

4. IMPLEMENTASI SISTEM

Pada bab ini dibahas implementasi dari sistem yang telah dijelaskan pada bab sebelumnya. Implementasi sistem ini terbagi dalam 4 bagian, yaitu implementasi driver class, implementasi mapper class, implementasi combiner class, dan implementasi reducer class.

4.1 Instalasi dan Konfigurasi Owncloud

Sebelum melakukan instalasi aplikasi Owncloud, maka dilakukan pengecekan terhadap paket – paket yang terdapat di dalam Sistem Operasi Ubuntu. Langkah – langkah yang dilakukan adalah sebagai berikut:

1. Melakukan *Login* ke dalam sistem operasi (Ubuntu Desktop 14.04) yang digunakan oleh server.
2. Membuka terminal dan melakukan pengecekan paket – paket yang terdapat di dalam owncloud dengan perintah “`sudo apt-get update`”. Perintah tersebut digunakan untuk menyatakan apabila paket – paket yang terdapat di dalam Ubuntu adalah yang terbaru.
3. Menginstall Server Apache 2 dengan perintah “`sudo apt-get install apache2`”.
4. Melakukan instalasi mysql dengan “`sudo apt-get install mysql`”. Setelah menjalankan command tersebut, maka user akan diminta password dari user “root” pada mysql sebagai administrator dari database.
5. Melakukan instalasi phpmyadmin sebagai GUI untuk konfigurasi dari mysql dengan command “`sudo apt-get install phpmyadmin`”. Setelah melakukan instalasi dari phpmyadmin, user akan diminta password dari user “root” mysql yang telah diminta pada langkah sebelumnya, dan juga web server yang digunakan oleh server, pada aplikasi ini menggunakan apache2.
6. Setelah selesai menginstall phpmyadmin, maka dilakukan penggantian terhadap jumlah maksimal file yang dapat diunggah oleh php, dengan

menggunakan perintah “sudo nano /etc/php5/apache2/php.ini”, lalu mengganti line “upload_max_filesize = 2M” dengan “upload_max_filesize = 700M”, dan line “post_max_size = 700M” dengan “post_max_size = 700M”. Ukuran file *default* yang dapat diunggah oleh sebuah server adalah 2 MB, namun dengan mengganti command ini, maka jumlah file yang diupload akan disesuaikan dengan *setting*. Di dalam aplikasi ini, ukuran maksimal yang diunggah adalah 700 MB.

7. Memberikan akses kepada user selain root pada server agar dapat mengganti dan memodifikasi file yang terdapat di dalam server dengan memberi perintah “sudo nano etc/apache2/httpd.conf” dan mengganti line “require all deny” pada bagian lokasi dari file server, yaitu pada direktori /var/www/, menjadi “require all granted”.
8. Menggunakan command “sudo service apache2 restart” untuk meng-apply perubahan yang telah dilakukan di dalam server.
9. Melakukan Instalasi JDK pada server, dengan menambah terlebih dahulu *repository* dari JDK dengan menggunakan command line “sudo apt-add-repository ppa:webupd8team/java”, setelah menambahkan command line tersebut, maka user harus menggunakan perintah “sudo apt-get update” agar server mendapatkan info dari paket yang terdapat pada repository. Setelah selesai, maka user akan mengeksekusi perintah “sudo apt-get install oracle-java8-installer” dan menunggu proses instalasi sampai selesai.
10. Agar proses enkripsi dan dekripsi pada server dapat berjalan dengan lancar, maka user harus menunduh file Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy yang berupa “jce_policy-8.zip”. Setelah proses unduh selesai, maka user akan mengekstrak file local_policy.jar and US_export_policy.jar dan mengganti file - file dengan nama yang sama di dalam folder instalasi java, tepatnya pada jre/lib/security/.

11. Menunduh file Owncloud yang terdapat di dalam website berupa “owncloud-9.1.2.zip” dan mengekstrak folder tersebut ke dalam direktori server apache, yaitu “/var/www/html”.
12. Pada phpmyadmin, Login sebagai “root” dan akan dilakukan pembuatan database yang akan digunakan oleh Owncloud dan user baru mysql sebagai administrator dari server, user baru tersebut akan diberikan izin untuk melakukan insert, update, dan delete terhadap database dari owncloud yang akan digunakan.
13. Setelah proses penambahan user selesai, maka buka halaman web utama dari browser, lalu akan dilakukan setting pertama oleh administrator. Setting pertama yang dilakukan adalah setting dari direktori data user, username dan password dari administrator, username, password, dan nama database dari mysql, dan memilih mysql sebagai database yang digunakan. Jika sudah selesai, maka proses instalasi telah selesai.

4.2 Implementasi Login dan Verifikasinya pada Aplikasi Mobile

Fungsi Login pada aplikasi ini berguna agar user dapat masuk dan mengakses file – file yang berada di dalam server. Fungsi Login ini dibagi menjadi beberapa bagian

4.2.1 Setting Credential pada Owncloud

Pada bagian ini, user akan diminta untuk memasukkan username dan password masing - masing. Setiap aplikasi akan memiliki *Client* dan *Handler* masing – masing, yang berasal dari API yang dimiliki oleh Owncloud. Client berfungsi untuk menyimpan data dari Username dan Password user, sedangkan Handler digunakan untuk menjalankan request – request yang diminta oleh user. Deklarasi dari client dan handler Owncloud dapat dilihat pada gambar ini.

