activity\_loginregister.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".LoginRegisterActivity">

<FrameLayout

android:id="@+id/frameContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</FrameLayout>

</LinearLayout>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<FrameLayout

android:id="@+id/frameContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</FrameLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottomNav"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

app:menu="@menu/bottom\_navigation"/>

</RelativeLayout>

fragment\_login.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:textSize="14pt"

android:text="Login"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Username"/>

<EditText

android:id="@+id/edtPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="PIN"/>

<Button

android:id="@+id/btnLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Login"/>

</LinearLayout>

fragment\_register.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:textSize="14pt"

android:text="Register"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Username"/>

<EditText

android:id="@+id/edtName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Name"/>

<EditText

android:id="@+id/edtPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="PIN"/>

<Button

android:id="@+id/btnRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Register"/>

</LinearLayout>

fragment\_history.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Jenis Transaksi:"/>

<RadioGroup

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:orientation="horizontal">

<RadioButton

android:id="@+id/rbIn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="6pt"

android:text="Uang Masuk"/>

<RadioButton

android:id="@+id/rbOut"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="6pt"

android:text="Uang Keluar"/>

<RadioButton

android:id="@+id/rbAll"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="6pt"

android:text="Semua"/>

</RadioGroup>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:text="Pencarian bedasarkan username:"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:hint="Username"/>

<Button

android:id="@+id/btnShow"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:text="Tampilkan"/>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rvHistories"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

</androidx.recyclerview.widget.RecyclerView>

</LinearLayout>

fragment\_transfer.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"

android:text="Jenis Transaksi: "/>

<RadioGroup

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:orientation="horizontal">

<RadioButton

android:id="@+id/rbTransfer"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="8pt"

android:text="Transfer Dana"/>

<RadioButton

android:id="@+id/rbRequest"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Request Dana"/>

</RadioGroup>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Masukkan Nominal Dana:"/>

<EditText

android:id="@+id/edtNominal"

android:inputType="number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Masukkan ID Tujuan:"/>

<EditText

android:id="@+id/edtDestination"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<Button

android:id="@+id/btnSubmit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10pt"

android:textSize="10pt"

android:text="Submit"/>

</LinearLayout>

layout\_feeds.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/linearContainer"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/tvDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/tvDateTime"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<Button

android:id="@+id/btnSend"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Send Money"/>

</LinearLayout>

</LinearLayout>

layout\_history.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10pt">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tvDateTime"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<TextView

android:id="@+id/tvMoney"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:gravity="right"

android:textSize="8pt"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="5pt">

<TextView

android:id="@+id/tvDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<TextView

android:id="@+id/tvCode"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:gravity="right"

android:textSize="8pt"/>

</LinearLayout>

</LinearLayout>

options\_menu.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuLogin"

android:title="Login"/>

<item

android:id="@+id/menuRegister"

android:title="Register"/>

<item

android:id="@+id/menuLogout"

android:title="Logout"/>

</menu>

bottom\_navigation.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuHome"

android:title="Home"

android:icon="@drawable/home"/>

<item

android:id="@+id/menuHistory"

android:title="History"

android:icon="@drawable/history"/>

<item

android:id="@+id/menuTransfer"

android:title="Transfer"

android:icon="@drawable/transfer"/>

</menu>

LoginRegisterActivity.java

public class LoginRegisterActivity extends AppCompatActivity {

FrameLayout frameContainer;

Intent page;

ArrayList<User> users;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_loginregister);

frameContainer= findViewById(R.id.frameContainer);

users= new ArrayList<>();

getExtrasData();

showFragment(new LoginFragment());

Toast.makeText(this, users.toString(), Toast.LENGTH\_SHORT).show();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.options\_menu, menu);

menu.findItem(R.id.menuLogout).setVisible(false);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()) {

case R.id.menuLogin:

showFragment(new LoginFragment());

break;

case R.id.menuRegister:

showFragment(new RegisterFragment());

break;

}

return super.onOptionsItemSelected(item);

