UTS Pengembangan Aplikasi Mobile

Nama	: Christian
NIM	: 120140056
Kelas	: RB

Source Code

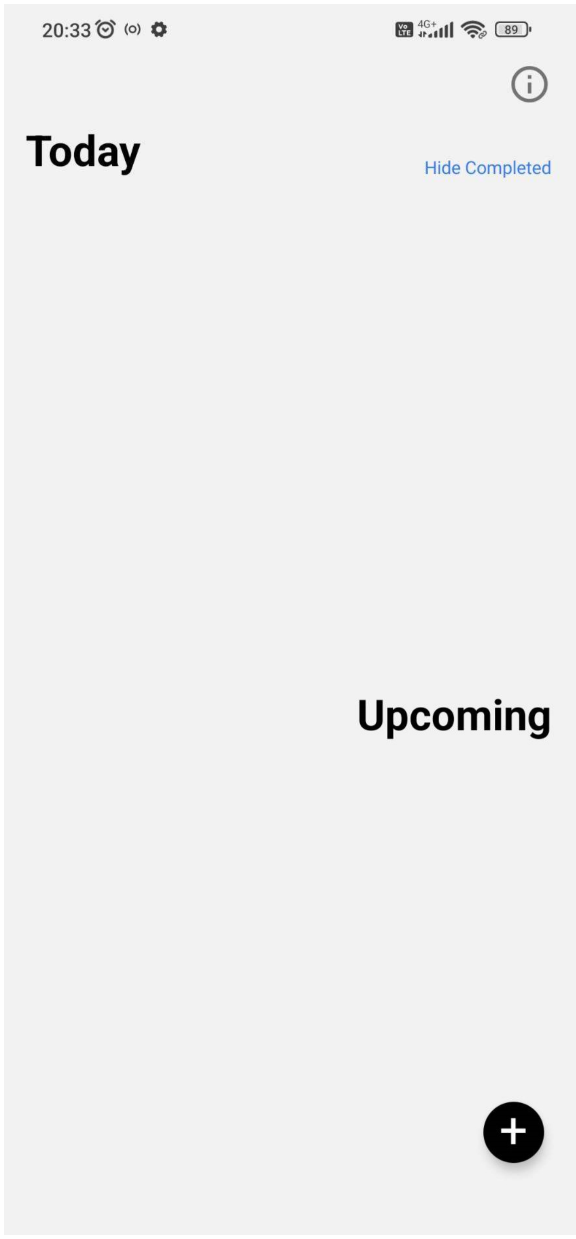
```
import * as React from 'react';
import { StyleSheet, Text, View, FlatList, SafeAreaView, TouchableOpacity, Image,
TextInput, Switch, Alert} from 'react-native';
import { Entypo } from '@expo/vector-icons';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import { useNavigation } from '@react-navigation/native';
import DateTimePicker from '@react-native-community/datetimepicker';
import { createSlice } from '@reduxjs/toolkit';
import { configureStore } from '@reduxjs/toolkit';
import { Provider } from 'react-redux';
import { useDispatch, useSelector } from 'react-redux';
import AsyncStorage from '@react-native-async-storage/async-storage';
import moment from 'moment';
import { MaterialIcons } from '@expo/vector-icons';
import * as Notifications from 'expo-notifications';
import * as Device from 'expo-device';

const Stack = createNativeStackNavigator();

Notifications.setNotificationHandler({
  handleNotification: async () => ({
      shouldShowAlert: true,
      shouldPlaySound: true,
      shouldSetBadge: true,
  }),
})

const initialState = {
  todos: [],
};
const todosSlice = createSlice({
  name: 'todos',
  initialState,
  reducers: {
    setTodosReducers:(state, action) => {
      state.todos = action.payload
      console.log(state.todos);
    },
    addTodoReducer:(state, action) => {
      state.todos.push(action.payload)
    },
    hideCompletedReducer:(state) => {
      state.todos = state.todos.filter(todo => !todo.isCompleted)
    },
    updateTodoReducer: (state, action) => {
      state.todos = state.todos.map(todo => {
        if(todo.id === action.payload.id) {
          todo.isCompleted = !todo.isCompleted
        }
        return todo;
      })
    },
    deleteTodoReducer: (state, action) => {
      const id = action.payload;
      const todos = state.todos.filter(todo => todo.id !== id);
      state.todos = todos;
    }
  },
});
const { setTodosReducers, addTodoReducer, updateTodoReducer, hideCompletedReducer,
deleteTodoReducer } = todosSlice.actions;
const todosReducer = todosSlice.reducer;
const store = configureStore({
  reducer: {
    todos: todosReducer,
  }
});

function Checkbox ({
  id,
  text,
  isCompleted,
  isToday,
  hour
}) {
  const dispatch = useDispatch();
  const listTodos = useSelector(state => state.todos.todos);

  const handleCheckbox = () => {
    try {
      dispatch(updateTodoReducer({id, isCompleted}));
      AsyncStorage.setItem('@Todos', JSON.stringify(
        listTodos.map(todo => {
          if(todo.id === id) {
            return {...todo, isCompleted: !todo.isCompleted}
          }
          return todo;
        })
      ))
      console.log('todo saved correctly');
    } catch (e) {
      console.log(e);
    }
  }
  return isToday ? (
    <TouchableOpacity onPress={handleCheckbox} style={isCompleted ? styles.CheckboxChecked
: styles.CheckboxUnchecked}>
```

