

LAMPIRAN

1. Lampiran data *source* JSON

```
{
  "Chat" : {
    "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
      "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
        "seen" : true,
        "timestamp" : 1550924424948
      }
    },
    "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
      "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
        "seen" : false,
        "timestamp" : 1550924424977
      }
    }
  },
  "Friends" : {
    "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
      "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
        "date" : "Mar 3, 2019 04:58:04"
      }
    },
    "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
      "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
        "date" : "Mar 3, 2019 04:58:04"
      }
    }
  },
  "Notifications" : {
    "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
      "-L_-PM3rQ0K0YAoun-Yw" : {
        "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
        "type" : "request"
      }
    }
  },
  "Users" : {
    "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
      "device_token" : "dwZVXdR56B0:APA91bG-
h_4t5bSIZzyFeh5DEQrG6iiGPuTTdr2qPODS6zMXg6Qs3kZNOhMTF6dVUujNwoJKiO7puCC
M51vbUd-rSFyZd3XXZaGZNqH6Cu1Mmn5yJGzLji7HxtLwY_bb6huBZD27J0YBK",
      "image" : "https://firebasestorage.googleapis.com/v0/b/chatkuy-
45752.appspot.com/o/profile_images%2FnCb30bnOr0OGVpgJMhYmMTnstAj1.jpg?alt=media&t
oken=7a3b654d-cf9e-42dd-b487-9abf3dac53f7",
      "name" : "Aulia Rifqi Nurahman",
      "online" : 1551564077277,
      "status" : "Android Developer",
      "thumb_image" : "https://firebasestorage.googleapis.com/v0/b/chatkuy-
45752.appspot.com/o/profile_images%2FnCb30bnOr0OGVpgJMhYmMTnstAj1.jpg?alt=media&t
oken=7a3b654d-cf9e-42dd-b487-9abf3dac53f7"
```

```

},
  "wkrPBD5frLZmxLdeZjUSrgTNTToh1" : {
    "device_token" : "dwZVXdR56B0:APA91bG-
h_4t5bSIZzyFeh5DEQrG6iiGPuTTdr2qPODS6zMXg6Qs3kZNOhMTF6dVUujNwoJKiO7puCC
M51vbUd-rSFyzd3XZaGZNqH6Cu1Mmn5yJGzLji7HxtLwY_bb6huBZD27J0YBK",
    "image" : "https://firebasestorage.googleapis.com/v0/b/chatkuy-
45752.appspot.com/o/profile_images%2FwkrPBD5frLZmxLdeZjUSrgTNTToh1.jpg?alt=media&to
ken=307eac76-0804-431e-ae34-3f02c1e860e1",
    "name" : "Ari Nurahman",
    "online" : 1551563893476,
    "status" : "pusing..pusing..pusing",
    "thumb_image" : "https://firebasestorage.googleapis.com/v0/b/chatkuy-
45752.appspot.com/o/profile_images%2FwkrPBD5frLZmxLdeZjUSrgTNTToh1.jpg?alt=media&to
ken=307eac76-0804-431e-ae34-3f02c1e860e1"
  }
},
  "messages" : {
    "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
      "wkrPBD5frLZmxLdeZjUSrgTNTToh1" : {
        "-LYcMHZrotR8J-19R4p9" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "test",
          "seen" : false,
          "time" : 1550086648219,
          "type" : "text"
        },
        "-LYcMJ5xn4RiirYEdRsY" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "12345",
          "seen" : false,
          "time" : 1550086652758,
          "type" : "text"
        },
        "-LYcMJf_9XgVcNFJGazJ" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "1",
          "seen" : false,
          "time" : 1550086655090,
          "type" : "text"
        },
        "-LYcMJqThca1CTPzEMxD" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "2",
          "seen" : false,
          "time" : 1550086655768,
          "type" : "text"
        },
        "-LYcMJy2Ti9E43eFr_Vf" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "3",
          "seen" : false,
          "time" : 1550086656257,
          "type" : "text"
        }
      }
    }
  }
}

```

```

    },
    "wkrPBD5frLZmxLdeZjUSrgTNToh1" : {
      "nC30bnOr0OGVpgJMhYmMTnstAj1" : {
        "-LYcMHzrotR8J-19R4p9" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "test",
          "seen" : false,
          "time" : 1550086648219,
          "type" : "text"
        },
        "-LYcMJ5xn4RiirYEdRsY" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "12345",
          "seen" : false,
          "time" : 1550086652758,
          "type" : "text"
        },
        "-LYcMJf_9XgVcNFJGazJ" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "1",
          "seen" : false,
          "time" : 1550086655090,
          "type" : "text"
        },
        "-LYcMJqThca1CTPzEMxD" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "2",
          "seen" : false,
          "time" : 1550086655768,
          "type" : "text"
        },
        "-LYcMJy2Ti9E43eFr_Vf" : {
          "from" : "nC30bnOr0OGVpgJMhYmMTnstAj1",
          "message" : "3",
          "seen" : false,
          "time" : 1550086656257,
          "type" : "text"
        }
      }
    }
  }
}
}
}
}

```

2. Lampiran *coding start activity*

```

public class StartActivity extends AppCompatActivity {

    private Button mReqBtn;
    private Button mLogBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start);
    }
}

```

```

mReqBtn = (Button) findViewById(R.id.start_req_btn);
mLogBtn = (Button) findViewById( R.id.start_login_btn );

mReqBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent req_intent = new Intent(StartActivity.this, RegisterActivity.class);
        startActivity(req_intent);

    }
});

mLogBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent log_intent = new Intent( StartActivity.this, LoginActivity.class );
        startActivity( log_intent );
    }
});
}
}

```

3. Lampiran *coding login activity*

```

public class LoginActivity extends AppCompatActivity {

    private Toolbar mToolbar;
    private TextInputLayout mLoginEmail;
    private TextInputLayout mLoginPassword;

    private Button mLogin_btn;
    private ProgressDialog mLoginProgress;
    private FirebaseAuth mAuth;

    private DatabaseReference mUserDatabase;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_login );

        mAuth = FirebaseAuth.getInstance();

        mToolbar = (Toolbar) findViewById( R.id.login_toolbar );
        setSupportActionBar( mToolbar );
        getSupportActionBar().setDisplayHomeAsUpEnabled( true );
        getSupportActionBar().setTitle( "Login" );

        mLoginProgress = new ProgressDialog( this );

