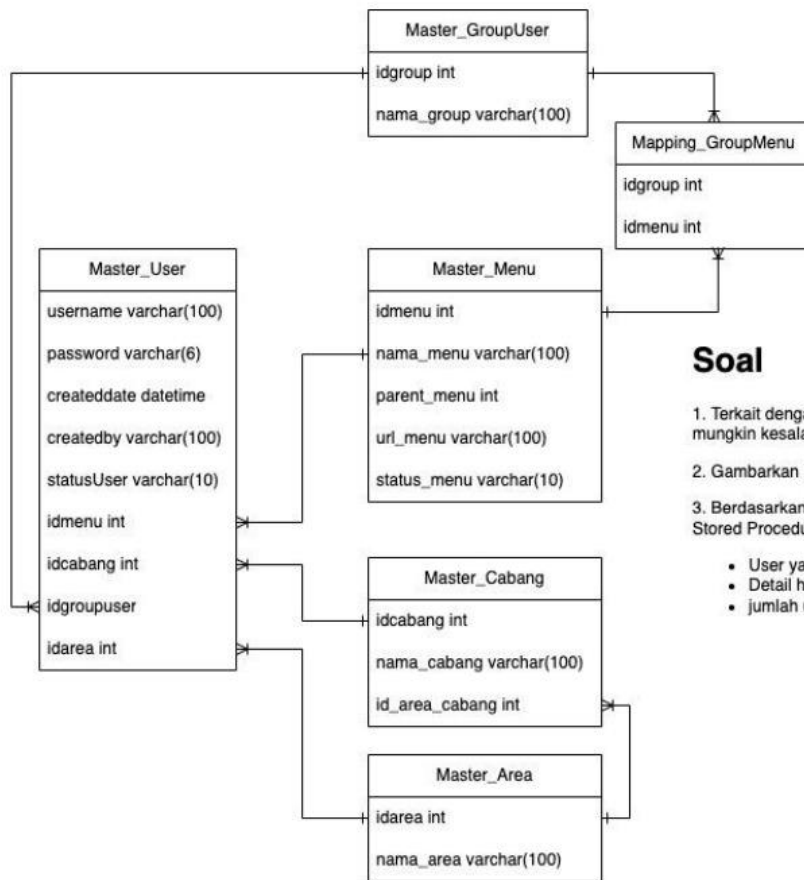


Soal Test Code

BCA Finance 2024



Soal

1. Terkait dengan normalisasi database, sebutkan sebanyak mungkin kesalahan pada diagram database disamping
2. Gambarkan ERD yang sudah dinormalisasi
3. Berdasarkan ERD yang sudah dinormalisasi, Buatlah View / Stored Procedure untuk menarik data :
 - User yang masih aktif
 - Detail hak akses menu per user
 - jumlah user aktif yang terdaftar per area

Jawaban soal test code BCA finance 2024

1. Kesalahan normalisasi dalam diagram database

Berdasarkan prinsip normalisasi, berikut beberapa kesalahan yang dapat ditemukan pada diagram ERD di gambar.

a. Panjang password tidak aman

- Password pada tabel Master_User memiliki type varchar(6), yang terlalu pendek dan tidak aman.
- Sebaiknya gunakan varchar(255) dengan hashing untuk keamanan.

b. Redundansi Penyimpanan

Idgroup di Master_User

- Master_User sudah memiliki idgroupuser, tetapi ada juga tabel Mapping_GroupMenu yang menghubungkan idgroup dan idmenu.

- Bisa dibuat tabel relasi tambahan agar akses lebih fleksibel.

c. Kurangnya Konvensi Penamaan

- statusUser dan status_menu lebih baik menggunakan tipe ENUM atau BOOLEAN jikahanya memiliki dua status (aktif/nonaktif).
- createBy lebih baik sebagai created_by untuk meningkatkan keterbacaan.

d. Kurang Foreign Key di *Mapping_GroupMenu*

- Tidak ada hubungan eksplisit antara idgroup dari Master_GroupUser dan Mapping_GroupMenu.

e. Redundansi Data dalam *Master_Cabang* dan *Master_Area*

- Master_Cabang sudah memiliki ide_area_cabang, sehingga Master_User tidak perlu menyimpan idarea secara langsung

2. ERD yang sudah Dinormalisasi

ERD yang sudah dinormalisasi akan memiliki beberapa perubahan :

- Memisahkan relasi user dan group agar lebih fleksibel.
- Menambahkan foreign key yang lebih jelas
- Menormalisasi field password untuk keamanan.