Segmen Program 4.1 Class client dan Handler Owncloud

```
public static OwnCloudClient mainclient;  
public static Handler mainhandler;
```

Setelah pendeklarasian, maka user akan mengambil data dari username dan password yang diinputkan, dan proses pengecekan *credential* dari Owncloud dimulai. Proses dapat dilihat pada segmen program 4.2.

Segmen Program 4.2 Fungsi untuk memanggil login

```
public void trylogin(View view)
{
    EditText usr = (EditText)
findViewById(R.id.username);
    EditText pass = (EditText)
findViewById(R.id.password);
    String username = usr.getText().toString();
    String password = pass.getText().toString();
    mainclient.setCredentials(

OwnCloudCredentialsFactory.newBasicCredentials(
                                username,
                                password
                                )

    );

    GetRemoteUserInfoOperation gg = new
GetRemoteUserInfoOperation();
    gg.execute(mainclient, this, mainhandler);

}
```

Setelah melakukan setting pada client, maka akan dilakukan pengambilan data dari user, jika berhasil, maka proses akan dialihkan pada *listener* yang berfungsi untuk mendapatkan response dari server. Proses dapat dilihat pada gambar ini.

Segmen Program 4.3 Penghandle'an hasil dari fungsi login

```
public void onRemoteOperationFinish(RemoteOperation caller,
RemoteOperationResult result) throws IOException,
NoSuchAlgorithmException {
    if (!result.isSuccess())
    {
        Toast.makeText(this, "Username or Password is
Wrong", Toast.LENGTH_SHORT).show();
    }
    else if (caller instanceof
GetRemoteUserInfoOperation)
    {
```



4.2.2 Verifikasi Menggunakan E-mail

Setelah Listener berhasil mendapatkan data dari user, maka akan segera dilakukan pengecekan apakah user sudah memasukkan e-mail ke dalam server atau tidak, jika belum maka user akan diminta untuk memasukkan e-mail. Data tersebut akan dikirim ke server untuk disimpan. Proses di client dapat dilihat pada gambar ini dan proses pada server dapat dilihat pada gambar itu.

Segmen Program 4.4 kode php untk mengabil data e-mail

```
if (isset($_POST['getemail'])) {
    $username = $_POST['username'];
    $query = mysqli_query($con, "SELECT value FROM
`oc_cards_properties` join oc_cards on oc_cards.id =
oc_cards_properties.cardid WHERE uri =
'Database: ".$username.".vcf' and name = 'email'");
    if (mysqli_num_rows($query) == 0)
    {
        echo "no email found";
    }
    else
    {
        $row = mysqli_fetch_array($query);
        echo $row['value'];
    }
}
```

Segmen Program 4.5 Fungsi di aplikasi mobile untu menambah e-mail

```
private String addemail(String username, String email) throws
IOException {
    URL url = new
URL("http://203.189.121.208:125/aaa/getemail.php");
    String response = "";
    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setReadTimeout(50000);
    conn.setConnectTimeout(30000);
```

```

conn.setRequestMethod("POST");
conn.setDoInput(true);
conn.setDoOutput(true);

Uri.Builder builder = new Uri.Builder()
    .appendQueryParameter("setemail","1")
    .appendQueryParameter("email",email)
    .appendQueryParameter("username", username);

String query = builder.build().getEncodedQuery();

OutputStream os = conn.getOutputStream();
BufferedWriter writer = new BufferedWriter(
    new OutputStreamWriter(os, "UTF-8"));
writer.write(query);
writer.flush();
writer.close();
os.close();

conn.connect();

int responseCode=conn.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {
    String line;
    BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    while ((line=br.readLine()) != null) {
        response+=line;
    }
}
else {
    response="";
}
Log.d("RESULT",response);
return response;
}

```

Segmen Program 4.6 Fugsi di php untuk menyimpan e-mail

```

elseif (isset($_POST['setemail'])) {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $query = mysqli_query($con,"SELECT id FROM `oc_cards` where
uri = 'Database:".$username.".vcf'");
    $row = mysqli_fetch_array($query);
    $id = $row[0];
    $query = mysqli_query($con,"INSERT INTO oc_cards_properties
values('','2','$id','EMAIL','$email','0')");
}

```

Setelah dilakukan pengecekan e-mail, maka akan dilakukan proses pengecekan apakah user telah memiliki public key dan private key di dalam *smartphone*. Jika belum maka akan dilakukan pengecekan apakah user telah memiliki private key dan public key untuk handphone lain. Jika user telah memiliki public key, maka aplikasi akan mengirimkan *request* untuk mendapatkan key kepada administrator. Jika user baru pertama kali masuk ke dalam server melalui *smartphone*, atau *requestnya* telah disetujui oleh administrator, maka user server akan mengenerate private key dan public key. Public key akan disimpan di dalam server, sedangkan private key akan disimpan juga di dalam server namun dengan nama yang telah diacak. Setelah itu server akan mengirim e-mail kepada user untuk menunduh private key untuk aplikasi, dan aplikasi akan meminta private key tersebut jika user berusaha untuk menginstall.