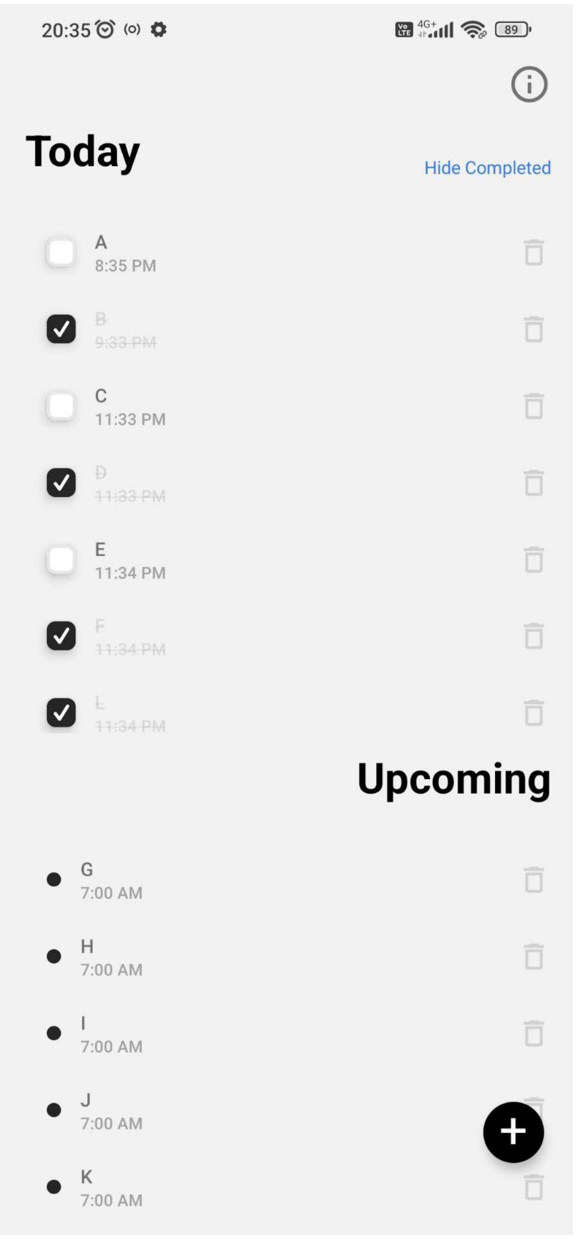
```
      {isCompleted && <Entypo name="check" size={16} color='#FAFAFA' />}
    </TouchableOpacity>
  ) : (
    <View style={styles.CheckboxIsToday}></View>
  )
}

function Todo({
  id,
  text,
  isCompleted,
  isToday,
  hour
}) {
  const [localHour, setLocalHour] = React.useState(new Date(hour));
  const todos = useSelector(state => state.todos.todos);
  const dispatch = useDispatch();

  const handleDeleteTodo = async () => {
    dispatch(deleteTodoReducer(id));
    try {
      await AsyncStorage.setItem('@Todos', JSON.stringify(
        todos.filter(todo => todo.id !== id)
      ));
      console.log('Todo deleted correctly');
    } catch (e) {
      console.log(e);
    }
  };

  return (
    <View style={styles.TodoContainer}>
      <View style={{flexDirection: 'row', alignItems: 'center'}}>
        <Checkbox
          id={id}
          text={text}
          isCompleted={isCompleted}
          isToday={isToday}
          hour={hour}
        />
        <View>
          <Text style={isCompleted ? [styles.TodoText, {textDecorationLine: 'line-
through', color: '#73737330'}] : styles.TodoText}>{text}</Text>
          <Text style={isCompleted ? [styles.TodoTime, {textDecorationLine: 'line-
through', color: '#73737330'}] : styles.TodoTime}>{moment(localHour).format('LT')}</Text>
        </View>
      </View>
      <TouchableOpacity onPress={handleDeleteTodo}>
        <MaterialIcons name='delete-outline' size={24} color='#73737340' />
      </TouchableOpacity>
    </View>
  )
}
```

```
function TodoList( {todosData}) {
    return (
        <FlatList
            data={todosData}
            keyExtractor={item => item.id.toString()}
            renderItem={({item}) => <Todo {...item}/>}
        />
    )
}

function Home() {
  const todos = useSelector(state => state.todos.todos);
  const [isHidden, setIsHidden] = React.useState(false);
  const [expoPushToken, setExpoPushToken] = React.useState('');
  const navigation = useNavigation();
  const dispatch = useDispatch();

  React.useEffect(() => {
    registerForPushNotificationsAsync().then(token => setExpoPushToken(token));
    const getTodos = async () => {
      try {
        const todos = await AsyncStorage.getItem('@Todos');
        if(todos !== null) {
          dispatch(setTodosReducers(JSON.parse(todos)));
        }
      } catch (e) {
        console.log(e);
      }
    }
    getTodos();
  }, []);

  const handleHidePress = async () => {
    if (isHidden) {
      setIsHidden(false);
      const todos = await AsyncStorage.getItem('@Todos');
      if (todos !== null) {
        dispatch(setTodosReducers(JSON.parse(todos)));
      }
      return;
    }
    setIsHidden(true);
    dispatch(hideCompletedReducer());
  }

  const registerForPushNotificationsAsync = async () => {
    let token;