```

```

mUserDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" );

mLoginEmail = (TextInputLayout) findViewById( R.id.login_email );
mLoginPassword = (TextInputLayout) findViewById( R.id.login_password );
mLogin_btn = (Button) findViewById( R.id.login_btn );

mLogin_btn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String email = mLoginEmail.getText().toString();
        String password = mLoginPassword.getText().toString();

        if (!TextUtils.isEmpty( email ) || !TextUtils.isEmpty( password )){

            mLoginProgress.setTitle( "Logging In" );
            mLoginProgress.setMessage( "Please Wait While We Check Your Credentials." );
            mLoginProgress.setCanceledOnTouchOutside( false );
            mLoginProgress.show();

            loginUser(email,password);
        }
    }
});

private void loginUser(String email, String password) {

    mAuth.signInWithEmailAndPassword( email, password ).addOnCompleteListener( new
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            if (task.isSuccessful){

                mLoginProgress.dismiss();

                String current_user_id = mAuth.getCurrentUser().getUid();
                String deviceToken = FirebaseInstanceId.getInstance().getToken();

                mUserDatabase.child( current_user_id ).child( "device_token" ).setValue(
                deviceToken ).addOnSuccessListener( new OnSuccessListener <Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {

                        Intent mainIntent = new Intent( LoginActivity.this, MainActivity.class );
                        mainIntent.addFlags( Intent.FLAG_ACTIVITY_NEW_TASK |
                        Intent.FLAG_ACTIVITY_CLEAR_TASK );
                        startActivity( mainIntent );
                        finish();

                    }
                });
            }
        }
    });
}

```

```

        } else {
            mLoginProgress.hide();
            Toast.makeText(LoginActivity.this, "Cannot Sign In. Please Check Your Email or
Password.", Toast.LENGTH_LONG).show();
        }
    }
});
}
}

```

4. Lampiran *coding register activity*

```

public class RegisterActivity extends AppCompatActivity {

    private TextInputLayout mDisplayName;
    private TextInputLayout mEmail;
    private TextInputLayout mPassword;
    private Button mCreateBtn;

    private Toolbar mToolbar;

    private DatabaseReference mDatabase;

    private ProgressDialog mRegProgress;

    // Firebase Auth
    private FirebaseAuth mAuth;

    private DatabaseReference mUserDatabase;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        //Toolbar Set
        mToolbar =(Toolbar) findViewById(R.id.register_toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setTitle("Create Account");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        mRegProgress = new ProgressDialog(this );

        //Firebase Auth
        mAuth = FirebaseAuth.getInstance();

        mUserDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" );

        //Android Fields

```

```

mDisplayName = (TextInputLayout) findViewById(R.id.req_display_name);
mEmail = (TextInputLayout) findViewById(R.id.req_email);
mPassword = (TextInputLayout) findViewById(R.id.req_password);
mCreateBtn = (Button) findViewById(R.id.req_create_btn);

mCreateBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String display_name = mDisplayName.getText().toString();
        String email = mEmail.getText().toString();
        String password = mPassword.getText().toString();

        if (!TextUtils.isEmpty( display_name ) || !TextUtils.isEmpty( email ) ||
!TextUtils.isEmpty( password )){

            mRegProgress.setTitle( "Registering User" );
            mRegProgress.setMessage( "Please Wait While We Create Your Account!" );
            mRegProgress.setCanceledOnTouchOutside( false );
            mRegProgress.show();

            register_user(display_name, email, password);

        }

    }
});
}

private void register_user(final String display_name, String email, String password) {

    mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            if(task.isSuccessful()){

                FirebaseUser current_user = FirebaseAuth.getInstance().getCurrentUser();
                String uid = current_user.getUid();

                mDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
uid );

                String deviceToken = FirebaseInstanceId.getInstance().getToken();

                HashMap<String, String> userMap = new HashMap<>( );
                userMap.put( "name", display_name );
                userMap.put( "status", "Hi there, I'm using NgobrolYuk! Chat App" );
                userMap.put( "image", "default" );
                userMap.put( "thumb_image", "default" );
                userMap.put( "device_token", deviceToken );
            }
        }
    });
}

```

```

        mDatabase.setValue( userMap ).addOnCompleteListener( new
OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {

        if(task.isSuccessful()){

            mRegProgress.dismiss();

            String current_user_id = mAuth.getCurrentUser().getUid();
            String deviceToken = FirebaseInstanceId.getInstance().getToken();

            mUserDatabase.child( current_user_id ).child( "device_token" ).setValue(
deviceToken ).addOnSuccessListener( new OnSuccessListener <Void>() {
                @Override
                public void onSuccess(Void aVoid) {

                    Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);
                    mainIntent.addFlags( Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK );
                    startActivity(mainIntent);
                    finish();

                }
            } );

        }

    }

});

} else {

    mRegProgress.hide();
    Toast.makeText(RegisterActivity.this, "Cannot Sign Up. Please Check The Form and
Try Again.", Toast.LENGTH_LONG).show();

}

}

});

}

}

```

5. Lampiran *coding main activity*

```

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private Toolbar mToolbar;

```

```

private ViewPager mViewPager;
private SectionsPagerAdapter mSectionsPagerAdapter;

private DatabaseReference mUserRef;

private DatabaseReference mDatabase;

private TabLayout mTabLayout;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mDatabase = FirebaseDatabase.getInstance().getReference();

    mAuth = FirebaseAuth.getInstance();

    mToolbar = (Toolbar) findViewById(R.id.main_page_toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("NgobrolYuk!");

    if (mAuth.getCurrentUser() != null){

        mUserRef = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
mAuth.getCurrentUser().getUid() );

    }

    //Tabs
    mViewPager = (ViewPager) findViewById( R.id.main_tabPager );
    mSectionsPagerAdapter = new SectionsPagerAdapter( getSupportFragmentManager() );

    mViewPager.setAdapter( mSectionsPagerAdapter );

    mTabLayout =(TabLayout) findViewById( R.id.main_tabs );
    mTabLayout.setupWithViewPager( mViewPager );

}

@Override
public void onStart() {
    super.onStart();
    // Check if user is signed in (non-null) and update UI accordingly.
    FirebaseUser currentUser = mAuth.getCurrentUser();

    if(currentUser == null){

        sendToStart();

    } else {

        mUserRef.child( "online" ).setValue( "true" );

```

```

    }
}

@Override
protected void onStop() {
    super.onStop();

    FirebaseUser currentUser = mAuth.getCurrentUser();

    if(currentUser != null) {

        mUserRef.child( "online" ).setValue( ServerValue.TIMESTAMP );

    }
}

private void sendToStart() {

    Intent startIntent = new Intent(MainActivity.this, StartActivity.class);
    startActivity(startIntent);
    finish();

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu( menu );
    getMenuInflater().inflate( R.menu.main_menu, menu );

    return true;

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected( item );

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

        mUserRef.child("online").setValue(ServerValue.TIMESTAMP);

        FirebaseAuth.getInstance().signOut();
        sendToStart();

    }

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

        Intent settingsIntent = new Intent( MainActivity.this, SettingsActivity.class );
        startActivity( settingsIntent );

    }
}