Segmen Program 4.7 Fungsi untuk mengecek -email

```

else if (devicekeycheck.equals("check email")){
    File privatekeypath = new
File(this.getFilesDir(),"privatekey-
"+mainclient.getCredentials().getUsername()+"-"+
Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID)+".bin");
    if (privatekeypath.exists()){
        Intent openmainpage = new Intent(this,
mainpage.class);
        this.startActivity(openmainpage);
        EditText usr = (EditText)
findViewById(R.id.username);
        EditText pass = (EditText)
findViewById(R.id.password);
        String username =
usr.getText().toString();
        String password =
pass.getText().toString();
        acc = new savedaccount(username,password);
    }
    else {
        new AlertDialog.Builder(this)
            .setTitle("Alert")
            .setMessage("Do you want to import
your private key?")
            .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                @Override
                public void

```

```

onClick(DialogInterface dialog, int which) {
    Intent i = new
Intent(MainActivity.this, FilePickerActivity.class);
i.putExtra(FilePickerActivity.EXTRA_ALLOW_MULTIPLE, false);

i.putExtra(FilePickerActivity.EXTRA_ALLOW_CREATE_DIR, false);

i.putExtra(FilePickerActivity.EXTRA_MODE,
FilePickerActivity.MODE_FILE);
i.putExtra(FilePickerActivity.EXTRA_START_PATH,
Environment.getExternalStorageDirectory().getPath());
startActivityForResult(i,
FILE_CODE);
    }
    })
    .setNegativeButton("No", new
DialogInterface.OnClickListener() {
        @Override
        public void
onClick(DialogInterface dialog, int which) {

Toast.makeText(MainActivity.this, "You can only login when you have
put your private key", Toast.LENGTH_SHORT).show();
        }
    })
    .show();
    }
}

```

Segmen Program 4.8 Fungsi untuk mengambil ID Device

```

elseif(isset($_POST['getdeviceid']))
{
    $username = $_POST['username'];
    $deviceid = $_POST['deviceid'];

    $query = mysqli_query($con, "SELECT * from oc_publickeys where
uid = '$username'");
    if (mysqli_num_rows($query)==0) {
        $privatekeypath = generateRandomString(15).".bin";
        $publickeypath = "../owndata/keys/publickey-" . $username . "-"
        . $deviceid . ".bin";
        shell_exec('java -jar KeyGenerator.jar
"' . $privatekeypath . '" "' . $publickeypath . '"');
        $query = mysqli_query($con, "INSERT into oc_publickeys
values ('$username', '$deviceid', '$publickeypath')");
        echo "device granted";
        $query2 = mysqli_query($con, "SELECT value FROM
`oc_cards_properties` join oc_cards on oc_cards.id =
oc_cards_properties.cardid WHERE uri =
'Database:.'" . $username . ".vcf' and name = 'email'");

        $row = mysqli_fetch_array($query2);
        $email = $row['value'];
    }
}

```

```

sendmail($email,'http://192.168.43.90/aaa/' . $privatekeypath);
    echo "email sent";
}
else{
    $query2 = mysqli_query($con,"SELECT * from oc_publickeys
where uid = '$username' and deviceid = '$deviceid'");
    if (mysqli_num_rows($query2)==0){
        $query3 = mysqli_query($con,"SELECT * from oc_requests
where userid = '$username' and deviceid = '$deviceid'");
        if
(mysqli_num_rows($query3)==0){mysqli_query($con,"INSERT into
oc_requests values ('','$username','$deviceid','0')");echo "access
requested";}
        else{
            $row = mysqli_fetch_array($query3);
            if ($row['status']==0)
                echo "wait for access";
            else echo "check email";
        }
    }
    else{
        echo "check email";
    }
}
}
elseif ($_POST['getrequest']) {
    $query2 = mysqli_query($con,"SELECT * from oc_requests where
status = '0'");
    $response = "";
    while($row = mysqli_fetch_array($query2)){
        $response.=$row['userid'].'<rr>'.$row['deviceid'].'<br>';
    }
    echo $response;
}
}

```

Segmen Program 4.9 Fungsi untuk mengenerate key RSA

```

public static void main(String[] args) throws
NoSuchAlgorithmException, FileNotFoundException, IOException {
    // TODO code application logic here

    File privatekey = new File(args[0]);
    File publickey = new File(args[1]);
    KeyPairGenerator keygen =
    KeyPairGenerator.getInstance("RSA");
    keygen.initialize(2048);
    FileOutputStream privatekeygen = new
    FileOutputStream(privatekey);
    FileOutputStream publickeygen = new
    FileOutputStream(publickey);
    KeyPair keys = keygen.generateKeyPair();
    privatekeygen.write(keys.getPrivate().getEncoded());
    publickeygen.write(keys.getPublic().getEncoded());
    privatekeygen.close();
}

```

```

        publicKeygen.close();
    }

```

Segmen Program 4.10 Fungsi untuk mengirim e-mail

```

<?php
$headers = 'From: <admin@petracloud.com>' . "\n" .
'Reply-To: <michaeladige@gmail.com>';
$headers .= "Content-Type: text/html; charset=UTF-8\r\n";
$email = $_POST['email'];
$privatekey = $_POST['privatekey'];
$message = "<p>Hi, Thank you for registering your device, here's
your private key, unduh it by clicking this link:</p>
<br>
". '<a href="'. $privatekey. '>Link to your private key </a>';
mail('<'. $email. '>', 'Your private key', $message, $headers,
'-fwebmaster@example.com');
?>

```

Di dalam proses *request*, Administrator dapat memilih apakah user dengan device yang digunakan dapat masuk ke dalam aplikasi, jika iya, maka proses yang terjadi akan sama seperti saat user pertama kali masuk dengan device pertamanya.