}

private void showFragment(Fragment fragment) {

Bundle argument= new Bundle();

argument.putSerializable("users", users);

FragmentTransaction trans= getSupportFragmentManager().beginTransaction();

fragment.setArguments(argument);

trans.replace(R.id.frameContainer, fragment);

trans.commit();

}

private void getExtrasData() {

page= getIntent();

if (page.hasExtra("users")) {

users= (ArrayList<User>) page.getSerializableExtra("users");

Toast.makeText(this, users.toString(), Toast.LENGTH\_SHORT).show();

}

}

}

MainActivity.java

public class MainActivity extends AppCompatActivity {

FrameLayout frameContainer;

BottomNavigationView bottomNav;

Intent page;

ArrayList<User> users;

int logged;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

frameContainer= findViewById(R.id.frameContainer);

bottomNav= findViewById(R.id.bottomNav);

getExtrasData();

showFragment(new HomeFragment());

bottomNav.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

switch (menuItem.getItemId()) {

case R.id.menuHome:

showFragment(new HomeFragment());

break;

case R.id.menuHistory:

showFragment(new HistoryFragment());

break;

case R.id.menuTransfer:

showFragment(new TransferFragment());

break;

}

return true;

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.options\_menu, menu);

menu.findItem(R.id.menuLogin).setVisible(false);

menu.findItem(R.id.menuRegister).setVisible(false);

menu.findItem(R.id.menuLogout).setVisible(true);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

if (item.getItemId() == R.id.menuLogout) {

page= new Intent(MainActivity.this, LoginRegisterActivity.class);

page.putExtra("users", users);

startActivity(page);

}

return super.onOptionsItemSelected(item);

}

private void getExtrasData() {

page= getIntent();

if (page.hasExtra("users")) {

users= (ArrayList<User>) page.getSerializableExtra("users");

Toast.makeText(this, users.toString(), Toast.LENGTH\_SHORT).show();

}

if (page.hasExtra("logged")) {

logged= page.getIntExtra("logged", -1);

Toast.makeText(this, logged+"", Toast.LENGTH\_SHORT).show();

}

}

private void showFragment(Fragment fragment) {

Bundle argument= new Bundle();

argument.putSerializable("users", users);

argument.putInt("logged", logged);

FragmentTransaction trans= getSupportFragmentManager().beginTransaction();

fragment.setArguments(argument);

trans.replace(R.id.frameContainer, fragment);

trans.commit();

}

}

LoginFragment.java

public class LoginFragment extends Fragment {

EditText edtUsername, edtPIN;

Button btnLogin;

Intent page;

ArrayList<User> users;

int logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_login, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

edtUsername= view.findViewById(R.id.edtUsername);

edtPIN= view.findViewById(R.id.edtPIN);

btnLogin= view.findViewById(R.id.btnLogin);

users= (ArrayList<User>) getArguments().getSerializable("users");

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username= edtUsername.getText().toString();

String PIN= edtPIN.getText().toString();

if (!username.isEmpty() && !PIN.isEmpty()) {

logged= -1;

for (int i= 0; i<users.size(); i++) {

User user= users.get(i);

if (user.getUsername().equals(username) && user.getPIN().equals(PIN)) {

logged= i;

break;

}

}

if (logged >= 0) {

page= new Intent(getContext(), MainActivity.class);

page.putExtra("users", users);

page.putExtra("logged", logged);

startActivity(page);

Toast.makeText(getContext(), "Login success!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Username/PIN wrong!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

Toast.makeText(view.getContext(), users.toString(), Toast.LENGTH\_SHORT).show();

}

}

RegisterFragment.java

public class RegisterFragment extends Fragment {

EditText edtUsername, edtName, edtPIN;

Button btnRegister;

ArrayList<User> users;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_register, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

edtUsername= view.findViewById(R.id.edtUsername);

edtName= view.findViewById(R.id.edtName);

edtPIN= view.findViewById(R.id.edtPIN);

btnRegister= view.findViewById(R.id.btnRegister);

users= (ArrayList<User>) getArguments().getSerializable("users");

btnRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username= edtUsername.getText().toString();