    if (Device.isDevice) {
      const { status: existingStatus } = await Notifications.getPermissionsAsync();
      let finalStatus = existingStatus;
      if (existingStatus !== 'granted') {
        const { status } = await Notifications.requestPermissionsAsync();
        finalStatus = status;
      }
      if (finalStatus !== 'granted') {
        alert('Failed to get push token for push notification!');
        return;
      }
      token = (await Notifications.getExpoPushTokenAsync({projectId: '68d9c76c-4802-4da5-
9c51-be2281aa8c79'})).data;
      console.log(token);
    } else {
        return;
    }
    if (Platform.OS === 'android') {
      Notifications.setNotificationChannelAsync('default', {
        name: 'default',
        importance: Notifications.AndroidImportance.MAX,
        vibrationPattern: [0, 250, 250, 250],
        lightColor: '#FF231F7C',
      });
    }
    return token;
}

  return (
    <SafeAreaView style={styles.HomeContainer}>
      <TouchableOpacity onPress={() => navigation.navigate('Info Menu')}>
        <MaterialIcons name='info-outline' size={30} color='#737373'
style={styles.HomeInfo} />
      </TouchableOpacity>
      <View style={{flexDirection: 'row', alignItems:'center', justifyContent: 'space-
between'}}>
      <Text style={styles.HomeTitle}>Today</Text>
        <TouchableOpacity onPress={handleHidePress}>
          <Text style={{color: '#3478f6'}}>{isHidden ? "Show Completed" : "Hide
Completed"}</Text>
        </TouchableOpacity>
      </View>

      <TodoList todosData={todos.filter(todo => todo.isToday)}/>

      <Text style={[styles.HomeTitle, {alignSelf: 'flex-end'}]}>Upcoming</Text>
      <TodoList todosData={todos.filter(todo => !todo.isToday)}/>
      <TouchableOpacity onPress={() => navigation.navigate('Add Menu')}
style={styles.HomeButton}>
        <Text style={styles.HomeButtonPlus}>+</Text>
      </TouchableOpacity>
    </SafeAreaView>
  )
}

function AddTodo() {
  const [name, setName] = React.useState('');
  const [date, setDate] = React.useState(new Date());
  const [isToday, setIsToday] = React.useState(false);
  const [withAlert, setWithAlert] = React.useState(false);
  const [showPicker, setShowPicker] = React.useState(false);
  const [Temp, setTemp] = React.useState(true);
  const listTodos = useSelector(state => state.todos.todos);
  const dispatch = useDispatch();
  const navigation = useNavigation();

  const addTodo = async () => {
    const newTodo = {
      id: Math.floor(Math.random() * 1000000),
      text : name,
      hour: isToday ? date.toString() : new Date(date).getDate() + 24 * 60 * 60 * 1000,
      isToday: isToday,
      isCompleted: false
    }
    try {
      await AsyncStorage.setItem('@Todos', JSON.stringify([...listTodos, newTodo]));
      dispatch(addTodoReducer(newTodo));
      console.log('Todo saved correctly');