Segmen Program 4.11 Fungsi untuk mengambil request

```

private String getrequest() throws IOException {
    URL url = new
URL("http://203.189.121.208:125/aaa/getemail.php");
    String response = "";
    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setReadTimeout(50000);
    conn.setConnectTimeout(30000);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);

    Uri.Builder builder = new Uri.Builder()
        .appendQueryParameter("getrequest", "1");
    String query = builder.build().getEncodedQuery();

    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(
        new OutputStreamWriter(os, "UTF-8"));
    writer.write(query);
    writer.flush();
    writer.close();
    os.close();
}

```

```

        conn.connect();

        int responseCode=conn.getResponseCode();

        if (responseCode == HttpsURLConnection.HTTP_OK) {
            String line;
            BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            while ((line=br.readLine()) != null) {
                response+=line;
            }
        }
        else {
            response="";
        }
        Log.d("RESULT",response);
        return response;
    }

```

Segmen Program 4.12 Fungsi untuk menset status dari e-mail

```

private String setrequest(String username, String deviceid, String
status) throws IOException {
    URL url = new
URL("http://203.189.121.208:125/aaa/getemail.php");
    String response = "";
    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setReadTimeout(50000);
    conn.setConnectTimeout(30000);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);

    Uri.Builder builder = new Uri.Builder()
        .appendQueryParameter("setrequest", "1")
        .appendQueryParameter("username",username)
        .appendQueryParameter("deviceid",deviceid)
        .appendQueryParameter("status",status);
    String query = builder.build().getEncodedQuery();

    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(
        new OutputStreamWriter(os, "UTF-8"));
    writer.write(query);
    writer.flush();
    writer.close();
    os.close();

    conn.connect();

    int responseCode=conn.getResponseCode();

    if (responseCode == HttpsURLConnection.HTTP_OK) {

```

```

        String line;
        BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        while ((line=br.readLine()) != null) {
            response+=line;
        }
    }
    else {
        response="";
    }
    Log.d("RESULT",response);
    return response;
}

```

4.3 Implementasi Membaca Folder pada Aplikasi Mobile

Pada proses ini user akan dapat melihat folder di dalam folder server. Aplikasi akan mengirimkan permintaan dengan *client* dan *handler* untuk melihat folder yang berada di dalam server. Jika berhasil, maka hasil dari request tersebut akan dimasukkan ke dalam ListView oleh aplikasi.

Segmen Program 4.13 Fungsi untuk membaca folder

```

public void StartRefresh(String path)
{
    ReadRemoteFolderOperation refreshOperation = new
ReadRemoteFolderOperation(path);
    refreshOperation.execute(MainActivity.mainclient, this,
MainActivity.mainhandler);
}
private void onSuccessfullRefresh(ReadRemoteFolderOperation
operation, RemoteOperationResult result) {
    filelist.clear();

    List<RemoteFile> files = new ArrayList<RemoteFile>();
    for(Object obj: result.getData()) {
        files.add((RemoteFile) obj);
    }
    if (files != null) {
        Iterator<RemoteFile> it = files.iterator();
        while (it.hasNext()) {
            filelist.add(it.next());
        }
        filelist.remove(filelist.getItem(0));
    }
    filelist.notifyDataSetChanged();
}

```

```

        //aaaa.setRefreshing(false);
    }

    public View getView(int position, View convertView, ViewGroup
parent) {
        View vi = convertView;
        if (convertView == null)
        {
            vi =
View.inflate(getContext(), R.layout.file_in_list, null);

            TextView textView =
(TextView)vi.findViewById(R.id.filename);
            ImageView fileimg =
(ImageView)vi.findViewById(R.id.filetype);
            if (getmime(position).equals("DIR"))
                fileimg.setImageResource(R.drawable.folder);
            else if (getmime(position).equals("image/png") ||
getmime(position).equals("image/jpeg"))
                fileimg.setImageResource(R.drawable.file_image);
            else if
(getmime(position).equals("application/vnd.openxmlformats-
officedocument.wordprocessingml.document"))
                fileimg.setImageResource(R.drawable.file_doc);
            else if (getmime(position).equals("application/pdf"))
                fileimg.setImageResource(R.drawable.file_pdf);
            else fileimg.setImageResource(R.drawable.file_text);
            String[] filenameee =
getItem(position).getRemotePath().split("/");
            textView.setText(filenameee[filenameee.length-1]);
            return vi;
        }
    }

```

4.4 Implementasi Unggah pada Aplikasi Mobile

Pada bagian ini, user akan menekan tombol dengan logo plus di pojok kanan bawah dan memilih Unggah. Penjelasan akan dibagi ke dalam bagian – bagian ini.

4.4.1 Penggunaan *Filepicker* dalam memilih file

Filepicker akan digunakan di dalam memilih file yang akan diunggah. Setelah user memilih maka proses - proses selanjutnya akan dilakukan.