ERD Normalisasi :

a. Master_User

- id_user (PK, INT, Auto Incerment)
- username (VARCHAR(100), UNIQUE)
- password_hash (VARCHAR(255))
- created_at (DATETIME)
- created_by (INT, FK ke ide_user)
- status_user (ENUM('active', 'inactive'))
- ide_group (FK ke Master_GroupUser)
- ide_cabang (FK ke Master_Cabang)

b. Master_GroupUser

- id_group (PK, INT)
- nama_group (VARCHAR(100))

c. Master_Menu

- id_menu (PK, INT)
- nama_menu (VARCHAR(100))

- parent_menu (FK ke id_menu, NULLABLE)
 - url_menu (VARCHAR(255))
 - status_menu (ENUM('active', 'inactive'))
- d. Mapping_GroupMenu
- id_group (FK ke Master_GroupUser)
 - id_menu (FK ke Master_Menu)
- e. Master Cabang
- id_cabang (PK, INT)
 - nama_cabang (VARCHAR(100))
 - id_area (FK ke Master_Area)
- f. Master_Area
- id_area (PK, INT)
 - nama_area (VARCHAR(100))

3. View /Stored Procedure untuk Menarik Data

Berikut adalah SQL untuk menarik data sesuai dengan kebutuhan soal

- a. View: User yang masih Aktif

```
CREATE VIEW active_users AS
SELECT id_user, username, id_group, id_cabang
FROM Master_User
WHERE status_user = 'active';
```



```
CREATE VIEW active_users AS
SELECT id_user, username, id_group, id_cabang
FROM Master_User
WHERE status_user = 'active';
```

- b. View : Detail Hak Akses Menu Per User

```
CREATE VIEW user_menu_access AS
SELECT u.id_user, u.username, g.nama_group, m.nama_menu, m.url_menu
FROM Master_User u
JOIN Master_GroupUser g ON u.id_group = g.id_group
JOIN Mapping_GroupMenu mg ON g.id_group = mg.id_group
```

```
JOIN Master_Menu m ON mg.id_menu = m.id_menu
```

```
WHERE u.status_user = 'active';
```

```
CREATE VIEW user_menu_access AS
SELECT u.id_user, u.username, g.nama_group, m.nama_menu, m.url_menu
FROM Master_User u
JOIN Master_GroupUser g ON u.id_group = g.id_group
JOIN Mapping_GroupMenu mg ON g.id_group = mg.id_group
JOIN Master_Menu m ON mg.id_menu = m.id_menu
WHERE u.status_user = 'active';
```

- c. View : Jumlah User Aktif Per Area

```
CREATE VIEW active_users_per_area AS
```

```
SELECT a.id_area, a.nama_area, COUNT(u.id_user) AS jumlah_user_aktif
```

```
FROM Master_User u
```

```
JOIN Master_Cabang c ON u.id_cabang = c.id_cabang
```

```
JOIN Master_Area a ON c.id_area = a.id_area
```

```
WHERE u.status_user = 'active'
```

```
GROUP BY a.id_area, a.nama_area;
```

```
CREATE VIEW active_users_per_area AS
SELECT a.id_area, a.nama_area, COUNT(u.id_user) AS
FROM Master_User u
JOIN Master_Cabang c ON u.id_cabang = c.id_cabang
JOIN Master_Area a ON c.id_area = a.id_area
WHERE u.status_user = 'active'
GROUP BY a.id_area, a.nama_area;
```