      if(withAlert) {
        await shceduleTodoNotification(newTodo);
      }
      navigation.goBack();
    } catch (e) {
      console.log(e);
    }
  }

  function handleDateChange(event, selectedDate) {
    const currentDate = selectedDate || date;
    setShowPicker(false);
    setDate(currentDate);
  }

  function handlePress() {
    setShowPicker(true);
    setTemp(false);
  }

  const shceduleTodoNotification = async (todo) => {
    //const trigger = todo.isToday ? todo.hour : new Date(todo.hour).getTime() + 24 * 60 *
60 * 1000;
    const trigger = new Date(todo.hour);
    try {
      await Notifications.scheduleNotificationAsync({
        content: {
          title: "Alert! You have a task to do!",
          body: todo.text,
        },
        trigger,
      });
      console.log('notification was scheduled');
    } catch (e) {
      alert('The notification failed to schedule, make sure the hour is valid');
    }
  };
  return(
    <View style={styles.AddContainer}>
      <Text style={styles.AddTitle}>New Task</Text>
      <View style={styles.AddInputContainer}>
        <Text style={styles.AddInputTitle}>Name</Text>
        <TextInput
          style={styles.AddTextInput}
          placeholder='Task'
          placeholderTextColor='#00000030'
          onChangeText={(text) => {setName(text)}}
        />
      </View>
      <View style={styles.AddInputContainer}>
        <Text style={styles.AddInputTitle}>Hour</Text>
        <TouchableOpacity onPress={handlePress}>
          <Text>{Temp ? "Set Time" : date.toLocaleTimeString('en-US', { hour12: false,
hour: '2-digit', minute: '2-digit' })}</Text>
        </TouchableOpacity>
        {showPicker && (
          <DateTimePicker
            value={date}
            mode={'time'}
            display="spinner"
            is24Hour={true}
            onChange={handleDateChange}
            onConfirm={() => setShowPicker(false)}
            style={{width: '80%'}}
          />
        )}
      </View>
      <View style={styles.AddInputContainer}>
        <Text style={styles.AddInputTitle}>Today</Text>
        <Switch
          value={isToday}
          onValueChange={(value) => { setIsToday(value) }}
        />
      </View>
      <View style={[styles.AddInputContainer, {alignItems: 'center'}]}>
        <View>
          <Text style={styles.AddInputTitle}>Alert</Text>
          <Text style={{color: '#00000040', fontSize: 12, maxWidth: '85%'}}>You will
receive an alert at the time you set for this reminder</Text>
        </View>
        <Switch
          value={withAlert}
          onValueChange={(value) => { setWithAlert(value) }}
        />
      </View>
      <TouchableOpacity style={styles.AddButton} onPress={addTodo}>
          <Text style={{color: '#fff'}}>Done</Text>
      </TouchableOpacity>
    </View>
  )
```

