diarahkan ke halaman "/initial" menggunakan fungsi "navigate()" dari React Router. Jika status respons adalah "unsuccessful", maka akan muncul sebuah alert dengan pesan yang diambil dari nilai "err\_code" pada respons. Jika

tidak ada respons atau respons status bukan "success" maupun "unsuccessful", maka akan muncul alert dengan pesan "Gagal Login". Jika terjadi error selama proses login, maka error tersebut dicetak ke console menggunakan fungsi "console.log()". Terakhir, variabel "loading" di-set kembali ke nilai false untuk menandakan bahwa proses login telah selesai.

Untuk pembuatan halaman utama menggunakan Reactjs dengan bantuan library react-router-dom untuk berpindah ke halaman yang dituju dan cookie-universal untuk menyimpan sesi login (token) dari admin yang sedang login. Source code yang dibuat mendefinisikan sebuah array bernama "registeredpage" yang berisi daftar halaman yang terdaftar pada aplikasi web, yaitu "Stok Mobil", "Penjualan Mobil", dan "Logout". Setiap halaman memiliki properti "name" untuk menyimpan nama halaman dan "route" untuk menyimpan rute URL yang sesuai. Selain itu, kode tersebut juga mendefinisikan variabel "cookies" yang digunakan untuk mengelola cookie pada aplikasi web.

Selanjutnya, terdapat komponen "Index" yang menggunakan hook "useNavigate" untuk mengelola navigasi antarhalaman dan fungsi "onClickeditem" untuk menangani klik pada elemen halaman. Fungsi "onClickeditem" memeriksa apakah halaman yang diakses adalah halaman logout. Jika iya, maka akan memanggil fungsi "logout()" dan mengarahkan pengguna kembali ke halaman awal ("/"). Jika tidak, maka akan melakukan navigasi ke halaman yang diinginkan. Fungsi "logout" digunakan untuk menghapus semua cookie yang terkait dengan aplikasi web. Selanjutnya, terdapat fungsi "validation" yang digunakan untuk memeriksa apakah pengguna telah login dengan memeriksa apakah cookie dengan key

"token" tersedia atau tidak. Jika tidak tersedia, maka pengguna akan diarahkan kembali ke halaman awal ("/"). Fungsi "validation" dijalankan menggunakan hook "useEffect" dengan parameter kedua berupa array kosong untuk menunjukkan bahwa fungsi ini hanya akan dijalankan sekali saat komponen "Index" pertama kali dirender.

Untuk pembuatan halaman stok mobil menggunakan Reactjs dengan bantuan library Material-UI, dan pengambilan data dari server side menggunakan fetch API dengan method GET untuk menampilkan data stok mobil dari database. Source code yang dibangun mendefinisikan sebuah fungsi bernama "getDatas" yang digunakan untuk mengambil data dari API dengan menggunakan metode GET. Fungsi ini mengirimkan permintaan ke endpoint "https://lzroto-be.herokuapp.com/api/v1/mobil/stok\_mobil" untuk mendapatkan daftar mobil yang tersedia. Fungsi ini menggunakan fungsi bawaan "fetch" untuk melakukan permintaan dan menggunakan header "Content-Type" dengan nilai "application/json" untuk menunjukkan bahwa data yang dikirimkan dan diterima berupa format JSON. Selain itu, fungsi ini juga menambahkan header "token" dengan nilai yang diambil dari cookie dengan key "token" dan menambahkan header "app\_version" dengan nilai "devlzrotoappsukses". Hal ini mungkin digunakan untuk mengidentifikasi aplikasi web yang melakukan permintaan. Setelah permintaan berhasil, data hasil permintaan diubah menjadi format JSON menggunakan fungsi "response.json()" dan disimpan ke dalam state "tableData" menggunakan fungsi "setTableData". Namun, jika terjadi kesalahan saat melakukan permintaan, maka fungsi ini akan menampilkan pesan error di

console dengan menggunakan fungsi "console.log(err)".