- d. Code Normalisasi ERD :

```
CREATE TABLE Master_GroupUser (
```

```
    idgroup INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nama_group VARCHAR(100) NOT NULL
```

```
);
```

```
CREATE TABLE Master_Area (
```

```
    idarea INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nama_area VARCHAR(100) NOT NULL
```

```
);
```

```
CREATE TABLE Master_Cabang (  
    idcabang INT PRIMARY KEY AUTO_INCREMENT,  
    nama_cabang VARCHAR(100) NOT NULL,  
    idarea INT NOT NULL,  
    FOREIGN KEY (idarea) REFERENCES Master_Area(idarea) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE Master_User (  
    iduser INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(100) UNIQUE NOT NULL,  
    password VARCHAR(255) NOT NULL, -- Disarankan menggunakan hashing  
    created_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    created_by INT, -- Referensi ke iduser jika ada sistem pengguna yang membuat  
user lain  
    status_user ENUM('active', 'inactive') NOT NULL,  
    idgroup INT,  
    idcabang INT,  
    idarea INT,  
    FOREIGN KEY (idgroup) REFERENCES Master_GroupUser(idgroup),  
    FOREIGN KEY (idcabang) REFERENCES Master_Cabang(idcabang),  
    FOREIGN KEY (idarea) REFERENCES Master_Area(idarea)  
);
```

```
CREATE TABLE Master_Menu (  
    idmenu INT PRIMARY KEY AUTO_INCREMENT,  
    nama_menu VARCHAR(100) NOT NULL,  
    parent_menu INT NULL, -- Untuk menu yang memiliki sub-menu  
    url_menu VARCHAR(100) NULL,  
    status_menu ENUM('active', 'inactive') NOT NULL,  
    FOREIGN KEY (parent_menu) REFERENCES Master_Menu(idmenu) ON  
DELETE SET NULL  
);
```

```

CREATE TABLE Mapping_GroupMenu (
    idgroup INT,
    idmenu INT,
    PRIMARY KEY (idgroup, idmenu),
    FOREIGN KEY (idgroup) REFERENCES Master_GroupUser(idgroup) ON
DELETE CASCADE,
    FOREIGN KEY (idmenu) REFERENCES Master_Menu(idmenu) ON DELETE
CASCADE
);

```

```

CREATE TABLE Master_GroupUser (
    idgroup INT PRIMARY KEY AUTO_INCREMENT,
    nama_group VARCHAR(100) NOT NULL
);

CREATE TABLE Master_User (
    iduser INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL, -- Disarankan menggunakan hashing
    created_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    created_by INT, -- Referensi ke iduser jika ada sistem pengguna yang membuat user lain
    status_user ENUM('active', 'inactive') NOT NULL,
    idgroup INT,
    idcabang INT,
    idarea INT,
    FOREIGN KEY (idgroup) REFERENCES Master_GroupUser(idgroup),
    FOREIGN KEY (idcabang) REFERENCES Master_Cabang(idcabang),
    FOREIGN KEY (idarea) REFERENCES Master_Area(idarea)
);

CREATE TABLE Master_Menu (
    idmenu INT PRIMARY KEY AUTO_INCREMENT,
    nama_menu VARCHAR(100) NOT NULL,
    parent_menu INT NULL, -- Untuk menu yang memiliki sub-menu
    url_menu VARCHAR(100) NULL,
    status_menu ENUM('active', 'inactive') NOT NULL,
    FOREIGN KEY (parent_menu) REFERENCES Master_Menu(idmenu) ON DELETE SET NULL
);

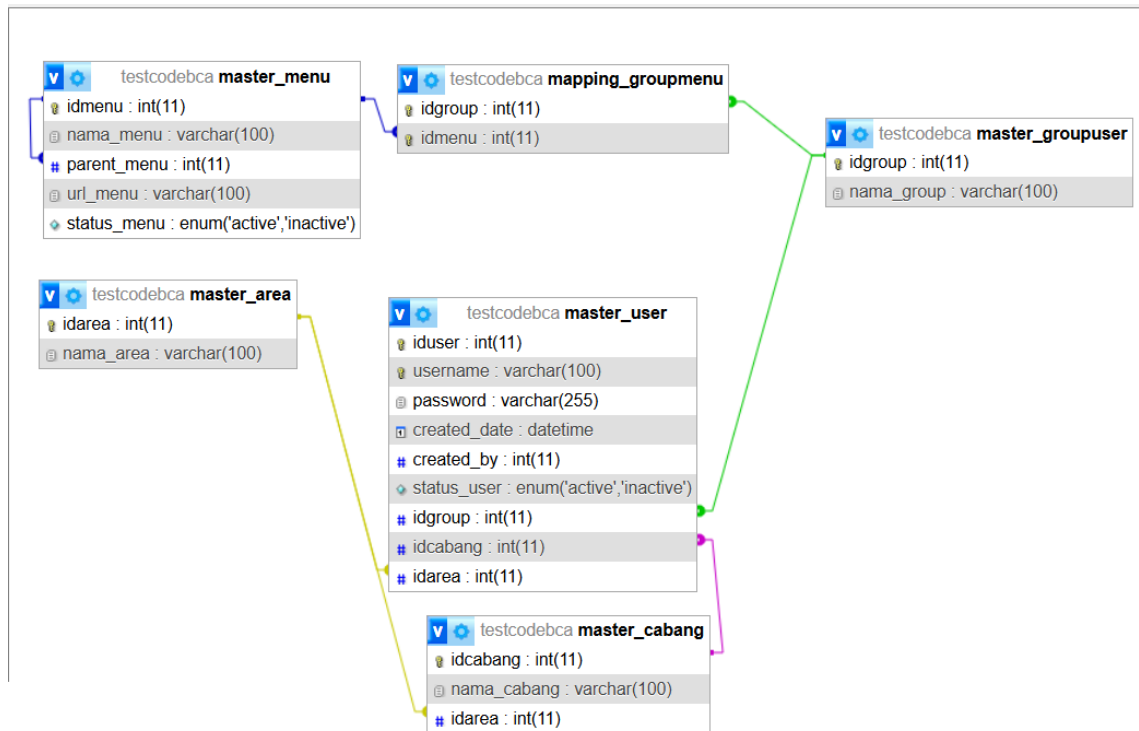
CREATE TABLE Mapping_GroupMenu (
    idgroup INT,
    idmenu INT,
    PRIMARY KEY (idgroup, idmenu),
    FOREIGN KEY (idgroup) REFERENCES Master_GroupUser(idgroup) ON DELETE CASCADE,
    FOREIGN KEY (idmenu) REFERENCES Master_Menu(idmenu) ON DELETE CASCADE
);

CREATE TABLE Master_Area (
    idarea INT PRIMARY KEY AUTO_INCREMENT,
    nama_area VARCHAR(100) NOT NULL
);

CREATE TABLE Master_Cabang (
    idcabang INT PRIMARY KEY AUTO_INCREMENT,
    nama_cabang VARCHAR(100) NOT NULL,
    idarea INT NOT NULL,
    FOREIGN KEY (idarea) REFERENCES Master_Area(idarea) ON DELETE CASCADE
);