String name= edtName.getText().toString();

String PIN= edtPIN.getText().toString();

if (!username.isEmpty() && !name.isEmpty() && !PIN.isEmpty()) {

boolean exist= false;

for (User user : users) {

if (user.getUsername().equals(username)) {

exist= true;

break;

}

}

if (!exist) {

users.add(new User(name, username, PIN));

refreshField();

Toast.makeText(getContext(), "Register success!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Username already used!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

Toast.makeText(view.getContext(), users.toString(), Toast.LENGTH\_SHORT).show();

}

private void refreshField() {

edtUsername.setText("");

edtName.setText("");

edtPIN.setText("");

}

}

HomeFragment.java

public class HomeFragment extends Fragment {

TextView tvLogged, tvBalance;

EditText edtAmount;

Button btnTopUp;

RecyclerView rvFeeds;

ArrayList<User> users;

FeedsAdapter adapter;

int logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_home, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

tvLogged= view.findViewById(R.id.tvLogged);

tvBalance= view.findViewById(R.id.tvBalance);

edtAmount= view.findViewById(R.id.edtAmount);

btnTopUp= view.findViewById(R.id.btnTopUp);

rvFeeds= view.findViewById(R.id.rvFeeds);

users= (ArrayList<User>) getArguments().getSerializable("users");

logged= getArguments().getInt("logged", -1);

adapter= new FeedsAdapter(users.get(logged).getFeeds());

rvFeeds.setHasFixedSize(true);

rvFeeds.setLayoutManager(new LinearLayoutManager(getContext()));

rvFeeds.setAdapter(adapter);

tvLogged.setText("Welcome, "+users.get(logged).getName()+"!");

tvBalance.setText("Your Balance: "+users.get(logged).getCurrencyFormat());

btnTopUp.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int amount= 0;

if (!edtAmount.getText().toString().isEmpty()) amount= Integer.parseInt(edtAmount.getText().toString());

if (amount > 0) {

users.get(logged).setBalance(amount);

tvBalance.setText("Your Balance: "+users.get(logged).getCurrencyFormat());

Toast.makeText(getContext(), "To up of "+amount+" success!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

adapter.setOnItemClickCallback(new FeedsAdapter.OnItemClickCallback() {

@Override

public void onItemClicked(int position) {

User destination= null;

for (User user : users) {

if (user.getUsername().equals(users.get(logged).getFeeds().get(position).getDestination().getUsername())) {

destination= user;

break;

}

}

if (destination != null) {

if (users.get(logged).getBalance() > 0) {

users.get(logged).setBalance(-users.get(logged).getFeeds().get(position).getNominal());

users.get(logged).setHistory(new Transaction(destination, users.get(logged).getFeeds().get(position).getNominal(), "send"));

destination.setBalance(users.get(logged).getFeeds().get(position).getNominal());

destination.setFeeds(new Transaction(users.get(logged), users.get(logged).getFeeds().get(position).getNominal(), "send"));

destination.setHistory(new Transaction(users.get(logged), users.get(logged).getFeeds().get(position).getNominal(), "request"));

users.get(logged).getFeeds().remove(position);

adapter.notifyDataSetChanged();

tvBalance.setText("Your Balance: "+users.get(logged).getBalance());

} else Toast.makeText(getContext(), "Not enough balance!", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

TransactionFragment.java

public class TransferFragment extends Fragment {

RadioButton rbTransfer, rbRequest;

EditText edtNominal, edtDestination;

Button btnSubmit;

ArrayList<User> users;

int logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_transfer, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

rbTransfer= view.findViewById(R.id.rbTransfer);

rbRequest= view.findViewById(R.id.rbRequest);

edtNominal= view.findViewById(R.id.edtNominal);

edtDestination= view.findViewById(R.id.edtDestination);

btnSubmit= view.findViewById(R.id.btnSubmit);

users= (ArrayList<User>) getArguments().getSerializable("users");

logged= getArguments().getInt("logged", -1);

btnSubmit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String type= "";

if (rbTransfer.isChecked()) type= "send";

else if (rbRequest.isChecked()) type= "request";

if (!type.isEmpty() && !edtNominal.getText().toString().isEmpty() && !edtDestination.getText().toString().isEmpty()) {