Untuk pembuatan halaman tambah stok mobil menggunakan Reactjs dengan bantuan library Material-UI, dan mengirim parameter body yang dibutuhkan menggunakan fetch API dengan method PUT ke database. Source code yang dibangun merupakan sebuah async function bernama `handleTambahStok()`. Function tersebut digunakan untuk menambahkan data mobil baru ke dalam database dengan mengirimkan request PUT ke endpoint "https://lzroto-be.herokuapp.com/api/v1/mobil/insert\_mobil" dengan parameter data mobil yang ingin ditambahkan. Data mobil yang ingin ditambahkan tersebut diambil dari beberapa variabel yang telah didefinisikan sebelumnya, seperti `merkMobil`, `tahunMobil`, `hargaMobil`, dan sebagainya. Request yang dikirimkan juga disertai dengan token authentication yang diambil dari cookies. Jika request berhasil, maka function akan menampilkan alert "Data Berhasil Ditambahkan", jika gagal maka akan menampilkan alert "Gagal Menambah Data".

Untuk pembuatan halaman detail mobil menggunakan Reactjs dengan bantuan library Material-UI, dan menggunakan fetch API dengan method POST ke database. Source code ini berfungsi untuk melakukan request ke API endpoint "https://lzroto-be.herokuapp.com/api/v1/mobil/stok\_mobil\_detail" dengan menggunakan method "POST" dengan mengirimkan data berupa id melalui request body. Request ini dilakukan menggunakan fetch API pada Javascript. Setelah mendapatkan response dari API, code akan melakukan pengecekan jika response tidak "ok" atau error, maka akan mengembalikan pesan error dengan status response. Jika response sukses, maka data yang diterima akan

di-set pada state `detaildata` menggunakan fungsi `setdetaildata`.

Untuk pembuatan halaman penjualan mobil menggunakan Reactjs dengan bantuan library Material-UI, dan mengirim parameter body yang dibutuhkan menggunakan fetch API dengan method PUT ke database. Source code ini terdiri dari dua async functions, yaitu `getMobilReady()` dan `handleSimpan()`. Pada

`getMobilReady()`, fungsi ini melakukan HTTP GET request ke endpoint `https://lzroto-be.herokuapp.com/api/v1/mobil/get_list_mobil_ready` dengan mode CORS dan header yang menyertakan token dan `app_version`. Hasil responsenya diubah ke JSON dan kemudian dimasukkan ke dalam state `platMobil`. Fungsi ini juga mengatur state loading menjadi true saat request sedang berjalan dan false saat selesai. Pada `handleSimpan()`, fungsi ini melakukan HTTP POST request ke endpoint yang tidak diberikan pada kode. Fungsi ini juga mengatur state loading menjadi true saat request sedang berjalan dan false saat selesai. Di dalam try-catch, fungsi ini mencoba mengakses response dari request dan melakukan beberapa operasi dengan menggunakan `ref`. Jika response berhasil, fungsi akan mengembalikan alert yang sesuai berdasarkan status. Namun, jika terdapat error pada proses request, maka error tersebut akan dicetak di konsol.

Selanjutnya pembuatan fungsi `logout` menggunakan bantuan library `cookie-universal` untuk menghapus sesi login (token) yang ada. Source code tersebut merupakan sebuah fungsi yang digunakan untuk handle sebuah event ketika sebuah item di klik. Fungsi tersebut memiliki parameter `val`, yang merupakan nilai yang dikirimkan ketika fungsi dipanggil. Jika nilai `val` sama dengan `"/logout"`, maka akan dipanggil fungsi `logout()` yang menghapus semua cookies yang ada. Setelah itu, akan

dipanggil fungsi `navigate("/")` yang akan mengarahkan pengguna kembali ke halaman utama. Jika nilai `val` bukan `"/logout"`, maka akan ditampilkan pesan `log "onclickeditem"` diikuti dengan nilai `val`, kemudian akan dipanggil fungsi `navigate(val)` yang akan mengarahkan pengguna ke halaman sesuai dengan nilai `val` yang diterima.

Pengujian merupakan tahapan penting dalam siklus pengembangan perangkat lunak. Dalam pengujian aplikasi, digunakan metode `blackbox`

yang berfokus pada persyaratan fungsional perangkat lunak. Metode ini difokuskan pada pengujian input dan output aplikasi untuk memastikan kesesuaian dengan harapan yang diinginkan. User dan admin dapat mengakses halaman website dan menggunakan berbagai fungsi yang telah disediakan. Pada Tabel 2 di bawah ini, terdapat hasil pengujian sistem dari setiap fungsi yang dilakukan dengan metode `blackbox`.

