

UJIAN AKHIR SEMESTER

APLIKASI PERENCANAAN NASABAH (REACT-NATIVE)

Mata Kuliah : Mobile Programming

Dosen : Drs. Dwi Atmodjo WP, M.KOM



PERBANAS INSTITUTE

Dikerjakan oleh:

Nama/HP : Arvito Ramadhanu

NIM : 1914000034

Kelas : Teknik Informatika '19 (Kelas Reguler)

FAKULTAS TEKNOLOGI INFORMASI
PERBANAS INSTITUTE
KAMPUS BEKASI

INSTRUKSI

Sebuah perusahaan leasing kendaraan bermotor berkeinginan membuat sebuah aplikasi untuk menghitung kredit yang disalurkan berdasarkan kebutuhan dari nasabah. Aplikasi ini akan didistribusikan secara umum dimana setiap nasabah yang akan mengajukan kredit dapat menghitung terlebih dahulu angsuran, bunga dan berapa kali angsuran yang harus dipenuhi oleh nasabah sampai kreditnya lunas. Adapun formula yang diberlakukan oleh perusahaan ini adalah sebagai berikut.

Uang muka yang harus dibayar adalah 30% dari nilai kredit sesuai dengan aturan yang berlaku. Suku bunga kredit mobil yang dikenakan adalah 4.8% per tahun (flat) dan jangka waktu yang diambil adalah 5 tahun. Sementara, untuk biaya asuransi dikenakan sebesar 5%, biaya provisi sebesar 0,5%, biaya polis asuransi sebesar Rp 40 ribu dan biaya administrasi sebesar Rp 700 ribu.

- $\text{Plafon pinjaman} = \text{Harga Kendaraan} - \text{Uang Muka}$
- $\text{Jumlah cicilan per bulan} = \text{Angsuran pokok per bulan} + \text{angsuran bunga per bulan}$
- $\text{Angsuran pokok per bulan} = \text{Plafon pinjaman} / \text{Tenor (jumlah bulan pinjaman)}$
- $\text{Angsuran bunga per bulan} = (\text{Plafon pinjaman} \times \text{Suku bunga}) / 12$
- $\text{Pembayaran pertama kali} = \text{Uang muka} + \text{Angsuran pertama} + \text{Asuransi (Persen dari harga Mobil)} + \text{Provisi (Persen dari plafon pinjaman)} + \text{Polis Asuransi} + \text{Administrasi}$

Buatlah aplikasi menggunakan **React-Native** untuk membantu nasabah membuat rencana pembelian kendaraan bermotor sesuai kesiapan yang dimilikinya.

Dibawah ini adalah contoh simulasi hutungan kredit yang berlaku jika harga kendaraan bermotor sebesar Rp.150.000.000,- dengan tenor 60 bulan.

Plafon pinjaman: $\text{Rp } 150.000.000 - \text{Rp } 45.000.000 \text{ (DP 30\% dari Harga Mobil)} = \text{Rp } 105.000.000$

Angsuran pokok per bulan : $\text{Rp } 105.000.000 / 60 \text{ bulan} = \text{Rp } 1.750.000$

Angsuran bunga per bulan : $(\text{Rp } 105.000.000 \times 4.8\%) / 12 = \text{Rp } 420.000$

Jumlah cicilan per bulan : $\text{Rp } 1.750.000 + \text{Rp } 420.000 = \text{Rp } 2.170.000$

Pembayaran pertama kali : $\text{Rp } 45.000.000 + \text{Rp } 2.170.000 + \text{Rp } 7.500.000 \text{ (5\% dari Rp } 150.000.000) + \text{Rp } 525.000 \text{ (0.5\% dari } \text{Rp } 105.000.000) + \text{Rp } 40.000 + \text{Rp } 700.000 = \text{Rp } 55.935.000$

Jadi, jumlah pembayaran pertama yang harus dilakukan oleh Andi adalah Rp 55.935.000 dan cicilan per bulannya sebesar Rp2.170.000 selama 5 tahun.