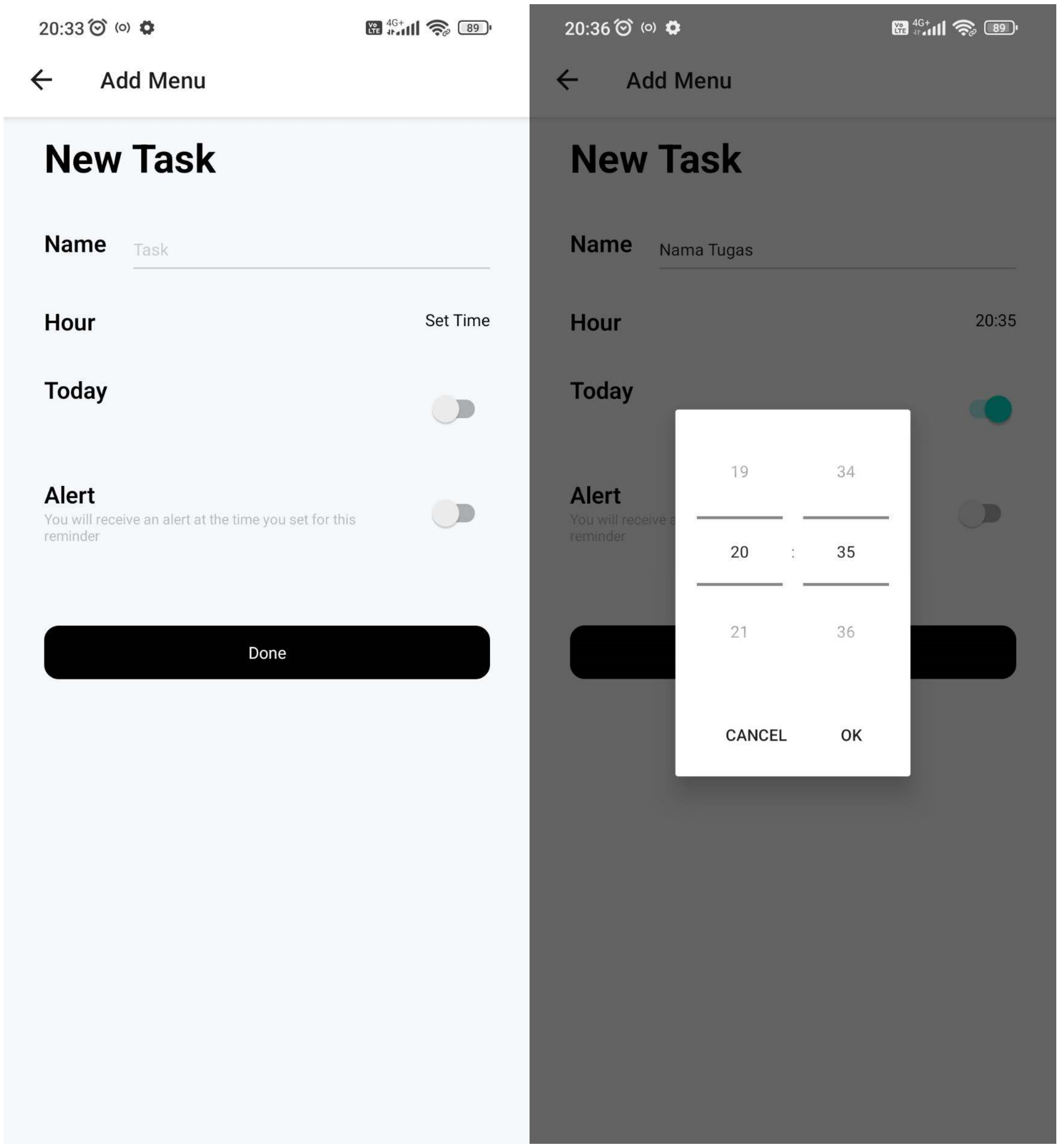
```
}

function Info() {
  return (
    <View style={styles.InfoContainer}>
      <Text style={styles.InfoTitle}>My To Do List</Text>
      <Text>Nama : Christian</Text>
      <Text>NIM : 120140056</Text>
      <Text>Dosen Pengampu : M Habib Algifari, S.Kom., M.T.I.</Text>
      <Text>Waktu Pengerjaan : 21 - 22 Maret 2023</Text>
      <Text></Text>
      <Text>Aplikasi ini dibuat untuk memenuhi Ujian Tengah Semester (UTS) mata kuliah 
Pengembangan Aplikasi Mobile (PAM, IF3026) TA 2022/2023.</Text>
      <Text></Text>
      <Text>GitHub : https://github.com/120140056/UTS-PAM-120140056</Text>
    </View>
  )
}

export default function App() {
  return (
    <Provider store={store}>
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen
            name='Home'
            component={Home}
            options={{headerShown: false}}
          />
          <Stack.Screen
            name='Add Menu'
            component={AddTodo}
            options={{presentation: 'modal'}}
          />
          <Stack.Screen
            name='Info Menu'
            component={Info}
            options={{presentation: 'modal'}}
          />
        </Stack.Navigator>
      </NavigationContainer>
    </Provider>
  );
}

const styles = StyleSheet.create({
  CheckboxChecked: {
    width: 20,
    height: 20,
    marginRight: 13,
    borderRadius: 6,
    backgroundColor: '#262626',
    alignItems: 'center',
    justifyContent: 'center',
    marginLeft: 15,
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: .3,
    shadowRadius: 5,
    elevation: 5,
  },
  CheckboxUnchecked: {
    width: 20,
    height: 20,
    marginRight: 13,
    borderWidth: 2,
    borderColor: '#E8E8E8',
    borderRadius: 6,
    backgroundColor: '#fff',
    marginLeft: 15,
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: .1,
    shadowRadius: 5,
    elevation: 5,
  },
  CheckboxIsToday: {
    width: 10,