```

Hasilnya :



Yang dinormalisasi :

Perbaikan yang dilakukan:

1. Normalisasi Nama Field:

- Menggunakan snake_case agar lebih mudah dibaca dan seragam.

2. Keamanan Password:

- Mengubah password VARCHAR(6) menjadi VARCHAR(255), karena umumnya password disimpan dalam bentuk hash (misal dengan bcrypt).

3. Enum untuk Status User & Status Menu:

- Menggunakan ENUM('active', 'inactive') untuk membatasi nilai yang mungkin pada status_user dan status_menu.

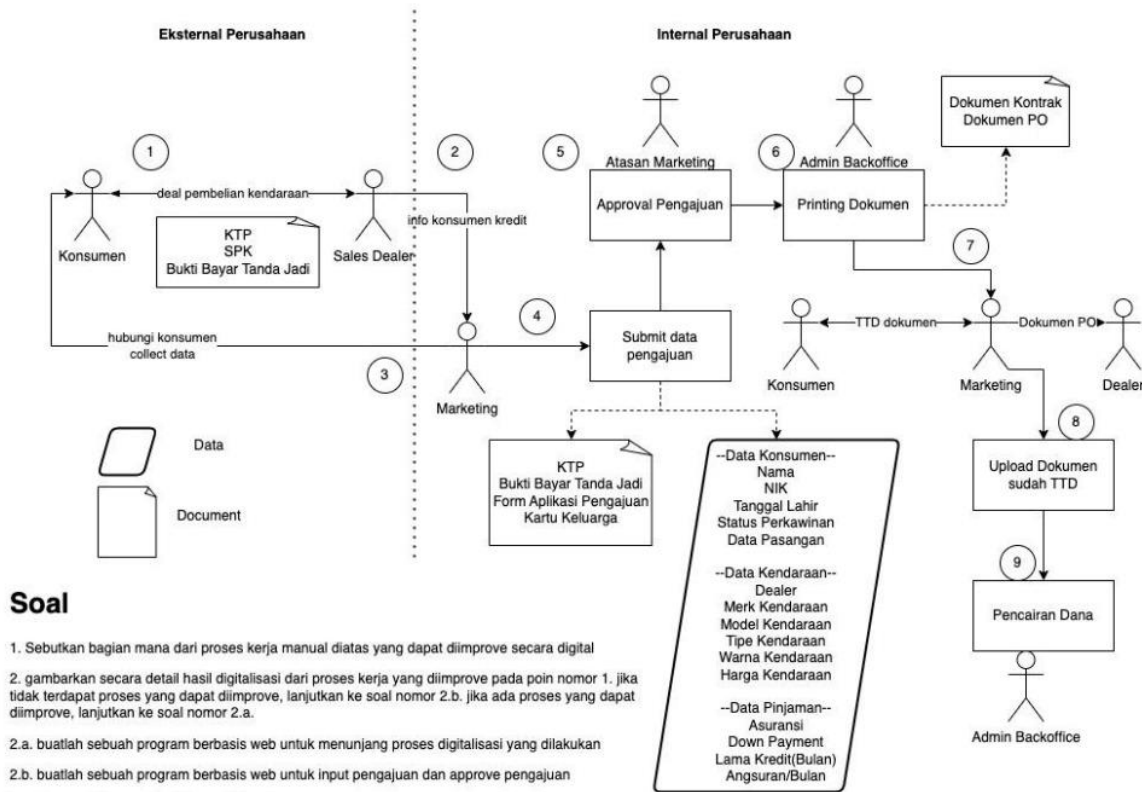
4. Penyesuaian Relasi:

- Menambahkan kunci asing (FOREIGN KEY) pada created_by, idgroup, idcabang, dan idarea untuk menjaga integritas referensial.
- Menambahkan relasi parent_menu untuk mendukung hierarki menu.

5. Kebijakan Hapus (ON DELETE CASCADE/SET NULL):

- Jika idmenu dihapus, sub-menu tidak akan ikut hilang (SET NULL).
- Jika idgroup atau idmenu dihapus, data di Mapping_GroupMenu juga ikut terhapus (CASCADE).
- Jika idarea dihapus, maka seluruh cabang di area tersebut juga ikut terhapus.

Perusahaan perkreditan PT. JKL, memiliki proses penerimaan pengajuan kredit kendaraan bermotor dengan flow seperti gambar dibawah. proses kerja tersebut masih dilakukan secara manual dimana pertukaran data dan dokumen dilakukan secara fisik(fotokopi, scan, print dokumen)



Jawaban Soal

1. Bagian dari proses manual yang dapat diimprove secara digital:

- Pengumpulan Data Konsumen dan Kendaraan (Langkah 1, 2, 3, 4)

→ Saat ini masih dilakukan secara manual dengan kertas (KTP, SPK, bukti bayar, form aplikasi). Bisa digantikan dengan sistem digital untuk input data langsung oleh konsumen/dealer.

- Approval Pengajuan oleh Atasan Marketing (Langkah 5)

→ Bisa dibuat sistem otomatis dengan notifikasi digital untuk approval berbasis sistem.

- Printing Dokumen (Langkah 6)

→ Dokumen bisa dibuat dalam bentuk digital untuk mengurangi cetak fisik.

- Tanda Tangan Dokumen (Langkah 7)

→ Menggunakan tanda tangan digital agar lebih cepat dan efisien.

- Upload Dokumen TTD (Langkah 8)

→ Bisa otomatis dilakukan dalam sistem tanpa harus diunggah secara manual.

2. Detail Hasil Digitalisasi dari Proses Kerja yang Diimprove

(Contoh Implementasi Digitalisasi)

A. Platform Web atau Aplikasi Mobile

- Konsumen dapat langsung mengunggah dokumen seperti KTP, SPK, dan Bukti Bayar melalui aplikasi.
- Dealer bisa memasukkan data kendaraan langsung ke sistem.

B. Sistem Approval Digital

- Marketing dan atasan dapat langsung memberikan approval melalui sistem tanpa perlu dokumen fisik.

C. Tanda Tangan Digital

- Konsumen dan marketing dapat memberikan tanda tangan secara digital tanpa perlu mencetak dokumen.

D. Otomatisasi Upload dan Pencairan Dana

- Setelah semua dokumen disetujui, sistem otomatis memproses pencairan dana ke dealer.