```

```

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

            Intent settingsIntent = new Intent( MainActivity.this, UsersActivity.class );
            startActivity( settingsIntent );

        }

        return true;
    }
}

```

6. Lampiran *coding chats fragment*

```

public class ChatsFragment extends Fragment {

    private RecyclerView mConvList;

    private DatabaseReference mConvDatabase;
    private DatabaseReference mMessageDatabase;
    private DatabaseReference mUsersDatabase;

    private FirebaseAuth mAuth;

    private String mCurrent_user_id;

    private View mMainView;

    public ChatsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        mMainView = inflater.inflate(R.layout.fragment_chats, container, false);

        mConvList = (RecyclerView) mMainView.findViewById(R.id.conv_list);
        mAuth = FirebaseAuth.getInstance();

        mCurrent_user_id = mAuth.getCurrentUser().getUid();

        mConvDatabase =
        FirebaseDatabase.getInstance().getReference().child("Chat").child(mCurrent_user_id);

        mConvDatabase.keepSynced(true);
        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
        mMessageDatabase =
        FirebaseDatabase.getInstance().getReference().child("messages").child(mCurrent_user_id);
        mUsersDatabase.keepSynced(true);

        LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
        layoutManager.setReverseLayout(true);

```

```

linearLayoutManager.setStackFromEnd(true);

mConvList.setHasFixedSize(true);
mConvList.setLayoutManager(linearLayoutManager);

// Inflate the layout for this fragment
return mView;
}

@Override
public void onStart() {
    super.onStart();

    FirebaseRecyclerOptions<Conv> options = new FirebaseRecyclerOptions.Builder<Conv>()
        .setQuery( mConvDatabase, Conv.class )
        .build();

    FirebaseRecyclerAdapter<Conv, ConvViewHolder> firebaseConvAdapter = new
    FirebaseRecyclerAdapter<Conv, ConvViewHolder>(options) {
        @NonNull
        @Override
        public ConvViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
        viewType) {
            View view = LayoutInflater.from( parent.getContext() )
                .inflate( R.layout.users_single_layout, parent, false );

            return new ConvViewHolder( view );
        }

        @Override
        protected void onBindViewHolder(@NonNull final ConvViewHolder convViewHolder,
        int i, @NonNull final Conv conv) {

            final String list_user_id = getRef(i).getKey();

            Query lastMessageQuery = mMessageDatabase.child(list_user_id).limitToLast(1);

            lastMessageQuery.addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(DataSnapshot dataSnapshot, String s) {

                    String data = dataSnapshot.child("message").getValue().toString();
                    convViewHolder.setMessage(data, conv.isSeen());

                }

                @Override
                public void onChildChanged(DataSnapshot dataSnapshot, String s) {

                }

                @Override
                public void onChildRemoved(DataSnapshot dataSnapshot) {

```

```

    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

mUsersDatabase.child(list_user_id).addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        final String userName = dataSnapshot.child("name").getValue().toString();
        String userThumb = dataSnapshot.child("thumb_image").getValue().toString();

        if(dataSnapshot.hasChild("online")) {

            String userOnline = dataSnapshot.child("online").getValue().toString();
            convViewHolder.setUserOnline(userOnline);

        }

        convViewHolder.setName(userName);
        convViewHolder.setUserImage(userThumb, getContext());

        convViewHolder.mView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent chatIntent = new Intent(getContext(), ChatActivity.class);
                chatIntent.putExtra("user_id", list_user_id);
                chatIntent.putExtra("user_name", userName);
                startActivity(chatIntent);

            }
        });

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});
}
};
}
};

```

```

        mConvList.setAdapter(firebaseConvAdapter);
        firebaseConvAdapter.startListening();
    }

    public static class ConvViewHolder extends RecyclerView.ViewHolder {

        View mView;

        public ConvViewHolder(View itemView) {
            super(itemView);

            mView = itemView;
        }

        public void setMessage(String message, boolean isSeen){

            TextView userStatusView = (TextView) mView.findViewById(R.id.user_single_status);
            userStatusView.setText(message);

            if(!isSeen){
                userStatusView.setTypeface(userStatusView.getTypeface(), Typeface.BOLD);
            } else {
                userStatusView.setTypeface(userStatusView.getTypeface(), Typeface.NORMAL);
            }
        }

        public void setName(String name){

            TextView userNameView = (TextView) mView.findViewById(R.id.user_single_name);
            userNameView.setText(name);
        }

        public void setUserImage(String thumb_image, Context ctx){

            CircleImageView userImageView = (CircleImageView)
            mView.findViewById(R.id.user_single_image);

            Picasso.get().load(thumb_image).placeholder(R.drawable.default_avatar).into(userImageView);
        }

        public void setUserOnline(String online_status) {

            ImageView userOnlineView = (ImageView)
            mView.findViewById(R.id.user_single_online_icon);

            if(online_status.equals("true")){

                userOnlineView.setVisibility(View.VISIBLE);
            }
        }
    }
}

```



```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate( savedInstanceState );
    setContentView( R.layout.activity_chat );

    mChatToolBar = (ToolBar) findViewById( R.id.chat_app_bar );
    setSupportActionBar( mChatToolBar );

    ActionBar actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled( true );
    actionBar.setDisplayShowCustomEnabled( true );

    mRootRef = FirebaseDatabase.getInstance().getReference();
    mAuth = FirebaseAuth.getInstance();
    mCurrentUserId = mAuth.getCurrentUser().getUid();

    mChatUser = getIntent().getStringExtra( "user_id" );
    String userName = getIntent().getStringExtra( "user_name" );

    LayoutInflater inflater = (LayoutInflater) this.getSystemService(
Context.LAYOUT_INFLATER_SERVICE );
    View action_bar_view = inflater.inflate( R.layout.chat_custom_bar, null );

    actionBar.setCustomView( action_bar_view );

    // ----- Custom Action Bar Item -----

    mTitleView = (TextView) findViewById( R.id.custom_bar_title );
    mLastSeenView = (TextView) findViewById( R.id.custom_bar_seen );
    mProfileImage = (CircleImageView) findViewById( R.id.custom_bar_image );

    mChatAddBtn = (ImageButton) findViewById( R.id.chat_add_btn );
    mChatSendBtn = (ImageButton) findViewById( R.id.chat_send_btn );
    mChatMessageView = (EditText) findViewById( R.id.chat_message_view );

    mAdapter = new MessageAdapter( messagesList );
    mMessagesList = (RecyclerView) findViewById( R.id.messages_list );
    mRefreshLayout = (SwipeRefreshLayout) findViewById( R.id.message_swipe_layout );
    mLinearLayout = new LinearLayoutManager( this );

    mMessagesList.setHasFixedSize( true );
    mMessagesList.setLayoutManager( mLinearLayout );
    mMessagesList.setAdapter( mAdapter );

    loadMessages();

    // ----- Image Storage -----

    mImageStorage = FirebaseStorage.getInstance().getReference();

    mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true);

    mTitleView.setText( userName );