    height: 10,
    marginHorizontal: 10,
    borderRadius: 10,
    backgroundColor: '#262626',
    marginRight: 13,
    marginLeft: 15,
  },
  TodoContainer: {
    marginBottom: 20,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between'
  },
  TodoText: {
    fontSize: 15,
    fontWeight: '500',
    color: '#737373'
  },
  TodoTime: {
    fontSize: 13,
    fontWeight: '500',
    color: '#a3a3a3'
  },
  HomeContainer: {
    flex: 1,
    paddingTop: 40,
    paddingHorizontal: 15
  },
  HomeInfo: {
    alignSelf: 'flex-end'
  },
  HomeTitle: {
    fontSize: 34,
    fontWeight: 'bold',
    marginBottom: 35,
    marginTop: 10
  },
  HomeButton: {
    width: 42,
    height: 42,
    borderRadius: 21,
    backgroundColor: '#000',
    position: 'absolute',
    bottom: 50,
    right: 20,
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 2
    },
    shadowOpacity: .5,
    shadowRadius: 5,
    elevation: 5

  },
  HomeButtonPlus: {
    fontSize: 40,
    color: '#fff',
    position: 'absolute',
    top: -5,
    left: 11
  },
  AddContainer: {
    flex: 1,
    backgroundColor: '#f7f8fa',
    paddingHorizontal: 30
  },
  AddTitle: {
    fontSize: 34,
    fontWeight: 'bold',
    marginBottom: 35,
    marginTop:10
  },
  AddInputTitle: {
    fontSize: 20,
    fontWeight: '600',
    lineHeight: 24
  },
  AddTextInput: {
    borderBottomColor: '#00000030',
    borderBottomWidth: 1,
    width: '80%'
  },
  AddInputContainer: {
    justifyContent: 'space-between',
    flexDirection: 'row',
    paddingBottom: 30
  },
  AddButton: {
    marginTop: 30,
    marginBottom: 15,
    alignItems: 'center',
    justifyContent: 'center',
    backgroundColor: '#000',
    height: 40,
    borderRadius: 11
  },
  InfoContainer: {
    flex: 1,
    backgroundColor: '#f7f8fa',
    paddingHorizontal: 30
  },
  InfoTitle: {
    fontSize: 34,
    fontWeight: 'bold',
    marginBottom: 35,
    marginTop:10
  },
});
```