2a. Program Berbasis Web untuk Digitalisasi Proses

Fitur:

Dashboard Admin & User

Form Input Data Konsumen & Kendaraan

Upload Dokumen Digital

Approval Digital oleh Atasan

Tanda Tangan Digital

Sistem Notifikasi & Tracking Status Pengajuan

```
import { useState } from "react";

import Button from "../components/Elements/Button";

import InputForm from "../components/Elements/Input";


function App() {

  // State untuk menyimpan data input form

  const [formData, setFormData] = useState({

    name: "",

    email: "",

    phone: "",

    ktp: "",

    vehicleType: "",

    vehicleModel: "",

    downPayment: "",

    loanTerm: "",

    monthlyInstallment: "",

  });


  // Fungsi untuk menangani perubahan input pada form

  const handleChange = (e) => {

    setFormData({ ...formData, [e.target.name]: e.target.value });

  };


  // Fungsi untuk menangani submit form
```

```

const handleSubmit = (e) => {

  e.preventDefault(); // Mencegah reload halaman

  console.log("Submitted Data:", formData); // Menampilkan data di console
};

return (

  <div className="flex justify-center min-h-screen items-center">

    <div className="w-full max-w-md p-6 bg-white shadow-md rounded-md">

      <h1 className="text-3xl font-bold mb-2 text-blue-600">

        Pengajuan Kredit Kendaraan

      </h1>

      <p className="font-medium text-slate-500 mb-4">

        Silakan isi data berikut untuk mengajukan kredit kendaraan.

      </p>

      <form onSubmit={handleSubmit}>

        {/* Input Nama Lengkap */}

        <InputForm

          label="Nama Lengkap"

          type="text"

          name="name"

          value={formData.name}

          onChange={handleChange}

        />

        {/* Input Email */}