Kriteria Penilaian

- 65 Jika Source-code dan screen capture hasil menampilkan jawaban yang diinginkan.
- 75 Jika Source-code dan screen capture hasil menampilkan jawaban yang diinginkan dan orisinal (tidak ada yang mirip dengan jawaban rekan sekelas).
- 85 Jika Source-code dan screen capture hasil menampilkan jawaban yang diinginkan dan menambahkan fitur atau library lain untuk mendukung aplikasi yang dibuat.
- 95 Jika Source-code dan screen capture hasil menampilkan jawaban yang diinginkan dan menambahkan fitur atau library lain untuk mendukung aplikasi yang dibuat dan menyertakan file “apk” pada saat upload jawaban.

LANGKAH Pengerjaan

- [illegible]

```
npx react-native init vitoKredit
npx react-native start
npx react-native run-android
```

- The image shows a development environment with Visual Studio Code on the left and a mobile emulator on the right. The VS Code editor is open to the file `App.js` in the `vitoKredit` project. The code is a React Native component that calculates insurance costs. It includes comments in Indonesian explaining the variables and the calculation process. The code defines a `constructor` with initial state values for `harga`, `bulan`, `hasil`, `cicilanBulanan`, `bayaranPertama`, `hitung`, `hargaMuka`, `plafonPinjaman`, `biayaAsuransi`, `provisi`, `pokokBulanan`, and `bungaBulanan`. It also includes a `hitung` function that calculates the insurance cost based on the input values.

The mobile emulator on the right shows the app's UI, which is a grid of 16 app icons. The icons are arranged in a 4x4 grid and include various standard Android app icons such as Photos, Expo Go, Gmail, Maps, Settings, Calculator, Calendar, Camera, Chrome, Clock, Contacts, Custom, Dev Tools, Email, Expo Go, Files, Gmail, Google, Maps, Messages, Phone, Photos, Play Mov., Play Mus., and Settings.

3. Berikut hasil akhir yang saya peroleh. Saya hanya menampilkan hasil akhir atau konklusi yang diminta. Untuk rumusan lainnya dikalkulasi pada background dan tidak ditampilkan.

The image shows two screenshots of a mobile application titled "VitoKredit | Hitung Pembayaran".

Left Screenshot (Initial State):

- Header: VitoKredit | Hitung Pembayaran
- Input field: Harga Kendaraan: (empty)
- Input field: Jumlah Bulan: (empty)
- Button: Hitung Pembayaran
- Output: Pembayaran Pertama: 0, Cicilan Per Bulan: 0

Right Screenshot (Calculated State):

- Header: VitoKredit | Hitung Pembayaran
- Input field: Harga Kendaraan: 150000000
- Input field: Jumlah Bulan: 60
- Button: Hitung Pembayaran
- Output: Pembayaran Pertama: 55935000, Cicilan Per Bulan: 2170000

Dibawah ini adalah contoh simulasi hutungan kredit yang berlaku jika harga kendaraan bermotor sebesar Rp.150.000.000,- dengan tenor 60 bulan.

Plafon pinjaman : Rp 150.000.000 – Rp 45.000.000 (DP 30% dari Harga Mobil)
= Rp 105.000.000

Angsuran pokok per bulan : Rp 105.000.000 / 60 bulan = Rp 1.750.000

Angsuran bunga per bulan : (Rp 105.000.000 x 4.8%) / 12 = Rp 420.000

Jumlah cicilan per bulan : Rp 1.750.000 + Rp 420.000 = Rp 2.170.000

Pembayaran pertama kali : Rp 45.000.000 + Rp 2.170.000 + Rp 7.500.000 (5% dari Rp 150.000.000) + Rp 525.000 (0.5% dari Rp 105.000.000) + Rp 40.000 + Rp 700.000 = Rp 55.935.000

Jadi, jumlah pembayaran pertama yang harus dilakukan oleh Andi adalah Rp 55.935.000 dan cicilan per bulannya sebesar Rp2.170.000 selama 5 tahun.

4. Untuk penjelasan algoritma, rumusan, dan variable saya jelaskan pada source code dengan comment.
5. Video singkat demo aplikasi dapat dilihat pada link berikut:
<https://imgur.com/aYmJC4B>