**Tabel 2.**  
**Tabel Pengujian Fungsi Aplikasi Dengan Metode Blackbox**

No	Fungsi	Skenario	Hasil Yang Diharapkan	Hasil Nyata	Hasil
1.	Mengakses halaman utama web pemasaran	Membuka halaman utama	Berhasil mengakses halaman utama dan menampilkan persediaan mobil ready	Berhasil mengakses halaman utama dan menampilkan persediaan mobil ready	Sukses
		Memfilter persediaan mobil	Berhasil Menampilkan persediaan mobil yang sudah difilter	Berhasil Menampilkan persediaan mobil yang sudah difilter	
		Memberikan estimasi harga kepada user yang ingin menjual mobil	Berhasil menampilkan harga ke user	Berhasil menampilkan harga ke user	
2.	Mengakses halaman login web admin panel	Membuka halaman login	Berhasil mengakses halaman login	Berhasil mengakses halaman login	Sukses
3.	Login admin	Memasukan Email dan Password (benar)	Berhasil masuk ke halaman utama admin panel	Berhasil masuk ke halaman utama admin panel	Sukses
		Memasukan Email dan Password (salah dan kosong)	Gagal masuk ke halaman utama admin panel	Gagal masuk ke halaman utama admin panel	Sukses

Tabel 2 lanjutan

4.	Mengakses halaman stok mobil	Melihat tabel data persediaan mobil yang ready ataupun sudah terjual	Berhasil Melihat tabel data persediaan mobil	Berhasil Melihat tabel data persediaan mobil	Sukses
		Memasukan data mobil baru ke dalam tabel persediaan mobil	Berhasil memasukan data mobil baru ke dalam tabel persediaan mobil	Berhasil memasukan data mobil baru ke dalam tabel persediaan mobil	Sukses
		Melihat detail mobil yang ada pada tabel persediaan mobil	Berhasil melihat detail mobil yang ada pada tabel persediaan mobil	Berhasil melihat detail mobil yang ada pada tabel persediaan mobil	Sukses
5.	Mengakses halaman penjualan mobil	Memasukan data mobil yang akan terjual dan mengupdate status menjadi sold di tabel persediaan mobil	Berhasil Memasukan data mobil yang terjual dan mengupdate status menjadi sold di tabel persediaan mobil	Berhasil Memasukan data mobil yang terjual dan mengupdate status menjadi sold di tabel persediaan mobil	Sukses
6.	Logout	Menekan tombol logout	Keluar dari halaman utama dan menuju halaman login	Keluar dari halaman utama dan menuju halaman login	Sukses

**Tabel 3.**  
**Hasil Pengujian Operasional Website**

No	Fungsi	Jenis Kesalahan			Kesalahan Inisialisasi	
		Fungsi Benar / Hilang	Kesalahan Interface	Kesalahan Struktur Data		Kesalahan Kinerja
1.	Mengakses halaman utama web pemasaran	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada
2.	Mengakses halaman login web admin panel	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada
3.	Login admin	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada
4.	Mengakses halaman stok mobil	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada



Tabel 3 lanjutan

5.	Mengakses halaman penjualan mobil	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada
6.	Logout	Tidak ada	Tidak ada	Tidak ada	Tidak ada	Tidak ada

**Tabel 4.**  
**Hasil Kuesioner Penilaian Pengguna**

No	Pertanyaan	Skala Penilaian					Penilaian (%)
		5	4	3	2	1	
1	Apakah tampilan website sudah sesuai fungsinya?	4	14	8	0	0	76,92
2	Apakah fitur yang terdapat di website ini mudah dipahami?	6	14	6	0	0	80
3	Apakah website ini mudah untuk digunakan?	6	17	3	0	0	82,30
4	Apakah website ini sudah efektif dan efisien?	3	12	10	1	0	73,07
5	Apakah website ini sudah cukup baik?	3	14	9	0	0	75,38
Total Penilaian							387,67
Rerata Penilaian							77,53%

Tabel 2 memperlihatkan pengujian enam fitur utama berjalan dengan baik dan berfungsi sesuai rencana awal pembuatan.