Segmen Program 4.14 Fungs untuk membuka picker

```

public void openpicker()
{
    Intent i = new Intent(getContext(),

```

```

FilePickerActivity.class);
        i.putExtra(FilePickerActivity.EXTRA_ALLOW_MULTIPLE,
false);
        i.putExtra(FilePickerActivity.EXTRA_ALLOW_CREATE_DIR,
false);
        i.putExtra(FilePickerActivity.EXTRA_MODE,
FilePickerActivity.MODE_FILE);
        .
        i.putExtra(FilePickerActivity.EXTRA_START_PATH,
Environment.getExternalStorageDirectory().getPath());
        startActivityForResult(i, FILE_CODE);

    }

```

4.4.2 Enkripsi File dengan metode AES

Key AES yang digunakan bersifat random pada tiap sesi unggah yang dilakukan. Setelah key AES degenerate, maka proses enkripsi file dimulai. Metode yang digunakan adalah AES 128-bit. Panjang byte yang dihasilkan adalah 16 byte, dan key AES yang dibentuk akan diubah dari byte menjadi String untuk memberi kemudahan di dalam proses pengiriman key ke server.

Segmen Program 4.15 Fungsi untuk enkripsi file dengan AES

```

private String randomizekey()
{
    String EncodedKey = "";
    SecureRandom rand = new SecureRandom();
    byte[] x = new byte[16];
    rand.nextBytes(x);
    EncodedKey = Base64.encodeToString(x, Base64.NO_WRAP);
    return EncodedKey;
}

private void encrypt(String key, File newfileunggah, File
afterencrypt)
{
    try{
        FileInputStream input = new
FileInputStream(newfileunggah);
        FileOutputStream output = new
FileOutputStream(afterencrypt);
        byte k[] = key.getBytes();
        SecretKeySpec seckey = new SecretKeySpec(k, "AES");
        Cipher enc = Cipher.getInstance("AES");
        enc.init(Cipher.ENCRYPT_MODE, seckey);
        CipherOutputStream cos = new
CipherOutputStream(output, enc);
        byte[] buf = new byte[1024];

```

```

        int read;
        while((read=input.read(buf))!=-1)
        {
            cos.write(buf,0,read);
        }
        input.close();
        output.flush();
        cos.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

```

4.4.3 Enkripsi key AES dengan *private key*

Setelah file dienkripsi, maka key AES yang digunakan dalam proses sebelumnya akan dienkripsi oleh private key yang dimiliki oleh user di dalam device. Proses ini akan menghasilkan String AES yang dienkripsi.

Segmen Program 4.16 Sfungi untuk menenkripsi key RSA

```

private String encryptAESwithRSA(String key) throws Exception {
    File privatekeyfile = new
File(getContext().getFilesDir(),"privatekey-
"+MainActivity.mainclient.getCredentials().getUsername()+"-"+
Settings.Secure.getString(getContext().getContentResolver(),
Settings.Secure.ANDROID_ID)+".bin");
    byte[] utf8 = key.getBytes("UTF8");
    FileInputStream fis = new
FileInputStream(privatekeyfile);
    byte k[] = readFromStream(fis);
    PKCS8EncodedKeySpec pkcs = new PKCS8EncodedKeySpec(k);
    KeyFactory kf = KeyFactory.getInstance("RSA");
    PrivateKey seckey = kf.generatePrivate(pkcs);
    Cipher enc = Cipher.getInstance("RSA");
    enc.init(Cipher.ENCRYPT_MODE, seckey);
    // Encrypt
    byte[] res = enc.doFinal(utf8);
    // Encode bytes to base64 to get a string
    return Base64.encodeToString(res,Base64.NO_WRAP);
}
public byte[] readFromStream(InputStream inputStream)
throws Exception {
    ByteArrayOutputStream baos = new
ByteArrayOutputStream();
    DataOutputStream dos = new DataOutputStream(baos);
    byte[] data = new byte[4096];
    int count = inputStream.read(data);
    while (count != -1) {

```

```

        dos.write(data, 0, count);
        count = inputStream.read(data);
    }

    return baos.toByteArray();
}

```

4.4.4 Proses Unggah File ke server

Setelah proses sebelumnya, file yang telah dienkrpsi akan diunggah ke dalam server menggunakan API Owncloud yang telah disediakan. File akan masuk ke dalam sesuai dengan direktori user yang dibuka di dalam aplikasi pada saat itu. File yang telah diunggah akan didekripsi pada proses berikutnya.