```

```

        mRootRef.child( "Users" ).child( mChatUser ).addValueEventListener( new
ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String online = dataSnapshot.child( "online" ).getValue().toString();
        String image = dataSnapshot.child( "image" ).getValue().toString();

        Picasso.get().load( image ).into( mProfileImage );

        if (online.equals( "true" )){

            mLastSeenView.setText( "Online" );

        } else {

            GetTimeAgo getTimeAgo = new GetTimeAgo();
            long lastTime = Long.parseLong( online );
            String lastSeenTime = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());
            mLastSeenView.setText( lastSeenTime );

        }

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }

});

        mRootRef.child( "Chat" ).child( mCurrentUserId ).addValueEventListener( new
ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        if (!dataSnapshot.hasChild( mChatUser )){

            Map chatAddMap = new HashMap( );
            chatAddMap.put( "seen", false );
            chatAddMap.put( "timestamp", ServerValue.TIMESTAMP );

            Map chatUserMap = new HashMap( );
            chatUserMap.put( "Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap );
            chatUserMap.put( "Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap );

            mRootRef.updateChildren( chatUserMap, new
DatabaseReference.CompletionListener() {
                @Override
                public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                    if (databaseError != null){
                        Log.d( "CHAT_LOG", databaseError.getMessage().toString() );
                    }

                }

            }

        }

    }

});

```

```

        }
    });

}

}

@Override
public void onCancelled(DatabaseError databaseError) {

}
});

mChatSendBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        sendMessage();

    }
});

mChatAddBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent galleryIntent = new Intent( );
        galleryIntent.setType( "image/*" );
        galleryIntent.setAction( Intent.ACTION_GET_CONTENT );

        startActivityForResult( Intent.createChooser( galleryIntent, "SELECT IMAGE" ),
GALLERY_PICK );

    }
});

mRefreshLayout.setOnRefreshListener( new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {

        mCurrentPage++;

        itemPos = 0;

        loadMoreMessages();

    }
});

}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult( requestCode, resultCode, data );
}

```

```

if (requestCode == GALLERY_PICK && resultCode == RESULT_OK){

    Uri imageUri = data.getData();

    final String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
    final String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

    DatabaseReference user_message_push = mRootRef.child( "messages" ).child(
mCurrentUserId ).child( mChatUser ).push();

    final String push_id = user_message_push.getKey();

    final StorageReference filepath = mImageStorage.child( "message_images" ).child(
push_id + ".jpg" );

    filepath.putFile( imageUri ).addOnCompleteListener( new OnCompleteListener
<UploadTask.TaskSnapshot>() {
        @Override
        public void onComplete(final Task<UploadTask.TaskSnapshot> task) {

            if (task.isSuccessful()){

                mImageStorage.child( "message_images" ).child( push_id + ".jpg"
).getDownloadUrl().addOnSuccessListener( new OnSuccessListener <Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {

                        final String download_url = uri.toString();

                        Map messageMap = new HashMap( );
                        messageMap.put( "message", download_url );
                        messageMap.put( "seen", false );
                        messageMap.put( "type", "image" );
                        messageMap.put( "time", ServerValue.TIMESTAMP );
                        messageMap.put( "from", mCurrentUserId );

                        Map messageUserMap = new HashMap( );
                        messageUserMap.put( current_user_ref + "/" + push_id, messageMap );
                        messageUserMap.put( chat_user_ref + "/" + push_id, messageMap );

                        mChatMessageView.setText( "" );

                        mRootRef.updateChildren( messageUserMap, new
DatabaseReference.CompletionListener() {
                            @Override
                            public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                                if (databaseError != null){

                                    Log.d( "CHAT_LOG", databaseError.getMessage().toString() );

                                }

                            }

                        }

                    }

                }

            }

        }

    }

}

```



```

@Override
public void onChildChanged(DataSnapshot dataSnapshot, String s) {

}

@Override
public void onChildRemoved(DataSnapshot dataSnapshot) {

}

@Override
public void onChildMoved(DataSnapshot dataSnapshot, String s) {

}

@Override
public void onCancelled(DatabaseError databaseError) {

}
});
}

private void loadMessages() {

    DatabaseReference messageRef = mRootRef.child( "messages" ).child( mCurrentUserId
).child( mChatUser );

    Query messageQuery = messageRef.limitToLast( mCurrentPage *
TOTAL_ITEMS_TO_LOAD );

    messageQuery.addChildEventListener( new ChildEventListener() {
@Override
public void onChildAdded(DataSnapshot dataSnapshot, String s) {

        Messages message = dataSnapshot.getValue(Messages.class);

        itemPos++;

        if (itemPos == 1){

            String messageKey = dataSnapshot.getKey();

            mLastKey = messageKey;
            mPrevKey = messageKey;

        }

        messagesList.add( message );
        mAdapter.notifyDataSetChanged();

        mMessagesList.scrollToPosition( messagesList.size() - 1 );

        mRefreshLayout.setRefreshing( false );

```

```

    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {

    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});
}

private void sendMessage() {

    String message = mChatMessageView.getText().toString();

    if (!TextUtils.isEmpty( message )){

        String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
        String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

        DatabaseReference user_message_push = mRootRef.child( "messages" ).child(
mCurrentUserId ).child( mChatUser ).push();

        String push_id = user_message_push.getKey();

        Map messageMap = new HashMap( );
        messageMap.put( "message", message );
        messageMap.put( "seen", false );
        messageMap.put( "type", "text" );
        messageMap.put( "time", ServerValue.TIMESTAMP );
        messageMap.put( "from", mCurrentUserId );

        Map messageUserMap = new HashMap( );
        messageUserMap.put( current_user_ref + "/" + push_id, messageMap );
        messageUserMap.put( chat_user_ref + "/" + push_id, messageMap );

        mChatMessageView.setText( "" );

mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true);

```

```
mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("timestamp").setValue(
    serverValue.TIMESTAMP);
```

```
mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("seen").setValue(false);
```

```
mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("timestamp").setValue(
    serverValue.TIMESTAMP);
```

```
        mRootRef.updateChildren( messageUserMap, new
        DatabaseReference.CompletionListener() {
            @Override
            public void onComplete(DatabaseError databaseError, DatabaseReference
            databaseReference) {

                if (databaseError != null){
                    Log.d( "CHAT_LOG", databaseError.getMessage().toString() );
                }

            }
        });
    }
}
```