6. Saya membuat signing key melalui CMD (admin privilege) dengan menggunakan Java Development Kit (navigasi: G:\Program Files\Java\jdk1.8.0_181\bin) Lalu, memasukkan perintah **keytool -genkeypair -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000** Kemudian saya memasukkan data saya seperti yang diminta oleh form.

```
Administrator: Command Prompt
What is your first and last name?
[Unknown]: Arvito Ramadhanu
What is the name of your organizational unit?
[Unknown]: TI
What is the name of your organization?
[Unknown]: Perbanas
What is the name of your City or Locality?
[Unknown]: Bekasi
What is the name of your State or Province?
[Unknown]: Jawa Barat
What is the two-letter country code for this unit?
[Unknown]: id
Is CN=Arvito Ramadhanu, OU=TI, O=Perbanas, L=Bekasi, ST=Jawa Barat, C=id correct?
[no]: y
Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=Arvito Ramadhanu, OU=TI, O=Perbanas, L=Bekasi, ST=Jawa Barat, C=id
Enter key password for <my-key-alias>
(RETURN if same as keystore password):
Re-enter new password:
[Storing my-release-key.keystore]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format
using "keytool -importkeystore -srckeystore my-release-key.keystore -destkeystore my-release-key.keystore -deststoretype
pkcs12".
```

7. Saya melakukan perubahan pada file gradle.properties dengan memasukkan:

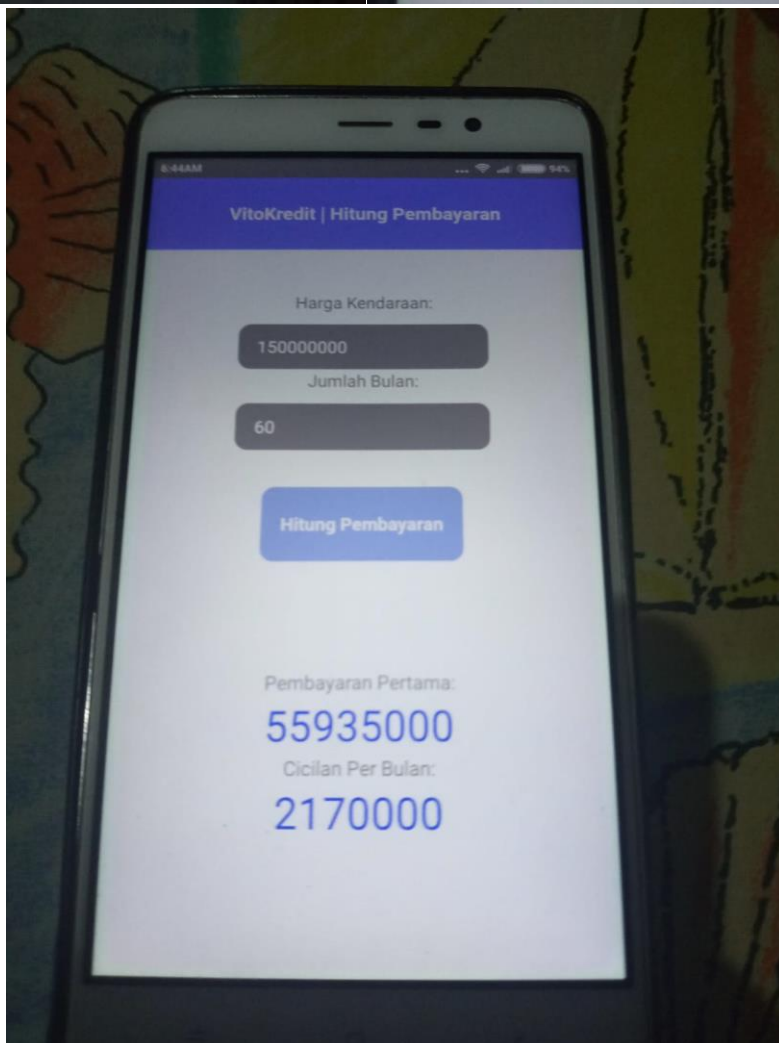
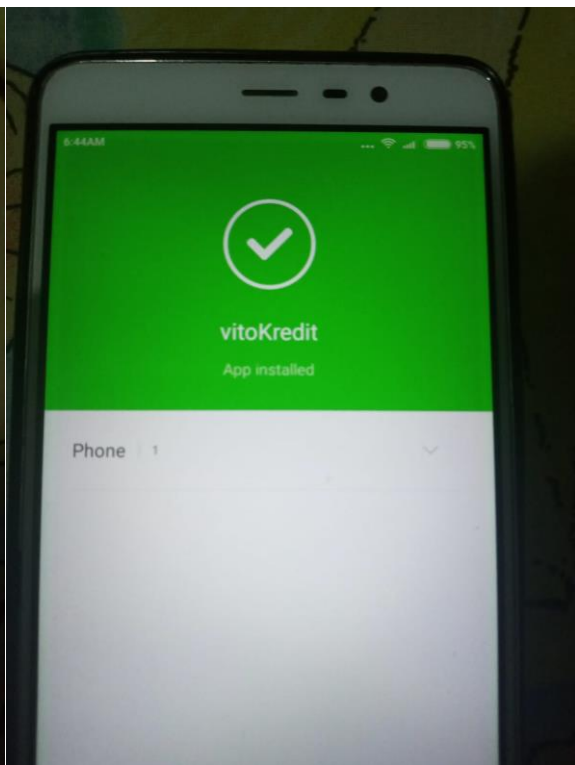
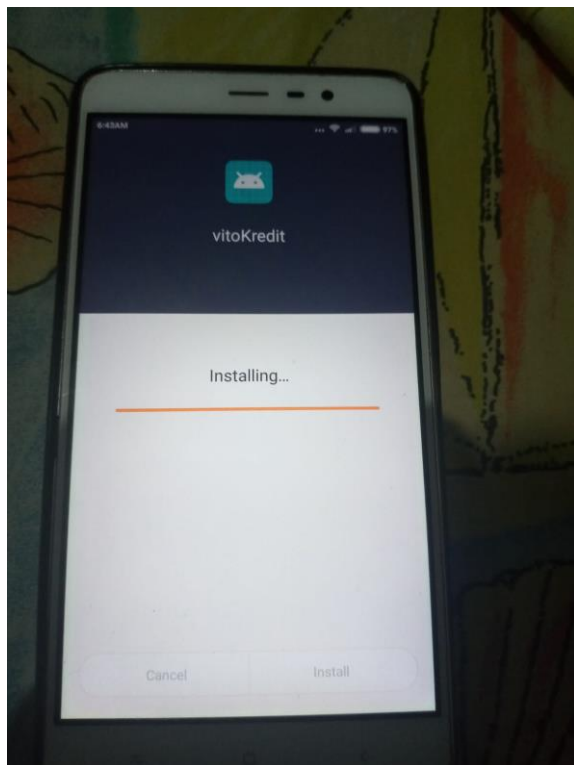
```
JS App.js  gradle.properties x  build.gradle
android > gradle.properties
29
30 MYAPP_RELEASE_STORE_FILE=my-release-key.keystore
31 MYAPP_RELEASE_KEY_ALIAS=CN=my-key-alias
32 MYAPP_RELEASE_STORE_PASSWORD=kilimanjaro
33 MYAPP_RELEASE_KEY_PASSWORD=kilimanjaro
```