Segmen Program 4.17 Proses unggah file kes server

```

Uri uri = data.getData();
// Do something with the URI
dd = System.currentTimeMillis();
File newfileunggah = new File (uri.getPath());
File encryptedfile = new File
(Environment.getExternalStorageDirectory().getAbsolutePath() +
"/" + newfileunggah.getName());
key = randomizekey();
deleteencrypted =
Environment.getExternalStorageDirectory().getAbsolutePath() +
"/" + newfileunggah.getName();
encrypt(key,newfileunggah,encryptedfile);
String remotePath = currentpath +
newfileunggah.getName();

UnggahRemoteFileOperation unggahOperation = new
UnggahRemoteFileOperation(encryptedfile.getAbsolutePath(),
remotePath, "text");

unggahOperation.addDataTransferProgressListener(this);
mProgressDialog.show();

unggahOperation.execute(MainActivity.mainclient, this,
MainActivity.mainhandler);
lastunggahefile = remotePath;

```

Segmen Program 4.18 Fungsi yang dipanggil seteah file berhasil terunggah

```

else if (caller instanceof UnggahRemoteFileOperation ) {
    mProgressDialog.dismiss();
    try {
        String encrypteddaes = encryptAESwithRSA(key);
    }
}

```

```

http(lastunggahefile,MainActivity.mainclient.getCredentials().getUsername(),encryptedaes);
    new File(deleteencrypted).delete();
} catch (IOException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
StartRefresh(currentpath);
}

```

4.4.5 Proses Dekripsi key AES dengan public key

Setelah file masuk ke dalam server, maka aplikasi akan mengirim key AES ke server untuk didekripsi. Key AES akan didekripsi dengan public key yang sesuai dengan username dan device yang dimiliki oleh user. Server akan menggunakan command line untuk menggunakan file jar untuk mendekripsi file.

Segmen Program 4.19 Fungs untuk mengirim key AES ke server

```

private void http(String filepath, String username, String encryptedaes) throws IOException {
    URL url = new URL("http://203.189.121.208:125/aaa/");
    String response = "";
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(50000);
    conn.setConnectTimeout(30000);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);

    Uri.Builder builder = new Uri.Builder()
        .appendQueryParameter("deviceid", Settings.Secure.getString(getContext().getContentResolver(), Settings.Secure.ANDROID_ID))
        .appendQueryParameter("filepath", filepath)
        .appendQueryParameter("username", username)
        .appendQueryParameter("aeskey", encryptedaes);
    String query = builder.build().getEncodedQuery();

    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-8"));
    writer.write(query);
    writer.flush();
    writer.close();
    os.close();
}

```

```

        conn.connect();

        int responseCode=conn.getResponseCode();

        if (responseCode == HttpURLConnection.HTTP_OK) {
            String line;
            BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            while ((line=br.readLine()) != null) {
                response+=line;
            }
        }
        else {
            response="";
        }
        Log.d("RESULT",response);
    }

```

Segmen Program 4.20 Poses di server untuk mengolah file engan key AES

```

<?php
$filepath = $_POST['filepath'];
$username = $_POST['username'];
$aeskey = $_POST['aeskey'];
$deviceid = $_POST['deviceid'];
$con =
mysqli_connect('localhost','owncloud','owncloud','owncloud_ownc
louddb2');
$query = mysqli_query($con,"select path, oc_storages.id from
oc_filecache join oc_storages on oc_filecache.storage =
oc_storages.numeric_id where path = 'files'.".$filepath."");
$publickey = "../owndata/keys/publickey-".$username."-
".$deviceid.".bin";

if (mysqli_num_rows($query)>1)
{
    while($row=mysqli_fetch_array($query))
    {
        if (substr($row[1], 6) == $username)
            $userpath = $username;
    }
}
elseif (mysqli_num_rows($query)==1) {
    $userpath = $username;
}else{
    $query2 = mysqli_query($con,"select uid_owner from
oc_shares where share_with = '$username' and file_target =
'$filepath'");
    if (mysqli_num_rows($query2)>0)
    {
        $row = mysqli_fetch_array($query2);
        $userpath = $row['share_with'];
    }
}

```

```

        else echo "file not found";
    }

    $file = '../owndata/' . $userpath . '/files' . $filepath;
    //echo $aeskey;
    // $file = "../owndata/admin/files/avatar.jpg";
    echo shell_exec('java -jar JavaApplication1.jar "' . $file . '"
    "' . $aeskey . '" "' . $publickey . '"');
}
?>

```

Segmen Program 4.21 Fungsi untuk mendecrypt key AES

```

public static String decryptaeswithRSA(String key, File
PublicKey) throws FileNotFoundException,
NoSuchAlgorithmException, InvalidKeySpecException,
NoSuchPaddingException, InvalidKeyException,
IllegalBlockSizeException, BadPaddingException, Exception
{
    byte[] decodedaes = Base64.getDecoder().decode(key);
    FileInputStream fis = new FileInputStream(PublicKey);
    byte k[] = readFromStream(fis);
    X509EncodedKeySpec pkcs = new X509EncodedKeySpec(k);
    KeyFactory kf = KeyFactory.getInstance("RSA");
    PublicKey seckey = kf.generatePublic(pkcs);
    Cipher enc = Cipher.getInstance("RSA", "BC");
    enc.init(Cipher.DECRYPT_MODE, seckey);
    byte[] result = enc.doFinal(decodedaes);
    String realaes = new String(result);
    return realaes;
}

public static byte[] readFromStream(InputStream
inputStream) throws Exception {
    ByteArrayOutputStream baos = new
ByteArrayOutputStream();
    DataOutputStream dos = new DataOutputStream(baos);
    byte[] data = new byte[4096];
    int count = inputStream.read(data);
    while (count != -1) {
        dos.write(data, 0, count);
        count = inputStream.read(data);
    }

    return baos.toByteArray();
}

```

4.4.6 Proses Dekripsi file di dalam server dengan metode AES

File yang telah diunggah ke server akan didekripsi dengan key AES yang juga telah didekripsi oleh private key. File juga didekripsi dengan menggunakan aplikasi java sama dengan langkah sebelumnya.