Screenshot





Gambar 1. Halaman Utama


20:33

Add Menu

New Task

Name Task

Hour Set Time

Today

Alert

You will receive an alert at the time you set for this reminder

Done

20:36

Add Menu

New Task

Name Nama Tugas

Hour 20:35

Today

Alert

You will receive a

reminder

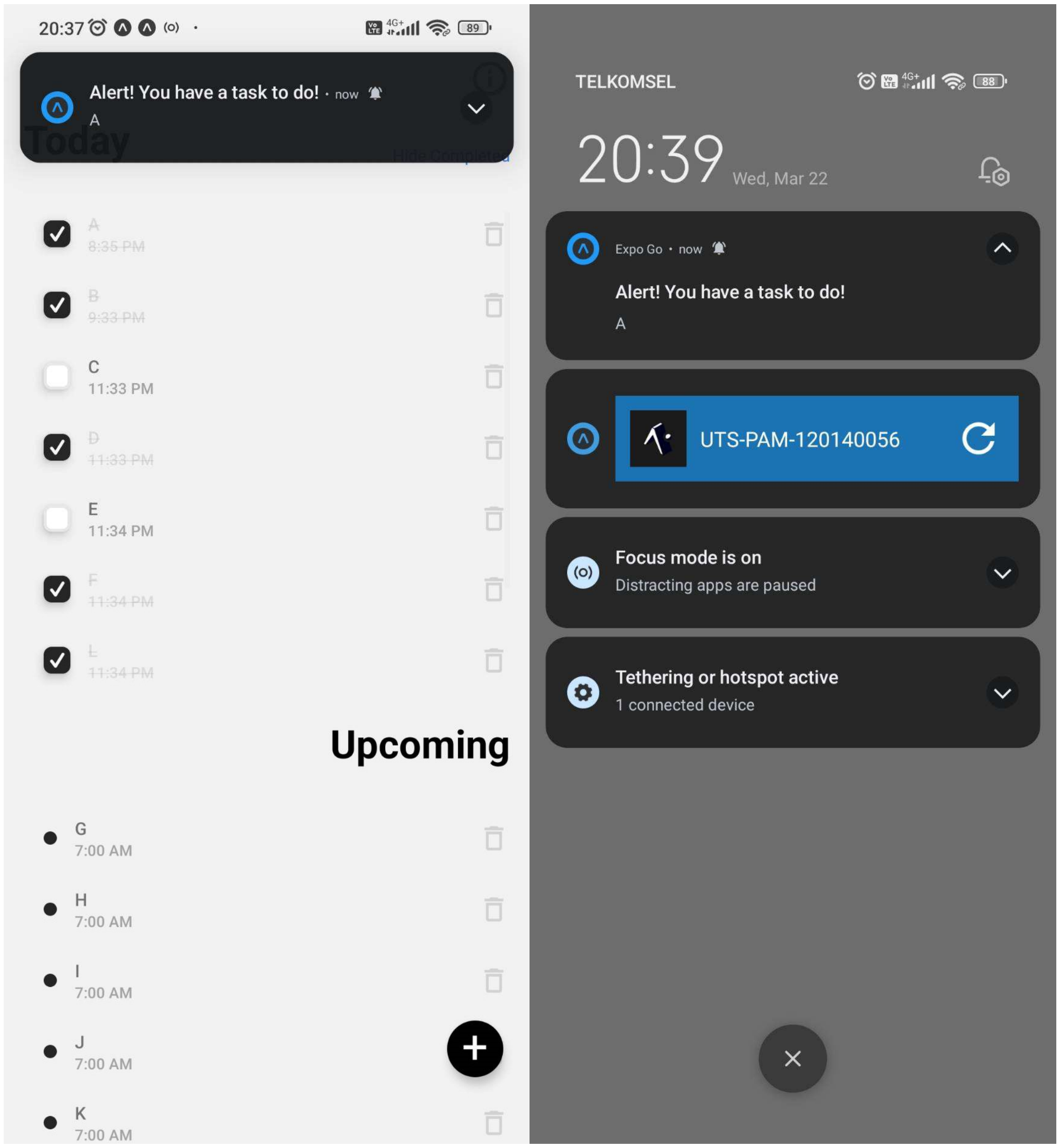
19 34

20 : 35

21 36

CANCEL OK

Gambar 2. Halaman Tambah Tugas


20:37

Alert! You have a task to do! · now

A

Today

Hide Completed

~~A~~
~~8:35 PM~~

~~B~~
~~9:33 PM~~

C
11:33 PM

~~D~~
~~11:33 PM~~

E
11:34 PM

~~F~~
~~11:34 PM~~

~~L~~
~~11:34 PM~~

Upcoming

G
7:00 AM

H
7:00 AM

I
7:00 AM

J
7:00 AM

K
7:00 AM

TELKOMSEL

20:39 Wed, Mar 22