```

```
<InputForm
  label="Email"
  type="email"
  name="email"
  value={formData.email}
  onChange={handleChange}
/>
```

```
{/* Input Nomor HP */}
```

```
<InputForm
  label="Nomor HP"
  type="text"
  name="phone"
  value={formData.phone}
  onChange={handleChange}
/>
```

```
{/* Input Nomor KTP */}
```

```
<InputForm
  label="Nomor KTP"
  type="text"
  name="ktp"
  value={formData.ktp}
  onChange={handleChange}
/>
```

```
{/* Input Tipe Kendaraan */}
```

```
<InputForm

  label="Tipe Kendaraan"

  type="text"

  name="vehicleType"

  value={formData.vehicleType}

  onChange={handleChange}

/>

{/* Input Model Kendaraan */}

<InputForm

  label="Model Kendaraan"

  type="text"

  name="vehicleModel"

  value={formData.vehicleModel}

  onChange={handleChange}

/>

{/* Input Down Payment */}

<InputForm

  label="Down Payment (DP)"

  type="number"

  name="downPayment"

  value={formData.downPayment}

  onChange={handleChange}

/>

{/* Input Lama Kredit dalam bulan */}
```

```

<InputForm
  label="Lama Kredit (bulan)"
  type="number"
  name="loanTerm"
  value={formData.loanTerm}
  onChange={handleChange}
/>

{/* Input Angsuran per Bulan */}

<InputForm
  label="Angsuran per Bulan"
  type="number"
  name="monthlyInstallment"
  value={formData.monthlyInstallment}
  onChange={handleChange}
/>

{/* Tombol Submit */}

<Button variant="bg-blue-600 w-full">Ajukan Kredit</Button>

</form>

</div>

</div>

);
}

export default App;

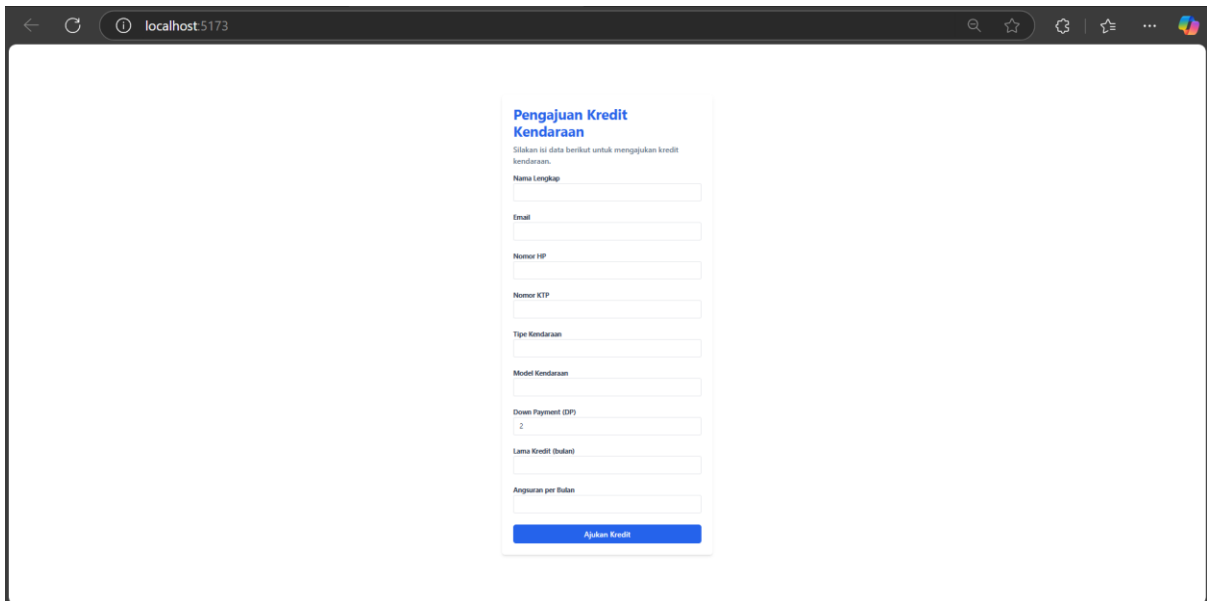
```

```

1 import { useState } from "react";
2 import Button from "../components/Elements/Button";
3 import InputForm from "../components/Elements/Input";
4
5 function App() {
6   // State untuk menyimpan data input form
7   const [formData, setFormData] = useState({
8     name: "",
9     email: "",
10    phone: "",
11    ktp: "",
12    vehicleType: "",
13    vehicleModel: "",
14    downPayment: "",
15    loanTerm: "",
16    monthlyInstallment: "",
17  });
18
19   // Fungsi untuk menangani perubahan input pada form
20   const handleChange = (e) => {
21     setFormData({ ...formData, [e.target.name]: e.target.value });
22   };
23
24   // Fungsi untuk menangani submit form
25   const handleSubmit = (e) => {
26     e.preventDefault(); // Mencegah reload halaman
27     console.log("Submitted Data:", formData); // Menampilkan data di console
28   };
29
30   return (
31     <div className="flex justify-center min-h-screen items-center">
32       <div className="w-full max-w-md p-6 bg-white shadow-md rounded-md">
33         <h1 className="text-3xl font-bold mb-2 text-blue-600">
34           Pengajuan Kredit Kendaraan
35         </h1>
36         <p className="font-medium text-slate-500 mb-4">
37           Silakan isi data berikut untuk mengajukan kredit kendaraan.
38         </p>
39         <form onSubmit={handleSubmit}>
40           <!-- Input Nama Lengkap -->
41           <InputForm
42             label="Nama Lengkap"
43             type="text"
44             name="name"
45             value={formData.name}
46             onChange={handleChange}
47           />
48           <!-- Input Email -->
49           <InputForm
50             label="Email"
51             type="email"
52             name="email"
53             value={formData.email}
54             onChange={handleChange}
55           />
56           <!-- Input Nomor HP -->
57           <InputForm
58             label="Nomor HP"
59             type="text"
60             name="phone"
61             value={formData.phone}
62             onChange={handleChange}
63           />
64           <!-- Input Nomor KTP -->
65           <InputForm
66             label="Nomor KTP"
67             type="text"
68             name="ktp"
69             value={formData.ktp}
70             onChange={handleChange}
71           />
72           <!-- Input Tipe Kendaraan -->
73           <InputForm
74             label="Tipe Kendaraan"
75             type="text"
76             name="vehicleType"
77             value={formData.vehicleType}
78             onChange={handleChange}
79           />
80           <!-- Input Model Kendaraan -->
81           <InputForm
82             label="Model Kendaraan"
83             type="text"
84             name="vehicleModel"
85             value={formData.vehicleModel}
86             onChange={handleChange}
87           />
88           <!-- Input Down Payment -->
89           <InputForm
90             label="Down Payment (DP)"
91             type="number"
92             name="downPayment"
93             value={formData.downPayment}
94             onChange={handleChange}
95           />
96           <!-- Input Lama Kredit dalam bulan -->
97           <InputForm
98             label="Lama Kredit (bulan)"
99             type="number"
100            name="loanTerm"
101            value={formData.loanTerm}
102            onChange={handleChange}
103          />
104          <!-- Input Angsuran per Bulan -->
105          <InputForm
106            label="Angsuran per Bulan"
107            type="number"
108            name="monthlyInstallment"
109            value={formData.monthlyInstallment}
110            onChange={handleChange}
111          />
112          <!-- Tombol Submit -->
113          <Button variant="bg-blue-600 w-full">Ajukan Kredit</Button>
114        </form>
115      </div>
116    );
117  }
118 }
119
120 export default App;