User destination= null;

for (User user : users) {

if (user.getName().equals(edtDestination.getText().toString()) && !user.getName().equals(users.get(logged).getName())) {

destination= user;

break;

}

}

if (destination != null) {

int nominal= Integer.parseInt(edtNominal.getText().toString());

if (type.equals("send")) {

if (users.get(logged).getBalance() > 0) {

users.get(logged).setBalance(-nominal);

users.get(logged).setHistory(new Transaction(destination, nominal, type));

destination.setBalance(nominal);

destination.setFeeds(new Transaction(users.get(logged), nominal, type));

destination.setHistory(new Transaction(users.get(logged), nominal, "request"));

Toast.makeText(getContext(), "Transfer to "+destination.getName()+" success!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Not enough balance!", Toast.LENGTH\_SHORT).show();

} else if (type.equals("request")) {

destination.setFeeds(new Transaction(users.get(logged), nominal, type));

Toast.makeText(getContext(), "Requested to "+destination.getName(), Toast.LENGTH\_SHORT).show();

}

} else Toast.makeText(getContext(), "ID not found!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "All field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

}

}

FeedsAdapter.java

public class FeedsAdapter extends RecyclerView.Adapter<FeedsAdapter.FeedsViewHolder> {

private ArrayList<Transaction> feeds;

private OnItemClickCallback onItemClickCallback;

public FeedsAdapter(ArrayList<Transaction> feeds) {

this.feeds = feeds;

}

public void setOnItemClickCallback(OnItemClickCallback onItemClickCallback) { this.onItemClickCallback = onItemClickCallback; }

@NonNull

@Override

public FeedsAdapter.FeedsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new FeedsAdapter.FeedsViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.layout\_feeds, parent, false));

}

@Override

public void onBindViewHolder(@NonNull FeedsAdapter.FeedsViewHolder holder, final int position) {

Transaction transaction= this.feeds.get(position);

if (transaction.getType().equals("send")) {

holder.linearContainer.setBackgroundColor(Color.GREEN);

holder.tvDescription.setText(transaction.getDestination().getName()+" telah mengirimkan dana sebesar "+transaction.getCurrencyFormat());

holder.btnSend.setVisibility(View.INVISIBLE);

} else if (transaction.getType().equals("request")) {

holder.linearContainer.setBackgroundColor(Color.YELLOW);

holder.tvDescription.setText(transaction.getDestination().getName()+" meminta dana sebesar "+transaction.getCurrencyFormat());

}

holder.btnSend.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (onItemClickCallback != null) {

onItemClickCallback.onItemClicked(position);

}

}

});

holder.tvDateTime.setText(transaction.getDateTime());

}

@Override

public int getItemCount() {

return this.feeds.size();

}

public class FeedsViewHolder extends RecyclerView.ViewHolder {

LinearLayout linearContainer;

TextView tvDescription, tvDateTime;

Button btnSend;

public FeedsViewHolder(@NonNull View itemView) {

super(itemView);

this.linearContainer= itemView.findViewById(R.id.linearContainer);

this.tvDescription= itemView.findViewById(R.id.tvDescription);

this.tvDateTime= itemView.findViewById(R.id.tvDateTime);

this.btnSend= itemView.findViewById(R.id.btnSend);

}

}

public interface OnItemClickCallback { void onItemClicked(int position); }

}

HistoryAdapter.java

public class HistoryAdapter extends RecyclerView.Adapter<HistoryAdapter.HistoryViewHolder>{

private ArrayList<Transaction> histories;

public HistoryAdapter(ArrayList<Transaction> histories) {

this.histories = histories;

}

@NonNull

@Override

public HistoryAdapter.HistoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new HistoryViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.layout\_history, parent, false));