8. Lampiran *coding friends fragment*

```
public class FriendsFragment extends Fragment {

    private RecyclerView mFriendsList;

    private DatabaseReference mFriendsDatabase;
    private DatabaseReference mUsersDatabase;

    private FirebaseAuth mAuth;

    private String mCurrent_user_id;

    private View mMainView;

    public FriendsFragment() {
        // Required empty public constructor
    }

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

        mMainView = inflater.inflate( R.layout.fragment_friends, container, false );
```

```

mFriendsList = (RecyclerView) mMainView.findViewById( R.id.friends_list );

 mAuth = FirebaseAuth.getInstance();
 mCurrent_user_id = mAuth.getCurrentUser().getUid();

 mFriendsDatabase = FirebaseDatabase.getInstance().getReference().child( "Friends" ).child(
 mCurrent_user_id );
 mFriendsDatabase.keepSynced( true );
 mUsersDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" );
 mUsersDatabase.keepSynced( true );

 mFriendsList.setHasFixedSize( true );
 mFriendsList.setLayoutManager( new LinearLayoutManager( getContext() ) );

 // Inflate the layout for this fragment
 return mMainView;
 }

 @Override
 public void onStart(){
 super.onStart();

 FirebaseRecyclerOptions<Friends> options = new
 FirebaseRecyclerOptions.Builder<Friends>()
 .setQuery( mFriendsDatabase, Friends.class )
 .build();

 FirebaseRecyclerAdapter<Friends, FriendsViewHolder> friendsRecyclerViewAdapter = new
 FirebaseRecyclerAdapter <Friends, FriendsViewHolder>(options) {

 @NonNull
 @Override
 public FriendsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
 viewType) {

 View view = LayoutInflater.from( parent.getContext() )
 .inflate( R.layout.users_single_layout, parent, false );

 return new FriendsViewHolder( view );

 }

 @Override
 protected void onBindViewHolder(@NonNull final FriendsViewHolder
 friendsViewHolder, int i, @NonNull Friends friends) {

 friendsViewHolder.setDate( friends.getDate() );

 final String list_user_id = getRef( i ).getKey();

 mUsersDatabase.child( list_user_id ).addValueEventListener( new ValueEventListener()
 {

 @Override
 public void onDataChange(DataSnapshot dataSnapshot) {

```

```

final String userName = dataSnapshot.child( "name" ).getValue().toString();
String userThumb = dataSnapshot.child( "thumb_image" ).getValue().toString();

if (dataSnapshot.hasChild( "online" )){

    String userOnline = dataSnapshot.child( "online" ).getValue().toString();
    friendsViewHolder.setUserOnline( userOnline );

}

friendsViewHolder.setName( userName );
friendsViewHolder.setUserImage( userThumb, getContext() );

friendsViewHolder.mView.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick( View v ) {

        CharSequence options[] = new CharSequence[]{ "Open Profile", "Send
Message" };

        AlertDialog.Builder builder = new AlertDialog.Builder( getContext() );

        builder.setTitle( "Select Options" );
        builder.setItems( options, new DialogInterface.OnClickListener() {
            @Override
            public void onClick( DialogInterface dialogInterface, int i ) {

                //Click Event for Each Item.
                if ( i == 0 ){

                    Intent profileIntent = new Intent( getContext(), ProfileeActivity.class );
                    profileIntent.putExtra( "user_id", list_user_id );
                    startActivity( profileIntent );

                }

                if ( i == 1 ){

                    Intent chatIntent = new Intent( getContext(), ChatActivity.class );
                    chatIntent.putExtra( "user_id", list_user_id );
                    chatIntent.putExtra( "user_name", userName );
                    startActivity( chatIntent );

                }

            }

        } );

        builder.show();

    }

} );
}

```

```

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });

}

};

mFriendsList.setAdapter( friendsRecyclerViewAdapter );
friendsRecyclerViewAdapter.startListening();

}

public static class FriendsViewHolder extends RecyclerView.ViewHolder {

    View mView;

    public FriendsViewHolder(View itemView) {
        super(itemView);

        mView = itemView;
    }

    public void setDate(String date){

        TextView userNameView = (TextView) mView.findViewById( R.id.user_single_status );
        userNameView.setText( date );

    }

    public void setName(String name){

        TextView userNameView = (TextView) mView.findViewById( R.id.user_single_name );
        userNameView.setText( name );

    }

    public void setUserImage(String thumb_image, Context ctx){

        CircleImageView userImageView = (CircleImageView) mView.findViewById(
R.id.user_single_image );

        Picasso.get().load( thumb_image ).placeholder( R.drawable.default_avatar ).into(
userImageView );

    }

    public void setUserOnline (String online_status){

        ImageView userOnlineView = (ImageView) mView.findViewById(
R.id.user_single_online_icon );
        if ( online_status.equals( "true" ) ){

```



```

holder.userName.setText( requestUserName );
holder.userStatus.setText( "I want to be your friend, let's chat with
NgobrolYuk!" );

holder.itemView.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        CharSequence options[] = new CharSequence[]{

            "Accept Friend Request",
            "Decline Friend Request"

        };

        AlertDialog.Builder builder = new AlertDialog.Builder(
getContext() );

        builder.setTitle( requestUserName + " Request Options" );

        builder.setItems( options, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

                if (i == 0){

                    final String currentDate =
DateFormat.getDateTimeInstance().format( new Date( ) );

                    FriendsDatabase.child( currentUserId ).child( list_user_id
).child( "date" )
                        .setValue( currentDate ).addOnCompleteListener( new
OnCompleteListener <Void>() {
                            @Override
                            public void onComplete(@NonNull Task<Void> task) {

                                if (task.isSuccessful()){

                                    FriendsDatabase.child( list_user_id ).child(
currentUserId ).child( "date" )
                                        .setValue( currentDate
).addOnCompleteListener( new OnCompleteListener <Void>() {
                                            @Override
                                            public void onComplete(@NonNull Task<Void>
task) {

                                                if (task.isSuccessful()){

                                                    FriendRequestsDatabase.child( currentUserId
).child( list_user_id ).removeValue()
                                                        .addOnCompleteListener( new
OnCompleteListener <Void>() {
                                                            @Override
                                                            public void onComplete(@NonNull
Task <Void> task) {

```



```

<Void> task) {

                                if (task.isSuccessful){

                                    Toast.makeText( getContext(),
"Friend Request Decline Successfully", Toast.LENGTH_SHORT ).show();

                                    }

                                }

                            });

                    }

                }

            });

        builder.show();

    }

});

builder.show();

}

});

}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

//Send Request Friend
else if (type.equals( "sent" )){

    Button request_sent_btn = holder.itemView.findViewById( R.id.acc_req_btn

);

    request_sent_btn.setText( "Request Sent" );

    holder.itemView.findViewById( R.id.decline_req_btn ).setVisibility(
View.INVISIBLE );

    UsersRef.child( list_user_id ).addValueEventListener( new
ValueEventListener() {

        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

            if (dataSnapshot.hasChild( "image" )){

                final String requestProfileImage = dataSnapshot.child( "image"