8. Saya melakukan perubahan pada file build.gradle dengan memasukkan:

```
signingConfigs {
    debug {
        storeFile file('my-release-key.keystore')
        storePassword 'kilimanjaro'
        keyAlias 'my-key-alias'
        keyPassword 'kilimanjaro'
    }
}
```

Kemudian menjalankan gradlew assembleRelease untuk membangun APK.

```
D:\>cd D:\Windows 10 User\Alex\Documents\reactnative project\vitoKredit\android
D:\Windows 10 User\Alex\Documents\reactnative project\vitoKredit\android>gradlew assembleRelease
BUILD SUCCESSFUL in 2m 27s
38 actionable tasks: 5 executed, 33 up-to-date
D:\Windows 10 User\Alex\Documents\reactnative project\vitoKredit\android>
```



SOURCE CODE

1. App.js

```
import React, { Component } from 'react';
import { View, Text, TextInput, TouchableOpacity } from 'react-native';

class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      // Variabel dari soal yang memiliki nilai awal
      sukuBunga:0.048,
      dp:0.3,
      persenAsuransi:0.05,
      polisAsuransi:0.005,
      biayaPolis:40000,
      admin:700000,

      // Variabel input
      harga:0,
      bulan:0,
      hasil:0,

      //Variabel untuk menghitung rumusan
      hargaMuka:0,
      plafonPinjaman:0,
      biayaAsuransi:0,
      provisi:0,
      pokokBulanan:0,
      bungaBulanan:0,

      // Hasil yang ingin ditampilkan
      cicilanBulanan:0,
      bayaranPertama:0,
    };
  }

  hitung = ()=>{
    // menghitung rumusan
    let menghitungHargaMuka =
parseFloat(this.state.harga)*parseFloat(this.state.dp);
    this.setState({hargaMuka:menghitungHargaMuka})
    let menghitungPlafonPinjaman = parseFloat(this.state.harga)-
parseFloat(menghitungHargaMuka);
    this.setState({plafonPinjaman:menghitungPlafonPinjaman})
    let menghitungBiayaAsuransi =
parseFloat(this.state.harga)*parseFloat(this.state.persenAsuransi);
    this.setState({biayaAsuransi:menghitungBiayaAsuransi})
```



```

    let menghitungProvisi =
parseFloat(menghitungPlafonPinjaman)*parseFloat(this.state.polisAsuransi);
    this.setState({provisi:menghitungProvisi})
    let menghitungPokokBulanan =
parseFloat(menghitungPlafonPinjaman)/parseFloat(this.state.bulan);
    this.setState({pokokBulanan:menghitungPokokBulanan})
    let menghitungBungaBulanan =
(parseFloat(menghitungPlafonPinjaman)*parseFloat(this.state.sukuBunga))/12;
    this.setState({bungaBulanan:menghitungBungaBulanan})

    // menghitung hasil
    let menghitungCicilanBulanan =
parseFloat(menghitungPokokBulanan)+parseFloat(menghitungBungaBulanan);
    this.setState({cicilanBulanan:menghitungCicilanBulanan})
    let menghitungBayaranPertama =
parseFloat(menghitungHargaMuka)+parseFloat(menghitungCicilanBulanan)+parseFloa
t(menghitungBiayaAsuransi)+parseFloat(menghitungProvisi)+parseFloat(this.state
.biayaPolis)+parseFloat(this.state.admin);
    this.setState({bayaranPertama:menghitungBayaranPertama})
  }

  render() {
    return (

      <View style={{flex:1}}>
        <View style={{
          flex:1,
          zIndex: 100,
          justifyContent:'center',
          alignItems:'center',
          elevation:5,
          backgroundColor:'#7f7cF1',
        }}>

          <Text style={{color:'#ffffff',fontWeight:'bold',
fontSize:16}}>VitoKredit | Hitung Pembayaran</Text>
        </View>

        <View style={{
          flex:8,
          justifyContent:'center',
          alignItems:'center',
          elevation:5,
          backgroundColor:'#ffffff',
        }}>

          <Text>Harga Kendaraan:</Text>

```



```

<TextInput
  value={this.state.harga}
  style={{
    marginHorizontal:20,
    marginTop:10,
    width:200,
    paddingVertical:10,
    paddingHorizontal:15,
    backgroundColor:'#7f7c82',
    color:'#fff',
    borderRadius:10
  }}
  keyboardType='number-pad'
  onChangeText={(value) => this.setState({harga:value})}
/>

<Text>Jumlah Bulan:</Text>

<TextInput
  value={this.state.bulan}
  style={{
    marginHorizontal:20,
    marginTop:10,
    width:200,
    paddingVertical:10,
    paddingHorizontal:15,
    backgroundColor:'#7f7c82',
    color:'#fff',
    borderRadius:10
  }}
  keyboardType='number-pad'
  onChangeText={(value) => this.setState({bulan:value})}
/>

<TouchableOpacity
  onPress={() => this.hitung()}
  style={{
    backgroundColor:'#8fa2db',
    marginTop:30,
    marginHorizontal:50,
    paddingHorizontal:15,
    paddingVertical:20,
    borderRadius:10
  }}>
  <Text
    style={{color:'#fff',textAlign:'center',fontWeight:'bold'}}>Hitung
    Pembayaran</Text>

```

```

    </TouchableOpacity>

    <View style={{
      marginHorizontal:50,
      marginTop:30,
      justifyContent:'center',
      alignContent:'center',
      paddingVertical:50
    }}>
      <Text style={{textAlign:'center'}}>Pembayaran Pertama:</Text>
      <Text
style={{color:'#0000FF',fontSize:30,textAlign:'center'}}>{this.state.bayaranPe
rtama}</Text>
      <Text style={{textAlign:'center'}}>Cicilan Per Bulan:</Text>
      <Text
style={{color:'#0000FF',fontSize:30,textAlign:'center'}}>{this.state.cicilanBu
lanan}</Text>
    </View>

  </View>

</View>
);
}
}

export default App;

```