}

@Override

public void onBindViewHolder(@NonNull HistoryAdapter.HistoryViewHolder holder, int position) {

Transaction transaction= this.histories.get(position);

holder.tvDateTime.setText(transaction.getDateTime());

holder.tvMoney.setText(transaction.getCurrencyFormat());

if (transaction.getType().equals("send")) {

holder.tvDescription.setText("Transfer ke "+transaction.getDestination().getName());

holder.tvCode.setText("DB");

holder.tvMoney.setTextColor(Color.RED);

holder.tvCode.setTextColor(Color.RED);

} else if (transaction.getType().equals("request")) {

holder.tvDescription.setText("Uang masuk dari "+transaction.getDestination().getName());

holder.tvCode.setText("CR");

holder.tvMoney.setTextColor(Color.GREEN);

holder.tvCode.setTextColor(Color.GREEN);

}

}

@Override

public int getItemCount() {

return this.histories.size();

}

public class HistoryViewHolder extends RecyclerView.ViewHolder {

TextView tvDateTime, tvMoney, tvDescription, tvCode;

public HistoryViewHolder(@NonNull View itemView) {

super(itemView);

this.tvDateTime= itemView.findViewById(R.id.tvDateTime);

this.tvMoney= itemView.findViewById(R.id.tvMoney);

this.tvDescription= itemView.findViewById(R.id.tvDescription);

this.tvCode= itemView.findViewById(R.id.tvCode);

}

}

}

User.java

public class User implements Serializable {

private String name, username, PIN;

private int balance;

private ArrayList<Transaction> feeds, histories;

public User(String name, String username, String PIN) {

this.name = name;

this.username = username;

this.PIN = PIN;

this.feeds= new ArrayList<>();

this.histories= new ArrayList<>();

}

@Override

public String toString() {

return this.username+" - "+this.PIN;

}

public String getCurrencyFormat() {

DecimalFormat currency= (DecimalFormat) DecimalFormat.getCurrencyInstance();

DecimalFormatSymbols rupiah= new DecimalFormatSymbols();

rupiah.setCurrencySymbol("Rp ");

rupiah.setMonetaryDecimalSeparator(',');

rupiah.setGroupingSeparator('.');

currency.setDecimalFormatSymbols(rupiah);

return currency.format(this.balance);

}

public String getName() { return this.name; }

public String getUsername() { return this.username; }

public String getPIN() { return this.PIN; }

public int getBalance() { return this.balance; }

public void setBalance(int balance) { this.balance += balance; }

public ArrayList<Transaction> getFeeds() { return this.feeds; }

public void setFeeds(Transaction feed) { this.feeds.add(feed); }

public ArrayList<Transaction> getHistories() { return this.histories; }

public void setHistory(Transaction history) { this.histories.add(history); }

}

Transaction.java

public class Transaction implements Serializable {

private User destination;

private int nominal;

private String type, dateTime;

public Transaction(User destination, int nominal, String type) {

this.destination = destination;

this.nominal = nominal;

this.type = type;

this.dateTime= this.getDateTimeFormat();

}

@Override

public String toString() {

return this.destination+" - "+this.nominal+" - "+this.type;

}

public String getCurrencyFormat() {

DecimalFormat currency= (DecimalFormat) DecimalFormat.getCurrencyInstance();

DecimalFormatSymbols rupiah= new DecimalFormatSymbols();

rupiah.setCurrencySymbol("Rp ");

rupiah.setMonetaryDecimalSeparator(',');

rupiah.setGroupingSeparator('.');

currency.setDecimalFormatSymbols(rupiah);

return currency.format(this.nominal);

}

private String getDateTimeFormat() {

Date date= Calendar.getInstance().getTime();

DateFormat formatter= new SimpleDateFormat("dd MMM, hh:mm");

return formatter.format(date);

}

public User getDestination() { return this.destination; }

public int getNominal() { return this.nominal; }

public String getType() { return this.type; }

public String getDateTime() { return this.dateTime; }

}