```



```

        Toast.makeText( getContext(),
        "Friend Request Cancel Successfully", Toast.LENGTH_SHORT ).show();
    }
}
});
}
}
});
}
}
builder.show();
}
});
}
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
}
});
}
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
}
});
}
}
@Override
@Override
public RequestsViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int
viewType) {
    View view = LayoutInflater.from( viewGroup.getContext() ).inflate(
R.layout.users_req_layout, viewGroup, false );
    RequestsViewHolder holder = new RequestsViewHolder( view );
    return holder;
}
};

```

```

        mRequestsList.setAdapter( adapter );
        adapter.startListening();
    }

    public static class RequestsViewHolder extends RecyclerView.ViewHolder{

        TextView userName, userStatus;
        CircleImageView profileImage;
        Button AcceptButton, DeclineButton;

        public RequestsViewHolder(@NonNull View itemView) {
            super( itemView );

            userName = itemView.findViewById( R.id.user_req_name );
            userStatus = itemView.findViewById( R.id.user_req_status );
            profileImage = itemView.findViewById( R.id.user_req_image );
            AcceptButton = itemView.findViewById( R.id.acc_req_btn );
            DeclineButton = itemView.findViewById( R.id.decline_req_btn );
        }
    }
}

```

10. Lampiran *coding find friends*

```

public class UsersActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    private EditText mSearchField;
    private ImageButton mSearchBtn;

    private RecyclerView mUsersList;

    private DatabaseReference mUsersDatabase;

    private LinearLayoutManager mLayoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_users );

        mToolbar = (Toolbar) findViewById( R.id.users_appBar );
        setSupportActionBar( mToolbar );

        getSupportActionBar().setTitle( "Find Friends" );
        getSupportActionBar().setDisplayHomeAsUpEnabled( true );

        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" );
    }
}

```

```

mLayoutManager = new LinearLayoutManager(this);

mSearchField = (EditText) findViewById( R.id.search_field );
mSearchBtn = (ImageButton) findViewById( R.id.search_btn );

mUsersList = (RecyclerView) findViewById( R.id.users_list );
mUsersList.setHasFixedSize( true );
mUsersList.setLayoutManager( mLayoutManager );

mSearchBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String searchText = mSearchField.getText().toString();
        firebaseUserSearch(searchText);

    }
});

}

private void firebaseUserSearch(String searchText) {

    FirebaseRecyclerOptions<Users> options = new FirebaseRecyclerOptions.Builder<Users>()
        .setQuery( mUsersDatabase.orderByChild( "name" ).startAt( searchText ).endAt(
searchText + "\uf8ff" ),Users.class )
        .build();

    FirebaseRecyclerAdapter<Users, UsersViewHolder> firebaseRecyclerAdapter = new
FirebaseRecyclerAdapter <Users, UsersViewHolder>(options) {

        @NonNull
        @Override
        public UsersViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
            View view = LayoutInflater.from( parent.getContext() )
                .inflate( R.layout.users_single_layout, parent, false );

            return new UsersViewHolder( view );
        }

        @Override
        protected void onBindViewHolder(@NonNull UsersViewHolder usersViewHolder, int i,
@NonNull Users users) {

            usersViewHolder.setName(users.getName());
            usersViewHolder.setUserStatus(users.getStatus());
            usersViewHolder.setUserImage(users.getThumb_image(), getApplicationContext());

            final String user_id = getRef( i ).getKey();

            usersViewHolder.mView.setOnClickListener( new View.OnClickListener() {
                @Override
                public void onClick(View v) {

```

```

        Intent profileIntent = new Intent( UsersActivity.this, ProfileeActivity.class );
        profileIntent.putExtra( "user_id", user_id );
        startActivity( profileIntent );

    }
    });

}
};

mUsersList.setAdapter(firebaseRecyclerAdapter);
firebaseRecyclerAdapter.startListening();

}

@Override
protected void onStart() {
    super.onStart();

    FirebaseRecyclerOptions<Users> options = new FirebaseRecyclerOptions.Builder<Users>()
        .setQuery( mUsersDatabase,Users.class )
        .build();

    FirebaseRecyclerAdapter <Users, UsersViewHolder> firebaseRecyclerAdapter = new
    FirebaseRecyclerAdapter <Users, UsersViewHolder>( options ) {
        @NonNull
        @Override
        public UsersViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
        viewType) {

            View view = LayoutInflater.from( parent.getContext() )
                .inflate( R.layout.users_single_layout, parent, false );

            return new UsersViewHolder( view );
        }
    }

    @Override
    protected void onBindViewHolder(@NonNull UsersViewHolder usersViewHolder, int i,
    @NonNull Users users) {

        usersViewHolder.setName(users.getName());
        usersViewHolder.setUserStatus(users.getStatus());
        usersViewHolder.setUserImage(users.getThumb_image(), getApplicationContext());

        final String user_id = getRef( i ).getKey();

        usersViewHolder.mView.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent profileIntent = new Intent( UsersActivity.this, ProfileeActivity.class );
                profileIntent.putExtra( "user_id", user_id );
                startActivity( profileIntent );
            }
        }
    }
}

```

```

        }
    });

    }
};

mUsersList.setAdapter(firebaseRecyclerAdapter);
firebaseRecyclerAdapter.startListening();

}

public static class UsersViewHolder extends RecyclerView.ViewHolder {

    View mView;

    public UsersViewHolder( View itemView) {
        super( itemView );

        mView = itemView;
    }

    public void setName(String name) {

        TextView userNameView = (TextView) mView.findViewById( R.id.user_single_name );
        userNameView.setText( name );
    }

    public void setUserStatus(String status){

        TextView userStatusView = (TextView) mView.findViewById( R.id.user_single_status );
        userStatusView.setText( status );
    }

    public void setUserImage(String thumb_image, Context ctx){

        CircleImageView userImageView = (CircleImageView) mView.findViewById(
R.id.user_single_image );

        Picasso.get().load( thumb_image ).placeholder( R.drawable.default_avatar ).into(
userImageView );
    }
}
}
}