```

Tampilan :



The screenshot shows a web browser window with the address bar displaying 'localhost:5173'. The main content area features a form titled 'Pengajuan Kredit Kendaraan'. Below the title is a subtitle: 'Silakan isi data berikut untuk mengajukan kredit kendaraan.' The form contains several input fields: 'Nama Lengkap', 'Email', 'Nomor HP', 'Nomor KTP', 'Tipe Kendaraan', 'Model Kendaraan', 'Down Payment (DP)' (with the value '2' entered), 'Lama Kredit (bulan)', and 'Angsuran per Bulan'. At the bottom of the form is a blue button labeled 'Ajukan Kredit'.

2b. Program Berbasis Web untuk Input Pengajuan dan Approve Pengajuan

Fitur:

Input Pengajuan

Konsumen atau dealer bisa mengajukan kredit kendaraan melalui web.

Approval Pengajuan

Atasan marketing dapat langsung melihat pengajuan dan memberikan approval secara digital.

Dashboard Status Pengajuan

Konsumen dan dealer bisa melihat status pengajuan real-time.

Integrasi dengan Sistem Pembayaran

Untuk pencairan dana otomatis setelah semua dokumen lengkap.

Slide presentasi :

Digitalisasi Proses Pengajuan Kredit Kendaraan PT JKL

Slide 1: Judul

- Digitalisasi Proses Pengajuan Kredit Kendaraan PT JKL
 - Nama: [Isi Nama]
 - Tanggal: [Isi Tanggal]
-

Slide 2: Pendahuluan

- PT JKL masih menggunakan proses manual dalam pengajuan kredit kendaraan.
 - Pertukaran data dan dokumen dilakukan secara fisik (fotokopi, scan, print).
 - Digitalisasi diperlukan untuk meningkatkan efisiensi dan mengurangi risiko kesalahan.
-

Slide 3: Alur Proses Manual Saat Ini

1. Konsumen membeli kendaraan dan mengumpulkan dokumen.
 2. Dealer mengirim informasi kredit ke marketing.
 3. Marketing menghubungi konsumen untuk mengumpulkan data.
 4. Marketing mengajukan data ke atasan untuk approval.
 5. Atasan marketing menyetujui pengajuan.
 6. Admin backoffice mencetak dokumen.
 7. Konsumen menandatangani dokumen fisik.
 8. Dokumen yang sudah ditandatangani diunggah ke sistem.
 9. Dana dicairkan oleh admin backoffice.
-

Slide 4: Masalah dalam Proses Manual

- Proses lama dan tidak efisien.
 - Risiko kehilangan atau kerusakan dokumen fisik.
 - Ketergantungan pada proses cetak dan tanda tangan fisik.
 - Sulitnya tracking status pengajuan secara real-time.
-

Slide 5: Solusi Digitalisasi

1. **Input Data Digital** – Konsumen dan dealer dapat mengisi formulir secara online.
 2. **Upload Dokumen Digital** – Dokumen seperti KTP, SPK, dan bukti pembayaran diunggah langsung.
 3. **Sistem Approval Digital** – Atasan marketing dapat menyetujui pengajuan tanpa dokumen fisik.
 4. **Tanda Tangan Digital** – Konsumen dan marketing dapat menandatangani dokumen secara elektronik.
 5. **Dashboard Real-Time** – Semua pihak dapat melihat status pengajuan secara langsung.
 6. **Otomasi Pencairan Dana** – Sistem dapat menghubungkan pencairan dana dengan status persetujuan.
-

Slide 6: Implementasi Program Berbasis Web

- **Fitur:**
 - Dashboard untuk Konsumen, Dealer, Marketing, dan Admin.
 - Form input pengajuan kredit.
 - Upload dokumen digital.
 - Approval digital oleh marketing dan atasan.
 - Sistem notifikasi untuk status pengajuan.

- Integrasi dengan sistem pencairan dana.

Slide	7:	Contoh	Tampilan	Sistem
--------------	-----------	---------------	-----------------	---------------

(Tampilkan wireframe atau contoh UI sederhana untuk sistem berbasis web)

Slide 8: Kesimpulan

- Digitalisasi akan mempercepat proses pengajuan kredit.
- Mengurangi ketergantungan pada dokumen fisik.
- Memudahkan tracking status pengajuan.
- Meningkatkan efisiensi kerja di PT JKL.

Slide 9: Q&A

- Tanya Jawab