Segmen Program 4.22 Fungs untuk menekripsi file

```
public static void main(String[] args) throws
FileNotFoundException, IOException, NoSuchAlgorithmException,
InvalidKeyException, NoSuchProviderException,
NoSuchPaddingException, InvalidKeySpecException,
IllegalBlockSizeException, BadPaddingException, Exception {
    // TODO code application logic here
    Security.addProvider(new BouncyCastleProvider());
    String filepath = args[0];
    String aeskey = args[1];
    String publickey = args[2];
    File fileunggahdafuq = new File (filepath);
    File publickeyfile = new File (publickey);
    //File savedaeskey = new File (filepath+".key");
    String key = null;
    key = decryptaeswithRSA(aeskey,publickeyfile);
    System.out.println(key);
    File afterencrypt = new File ("decrypted.jpg");
    FileInputStream input = new
FileInputStream(fileunggahdafuq);
    FileOutputStream output = new
FileOutputStream(afterencrypt);
    byte k[]=key.getBytes();
    SecretKeySpec seckey = new SecretKeySpec(k,"AES");
    Cipher enc = Cipher.getInstance("AES", "BC");
    enc.init(Cipher.DECRYPT_MODE,seckey);
    CipherOutputStream cos = new CipherOutputStream(output,
enc);
    byte[] buf;
    buf = new byte[1024];
    int read;
    while((read=input.read(buf))!=-1)
    {
        cos.write(buf,0,read);
    }
    input.close();
    output.flush();
    cos.close();
    fileunggahdafuq.delete();
    afterencrypt.renameTo(fileunggahdafuq);
}
```

4.5 Implementasi Pembuatan Folder pada Aplikasi Mobile

Pembuatan folder akan dieksekusi dengan user menekan tombol di pojok kanan bawah dan menekan pilihan folder. User akan menginputkan nama folder dan menekan tombol create folder pada dialog yang muncul. Pembuatan folder juga menggunakan API Owncloud dan hasilnya berupa folder yang berada di dalam direktori yang sedang dibuka oleh user pada saat itu. Setelah folder dibuat, maka proses *refresh* direktori tersebut akan dilakukan.

Segmen Program 4.23 Proses membuat folder

```
CreateRemoteFolderOperation createfolder = new
CreateRemoteFolderOperation(newfolder, false);

createfolder.execute(MainActivity.mainclient, NewFolderDialog.this, MainActivity.mainhandler);

fragment.StartRefresh(currentpath);
dismiss();
```

4.6 Implementasi Unduh pada Aplikasi Mobile

Proses Unduh akan dijalankan apabila user memilih file dari list yang ada dan menekan tombol unduh pada dialog yang muncul. Proses Unduh ini dibagi menjadi bagian – bagian sebagai berikut.

4.6.1 Proses Enkripsi File pada server dengan metode AES

File yang berada di dalam server akan dicari terlebih dahulu sesuai dengan username dari user, setelah file tersebut ditemukan, maka akan dilakukan proses enkripsi di dalam server dengan key AES yang juga di generate random. Proses ini akan menghasilkan file temporary yang telah dienkripsi.

Segmen Program 4.24 Proses mencari file di server

```
<?php
$filepath = $_POST['filepath'];
$username = $_POST['username'];
$deviceid = $_POST['deviceid'];
// $aeskey = $_POST['aeskey'];
```

```

$con =
mysql_connect('localhost','owncloud','owncloud','owncloud_own
cloudodb2');
$query = mysql_query($con,"select path, oc_storages.id from
oc_filecache join oc_storages on oc_filecache.storage =
oc_storages.numeric_id where path = 'files'.".$filepath.'" and
oc_storages.id = 'home:.'"$username.'"");
$publickey = "../owndata/keys/publickey-".$username.'"-'
".$deviceid.'"bin";
$date = date('m-d-Y h:i:s a');

if (mysql_num_rows($query)>1)
{
    while($row=mysql_fetch_array($query))
    {
        if (substr($row[1], 6) == $username)
            $userpath = $username;
    }
}
elseif (mysql_num_rows($query)==1) {
    $userpath = $username;
}
else{
    $query2 = mysql_query($con,"select uid_owner from
oc_share where share_with = '$username' and file_target =
'$filepath' and share_type = '0'");
    if (mysql_num_rows($query2)>0)
    {
        $row = mysql_fetch_array($query2);
        $userpath = $row['uid_owner'];
    }
    elseif(mysql_num_rows($query2)==0) {
        $query2 = mysql_query($con,"SELECT uid_owner FROM
`oc_share` WHERE `share_type`='1' and file_target =
'$filepath' and share_with in (select gid from oc_group_user
where uid = '$username')");
        if (mysql_num_rows($query2)>0)
        {
            $row = mysql_fetch_array($query2);
            $userpath = $row['uid_owner'];
        }
        else echo "file not found";
    }
}
}
$file = '../owndata/'$userpath.'/files'.$filepath;
$targetfile = $file . " " . $date.".temp";
//echo $aeskey;
//$file = "../owndata/admin/files/avatar.jpg";
echo shell_exec('java -jar JavaApplication2.jar
"$publickey.'" "$file.'" "$targetfile.'"')."<br>";
echo $targetfile;
?>