```

11. Lampiran *coding profile all user*

```

public class ProfileeActivity extends AppCompatActivity {

    private ImageView mProfileImage;
    private TextView mProfileName, mProfileStatus, mProfileFriendsCount;
    private Button mProfileSendReqBtn, mDeclineBtn;

    private DatabaseReference mUsersDatabase;

    private ProgressDialog mProgressDialog;

    private DatabaseReference mFriendReqDatabase;
    private DatabaseReference mFriendDatabase;
    private DatabaseReference mNotificationDatabase;

    private DatabaseReference mRootRef;

    private FirebaseUser mCurrent_user;

    private String mCurrent_state;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_profilee );

        final String user_id = getIntent().getStringExtra( "user_id" );

        mRootRef = FirebaseDatabase.getInstance().getReference();

        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
user_id );
        mFriendReqDatabase = FirebaseDatabase.getInstance().getReference().child( "Friend_req" );
        mFriendDatabase = FirebaseDatabase.getInstance().getReference().child( "Friends" );
        mNotificationDatabase = FirebaseDatabase.getInstance().getReference().child(
"Notifications" );
        mCurrent_user = FirebaseAuth.getInstance().getCurrentUser();

        mProfileImage = (ImageView) findViewById( R.id.profile_image );
        mProfileName = (TextView) findViewById( R.id.profile_displayName );
        mProfileStatus = (TextView) findViewById( R.id.profile_status );
        mProfileFriendsCount = (TextView) findViewById( R.id.profile_totalFriends );
        mProfileSendReqBtn = (Button) findViewById( R.id.profile_send_req_btn );
        mDeclineBtn = (Button) findViewById( R.id.profile_decline_btn );

        mCurrent_state = "not_friends";

        mDeclineBtn.setVisibility( View.INVISIBLE );
        mDeclineBtn.setEnabled( false );

        mProgressDialog = new ProgressDialog( this );
        mProgressDialog.setTitle( "Loading User Data" );
        mProgressDialog.setMessage( "Please Wait While We Load The User Data" );
        mProgressDialog.setCanceledOnTouchOutside( false );
        mProgressDialog.show();
    }
}

```

```

mUsersDatabase.addValueEventListener( new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String display_name = dataSnapshot.child( "name" ).getValue().toString();
        String status = dataSnapshot.child( "status" ).getValue().toString();
        String image = dataSnapshot.child( "image" ).getValue().toString();

        mProfileName.setText( display_name );
        mProfileStatus.setText( status );

        Picasso.get().load( image ).placeholder( R.drawable.default_avatar ).into(
mProfileImage );

        if(mCurrent_user.getUid().equals(user_id)){

            mDeclineBtn.setEnabled(false);
            mDeclineBtn.setVisibility(View.INVISIBLE);

            mProfileSendReqBtn.setEnabled(false);
            mProfileSendReqBtn.setVisibility(View.INVISIBLE);

        }

        // ----- FRIENDS LIST / REQUEST FEATURE -----

        mFriendReqDatabase.child( mCurrent_user.getUid() ).addListenerForSingleValueEvent(
new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                if (dataSnapshot.hasChild( user_id )){

                    String req_type = dataSnapshot.child( user_id ).child( "request_type"
).getValue().toString();

                    if (req_type.equals( "received" )){

                        mCurrent_state = "req_received";
                        mProfileSendReqBtn.setText( "Accept Friend Request" );

                        mDeclineBtn.setVisibility( View.VISIBLE );
                        mDeclineBtn.setEnabled( true );

                    } else if (req_type.equals( "sent" )){

                        mCurrent_state = "req_sent";
                        mProfileSendReqBtn.setText( "Cancel Friend Request" );

                        mDeclineBtn.setVisibility( View.INVISIBLE );
                        mDeclineBtn.setEnabled( false );

                    }

                }

            }

        }

    }

}

```

```

    }

    mProgressDialog.dismiss();

} else {

    mFriendDatabase.child( mCurrent_user.getUid()
).addListenerForSingleValueEvent( new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.hasChild( user_id )){

            mCurrent_state = "friends";
            mProfileSendReqBtn.setText( "Unfriend this Person" );

            mDeclineBtn.setVisibility(View.INVISIBLE);
            mDeclineBtn.setEnabled(false);

        }

        mProgressDialog.dismiss();

    }

});

}

}

@Override
public void onCancelled(DatabaseError databaseError) {

    mProgressDialog.dismiss();

}

});

}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});

}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});

}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});

}

mProfileSendReqBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

mProfileSendReqBtn.setEnabled( false );

// ----- NOT FRIENDS STATE -----

if (mCurrent_state.equals( "not_friends" )){

    DatabaseReference newNotificationref = mRootRef.child( "Notifications" ).child(
user_id ).push();
    String newNotificationId = newNotificationref.getKey();

    HashMap<String, String> notificationData = new HashMap <>( );
    notificationData.put( "from", mCurrent_user.getUid() );
    notificationData.put( "type", "request" );

    Map requestMap = new HashMap( );
    requestMap.put( "Friend_req/" + mCurrent_user.getUid() + "/" + user_id +
"/request_type", "sent" );
    requestMap.put( "Friend_req/" + user_id + "/" + mCurrent_user.getUid() +
"/request_type", "received" );
    requestMap.put( "Notifications/" + user_id + "/" + newNotificationId,
notificationData );

    mRootRef.updateChildren( requestMap, new
DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

            if (databaseError != null){

                Toast.makeText( ProfileeActivity.this, "There Was Some Error in Sending
Request", Toast.LENGTH_SHORT ).show();

            } else {

                mCurrent_state = "req_sent";
                mProfileSendReqBtn.setText( "Cancel Friend Request" );

            }

            mProfileSendReqBtn.setEnabled( true );

        }
    });

}

// ----- CANCEL REQUEST STATE -----

if (mCurrent_state.equals( "req_sent" )){

    mFriendReqDatabase.child( mCurrent_user.getUid() ).child( user_id
).removeValue().addOnSuccessListener( new OnSuccessListener <Void>() {
        @Override
        public void onSuccess(Void aVoid) {

```

```

        mFriendReqDatabase.child( user_id ).child( mCurrent_user.getUid()
    ).removeValue().addOnSuccessListener( new OnSuccessListener <Void>() {
        @Override
        public void onSuccess(Void aVoid) {

            mProfileSendReqBtn.setEnabled( true );
            mCurrent_state = "not_friends";
            mProfileSendReqBtn.setText( "Send Friend Request" );

            mDeclineBtn.setVisibility( View.INVISIBLE );
            mDeclineBtn.setEnabled( false );

        }
    } );

}

// ----- REQ RECEIVED STATE -----

if (mCurrent_state.equals( "req_received" )){

    final String currentDate = DateFormat.getDateTimeInstance().format( new Date( ) );

    Map friendsMap = new HashMap( );
    friendsMap.put( "Friends/" + mCurrent_user.getUid() + "/" + user_id + "/date",
currentDate );
    friendsMap.put( "Friends/" + user_id + "/" + mCurrent_user.getUid() + "/date",
currentDate );

    friendsMap.put( "Friend_req/" + mCurrent_user.getUid() + "/" + user_id, null );
    friendsMap.put( "Friend_req/" + user_id + "/" + mCurrent_user.getUid(), null );

    mRootRef.updateChildren( friendsMap, new
DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

            if (databaseError == null){

                mProfileSendReqBtn.setEnabled( true );
                mCurrent_state = "friends";
                mProfileSendReqBtn.setText( "Unfriend this Person" );

                mDeclineBtn.setVisibility( View.INVISIBLE );
                mDeclineBtn.setEnabled( false );

            } else {

                String error = databaseError.getMessage();
                Toast.makeText( ProfileeActivity.this, error, Toast.LENGTH_SHORT