Expo Go · now

Alert! You have a task to do!

A

UTS-PAM-120140056

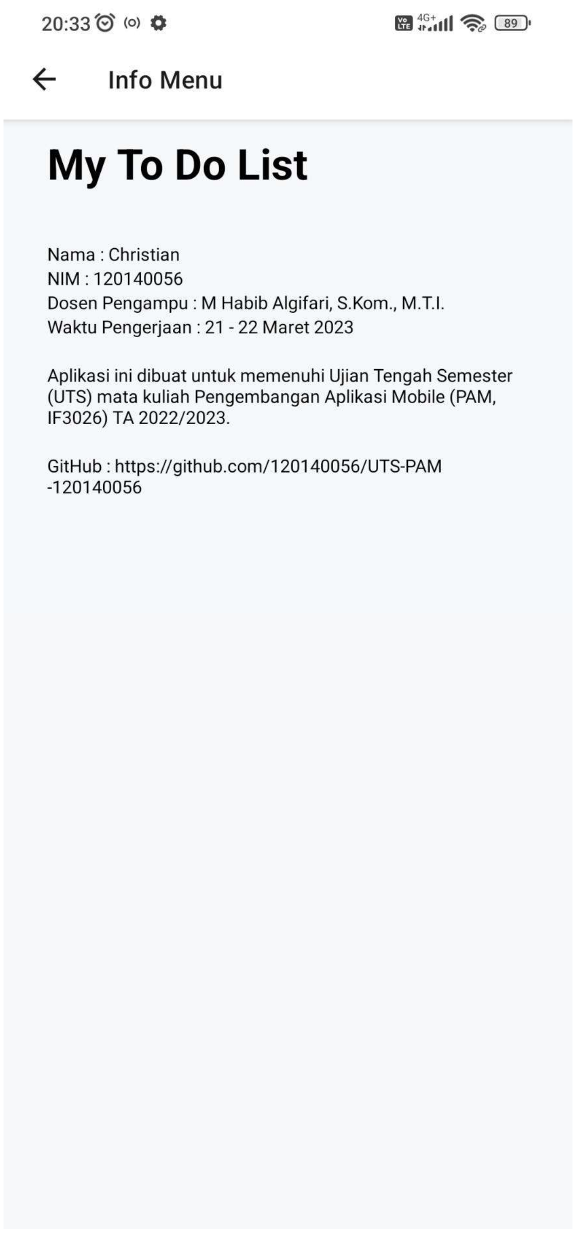
Focus mode is on

Distracting apps are paused

Tethering or hotspot active

1 connected device

Gambar 3. Notifikasi


20:33

Info Menu

My To Do List

Nama : Christian
NIM : 120140056
Dosen Pengampu : M Habib Algifari, S.Kom., M.T.I.
Waktu Pengerjaan : 21 - 22 Maret 2023

Aplikasi ini dibuat untuk memenuhi Ujian Tengah Semester (UTS) mata kuliah Pengembangan Aplikasi Mobile (PAM, IF3026) TA 2022/2023.

GitHub : https://github.com/120140056/UTS-PAM-120140056

Gambar 4. Halaman Profil Mahasiswa

Link GitHub

Link : https://github.com/120140056/UTS-PAM-120140056