Selanjutnya adalah melakukan pengujian terhadap 5 jenis kesalahan standar dalam pengoperasian website, tabel 3 di bawah ini menampilkan hasil pengujian tersebut.

Dari Hasil pengujian website yang disajikan pada tabel 3 dapat disimpulkan tidak ada kesalahan yang muncul ketika proses pengujian, sehingga semua fungsi yang ada pada website telah berjalan dengan baik dan layak untuk digunakan.

Pada tahap uji coba pengguna, dilakukan pengambilan data dari 26 responden sebagai sampel. Uji coba tersebut bertujuan untuk mengevaluasi sejauh mana fungsi website dapat berjalan dengan baik dan memberikan manfaat bagi penggunanya. Skala likert digunakan dalam pengujian ini, dengan responden yang terdiri dari keluarga dan teman penulis. Hasil dari pengujian ini dapat dilihat pada tabel 4 di bawah.

## KESIMPULAN DAN SARAN

Sistem pakar ini berhasil dibuat menggunakan bahasa pemrograman javascript framework yaitu Reactjs, dan MySQL untuk databasenya. Website ini dapat membantu memberikan solusi dalam melihat persediaan mobil yang dijual, memfilter persediaan mobil, dan memberi estimasi harga mobil yang ingin dijual oleh pelanggan menggunakan decision tree. Aplikasi admin panel membantu user dalam penyimpanan dokumen yang lebih praktis. Selanjutnya melalui pengujian dengan menggunakan metode blackbox dengan hasil semua fungsi berjalan dengan baik dan uji coba pengguna menggunakan metode skala likert dengan hasil presentase 74,66% dari 26 responden, yang artinya dalam skala likert website LZR OTO yang didalamnya terdapat sistem pakar untuk estimasi harga mobil bekas ini mendapat hasil yang baik. Untuk websitenya bisa diakses melalui link <https://lzroto-pemasaran.vercel.app/>.

Beberapa aspek dari sistem pakar berbasis website ini yang masih dapat dikembangkan, seperti; Website admin panel menambahkan fitur untuk absensi pegawai. Report penjualan dengan periode waktu tertentu. Kalkulasi pembelian secara kredit. Pengaturan akses user yang lebih baik.

Sedangkan untuk website pemasarannya dapat ditambahkan proses lebih lanjut dari tahap estimasi harga mobil secara mandiri, dilanjutkan dengan mekanisme proses pemeriksaan mobil, serta mengembangkan aplikasi ini berbasis mobile.

#### **DAFTAR PUSTAKA**

- Kusrini & Emha Taufiq Luthfi, 2009, *Algoritma Data Mining*, CV ANDI OFFSET, Yogyakarta.
- Lukmanul, Hakim, 2004. *Website Merupakan Fasilitas Internet*. Jakarta: Gramedia.
- D. A. N. Aman, "Belajar Di Internet Melalui Sosialisasi Edukasi Internet," vol. 1, no. 1, pp. 17–20, 2019.
- Hasan Abdurahman dan Asep Ririh Riswaya. 2014. *Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti*. *Jurnal Computech & Bisnis*, Vol. 8 No. 2
- Arifin, S., & Krisnadita, Y. (2017). *Aplikasi Plugin Transfer Domain Di PT Beon Intermedia*. *Jurnal Teknologi Informasi: Teori, Konsep, dan Implementasi*, 8(1), 1-84.
- Juliansyah, F., Utomo, S., Rachmanto, A., & Budiarto, S. *APLIKASI QUIZ DENGAN KONSEP GAMIFICATION BERBASIS WEB MENGGUNAKAN RUBY ON RAILS & REACT. JS*. *Jurnal Teknologi Informasi dan Komunikasi*, 11(2).  
<https://idprogrammer.com/pengertian-nodejs-lengkap>, IDprogrammer (2020).
- Salim, A. (2021). *Perancangan Frontend Aplikasi Pemandu Pariwisata Menggunakan Framework React. Js di Provinsi Jawa Barat*. *Tematik: Jurnal Teknologi Informasi Komunikasi (e-Journal)*, 8(1), 132-145.
- Kirsan, A. S., Arisa, N. N., & Putra, A. H. D. (2022). *Perancangan Sistem Informasi Manajemen Badan Amal*