```


12. Lampiran *coding settings activity*

```

public class SettingsActivity extends AppCompatActivity {

    private DatabaseReference mUserDatabase;
    private FirebaseUser mCurrentUser;

    // Android Layout
    private CircleImageView mDisplayImage;
    private TextView mName;
    private TextView mStatus;

    private Button mStatusBtn;
    private Button mImageBtn;

    private static final int GALLERY_PICK = 1;

    //Storage Firebase
    private StorageReference mImageStorage;
    private ProgressDialog mProgressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_settings );

        mDisplayImage = (CircleImageView) findViewById( R.id.settings_image );
        mName = (TextView) findViewById( R.id.settings_display_name );
        mStatus = (TextView ) findViewById( R.id.settings_status );

        mStatusBtn = (Button) findViewById( R.id.settings_status_btn );
        mImageBtn = (Button) findViewById( R.id.settings_image_btn );

        mImageStorage = FirebaseStorage.getInstance().getReference();

        mCurrentUser = FirebaseAuth.getInstance().getCurrentUser();

        String current_uid = mCurrentUser.getId();

        mUserDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
current_uid );
        mUserDatabase.keepSynced( true );

        mUserDatabase.addValueEventListener( new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

                String name = dataSnapshot.child( "name" ).getValue().toString();
                final String image = dataSnapshot.child( "image" ).getValue().toString();
                String status = dataSnapshot.child( "status" ).getValue().toString();
                String thumb_image = dataSnapshot.child( "thumb_image" ).getValue().toString();

                mName.setText( name );
                mStatus.setText( status );

                if (!image.equals( "default" )) {

```

```

        Picasso.get().load( image ).networkPolicy( NetworkPolicy.OFFLINE ).placeholder(
R.drawable.default_avatar ).into( mDisplayImage, new Callback() {
    @Override
    public void onSuccess() {

    }

    @Override
    public void onError(Exception e) {

        Picasso.get().load( image ).placeholder( R.drawable.default_avatar ).into(
mDisplayImage);

    }
} );

}

}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}
});

mStatusBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String status_value = mStatus.getText().toString();

        Intent status_intent = new Intent( SettingsActivity.this, StatusActivity.class );
        status_intent.putExtra( "status_value", status_value );
        startActivity( status_intent );

    }
});

mImageBtn.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent galleryIntent = new Intent( );
        galleryIntent.setType( "image/*" );
        galleryIntent.setAction( Intent.ACTION_GET_CONTENT );

        startActivityForResult( Intent.createChooser( galleryIntent, "SELECT IMAGE" ),
GALLERY_PICK );

    }
});

```

```

}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult( requestCode, resultCode, data );

    if(requestCode == GALLERY_PICK && resultCode == RESULT_OK){

        Uri imageUri = data.getData();

        CropImage.activity(imageUri)
            .setAspectRatio( 1, 1 )
            .setMinCropWindowSize( 1000, 500 )
            .start(this);

    }

    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {

        CropImage.ActivityResult result = CropImage.getActivityResult(data);

        if (resultCode == RESULT_OK) {

            mProgressDialog = new ProgressDialog( SettingsActivity.this );
            mProgressDialog.setTitle( "Uploading Image..." );
            mProgressDialog.setMessage( "Please Wait While We Upload and Process The Image."
);

            mProgressDialog.setCanceledOnTouchOutside( false );
            mProgressDialog.show();

            Uri resultUri = result.getUri();

            final File thumb_filePath = new File( resultUri.getPath() );

            //Getting the Current UID of the User and storing it in a String.
            final String currentUserUid = mCurrentUser.getUid();

            try {
                Bitmap thumb_bitmap = new Compressor( this )
                    .setMaxWidth( 200 )
                    .setMaxHeight( 200 )
                    .setQuality( 75 )
                    .compressToBitmap( thumb_filePath );

                ByteArrayOutputStream baos = new ByteArrayOutputStream();
                thumb_bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
                final byte[] thumb_byte = baos.toByteArray();

                //Saving the image in the Firebase Storage and naming the child with the UID.
                final StorageReference filepath =
mImageStorage.child("profile_images").child(currentUserUid + ".jpg");

```



```

        }
    }
});

} else {
    Toast.makeText(SettingsActivity.this, "Error in Uploading
Thumbnail", Toast.LENGTH_LONG).show();

    mProgressDialog.dismiss();
}

}
});

}
});

}
});

} else {
    Toast.makeText(SettingsActivity.this, "Error in Uploading",
Toast.LENGTH_LONG).show();

    mProgressDialog.dismiss();

}
}
});
}

} catch (IOException e) {
    e.printStackTrace();
}

//If the task is not successful then we will display an Error Message.
} else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE)
{
    Exception error = result.getError();

}
}

}

}

public static String random() {
    Random generator = new Random();
    StringBuilder randomStringBuilder = new StringBuilder();

```

```

    int randomLength = generator.nextInt(15);
    char tempChar;
    for (int i = 0; i < randomLength; i++){
        tempChar = (char) (generator.nextInt(96) + 32);
        randomStringBuilder.append(tempChar);
    }
    return randomStringBuilder.toString();
}
}

```

13. Lampiran *coding status activity*

```

public class StatusActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    private TextInputLayout mStatus;
    private Button mSavebtn;

    //Firebase
    private DatabaseReference mStatusDatabase;
    private FirebaseUser mCurrentUser;

    //Progress
    private ProgressDialog mProgress;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_status );

        //Firebase
        mCurrentUser = FirebaseAuth.getInstance().getCurrentUser();
        String current_uid = mCurrentUser.getId();

        mStatusDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
current_uid );

        mToolbar = (Toolbar) findViewById( R.id.status_appBar );
        setSupportActionBar( mToolbar );
        getSupportActionBar().setTitle( "Account Status" );
        getSupportActionBar().setDisplayHomeAsUpEnabled( true );

        String status_value = getIntent().getStringExtra( "status_value" );

        mStatus = (TextInputLayout) findViewById( R.id.status_input );
        mSavebtn = (Button) findViewById( R.id.status_save_btn );

        mStatus.getEditText().setText( status_value );

        mSavebtn.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```



```
mAuth = FirebaseAuth.getInstance();

if (mAuth.getCurrentUser() != null){

    mUserDatabase = FirebaseDatabase.getInstance().getReference().child( "Users" ).child(
mAuth.getCurrentUser().getUid() );

    mUserDatabase.addValueEventListener( new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {

            if (dataSnapshot !=null){

                mUserDatabase.child( "online" ).onDisconnect().setValue(
ServerValue.TIMESTAMP );

            }

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }

    });
}
}
}
```