```

Segmen Program 4.25 Proses Enkripsi file dengan AES

```
private static String randomizekey()
{
    String EncodedKey = "";
    SecureRandom rand = new SecureRandom();
    byte[] x = new byte[16];
    rand.nextBytes(x);
    EncodedKey = Base64.getEncoder().encodeToString(x);
    return EncodedKey;
}

private static void encrypt(String key, File
fileunggahdafaq, File afterencrypt)
{
    try{
        FileInputStream input = new
FileInputStream(fileunggahdafaq);
        FileOutputStream output = new
FileOutputStream(afterencrypt);
        byte k[]=key.getBytes();
        SecretKeySpec seckey = new SecretKeySpec(k,"AES");
        Cipher enc = Cipher.getInstance("AES","BC");
        enc.init(Cipher.ENCRYPT_MODE,seckey);
        CipherOutputStream cos = new
CipherOutputStream(output, enc);
        byte[] buf = new byte[16];
        int read;
        while((read=input.read(buf))!=-1)
        {
            cos.write(buf,0,read);
        }
        input.close();
        output.flush();
        cos.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

4.6.2 Proses Enkripsi key AES dengan public key pada server

Setelah dilakukan enkripsi file, maka key juga akan dienkripsi menggunakan public key sesuai dengan username dan device dari user. Hasilnya adalah String berisi key AES yang telah dienkripsi

Segmen Program 4.26 Fungsi untuk menenkripsi key AES

```
private static String encryptAESwithRSA(String key, File
publickeyfile) throws Exception {

    byte[] utf8 = key.getBytes("UTF8");
```

```

        FileInputStream fis = new FileInputStream(publickeyfile);
        byte k[] = readFromStream(fis);
        X509EncodedKeySpec pkcs = new X509EncodedKeySpec(k);
        KeyFactory kf = KeyFactory.getInstance("RSA");
        PublicKey seckey = kf.generatePublic(pkcs);
        Cipher enc = Cipher.getInstance("RSA","BC");
        enc.init(Cipher.ENCRYPT_MODE, seckey);
        // Encrypt
        byte[] res = enc.doFinal(utf8);
        // Encode bytes to base64 to get a string
        return Base64.getEncoder().encodeToString(res);
    }
    public static byte[] readFromStream(InputStream inputStream)
    throws Exception {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        DataOutputStream dos = new DataOutputStream(baos);
        byte[] data = new byte[4096];
        int count = inputStream.read(data);
        while (count != -1) {
            dos.write(data, 0, count);
            count = inputStream.read(data);
        }

        return baos.toByteArray();
    }
}

```

4.6.3 Proses Unduh File ke dalam device

Proses unduh pada device menggunakan class *AsyncTask* yang dimodifikasi untuk menunduh file. Aplikasi mendapatkan response Key AES dan path dari file terenkripsi sementara yang dihasilkan dari proses pada server. File yang diunduh akan dimasukkan ke dalam folder unduh.

Segmen Program 4.27 Fungsi menunduh File

```

private void startUnduh (String path) throws Exception {
    File downFolder=
    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
    Y_UNDUHS);
    String response =
    encryptserver(path,MainActivity.mainclient.getCredentials().getUse
    rname());
    String[] arrresponse = response.split("<br>");
    arrresponse[1] =
    "http://203.189.121.208:125"+arrresponse[1].substring(2);
    arrresponse[1] = arrresponse[1].replaceAll(" ", "%20");
    Log.d("RESPONSE", response);

    String unduhpath =
    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
    Y_UNDUHS).getAbsolutePath()+"/"+path.substring(currentpath.length(
    ));
}

```

```

        byte[] AESkey = decryptAESwithRSA(arrresponse[0]);
        File afterdecrypted = new
File(Environment.getExternalStorageDirectory().getAbsolutePath()+
/"+path.substring(currentpath.length()));
        UnduhFileFromURL dltask = new
UnduhFileFromURL(unduhpath,this,AESkey,afterdecrypted);
        dltask.execute(arrresponse[1]);

```

Segmen Program 4.28 Fungsi untuk mendapatkan data di server

```

protected String doInBackground(String... f_url) {

    int count;
    try {
        String fileURL = f_url[0];
        if (fileURL.contains(" ")) {
            fileURL = fileURL.replace(" ", "%20");
        }
        URL url = new URL(fileURL);
        URLConnection connection = url.openConnection();
        connection.connect();
        // getting file length
        int lengthOfFile = connection.getContentLength();

        // input stream to read file - with 8k buffer
        InputStream input = new
BufferedInputStream(url.openStream(), 8192);

        OutputStream output = new
FileOutputStream(filepath);

        byte data[] = new byte[1024];

        long total = 0;
        while ((count = input.read(data)) != -1) {
            total += count;
            // publishing the progress....
            // After this onProgressUpdate will be called
            publishProgress("" + (int) ((total * 100) /
lengthOfFile));
            dialog.setProgress((int)(total * 100) /
lengthOfFile);
            // writing data to file
            output.write(data, 0, count);
        }

        // flushing output
        output.flush();

        // closing streams
        output.close();
        input.close();

```

```

        return "SUCCESS";

    } catch (Exception e) {
        return "FAILED";
    }
}

```

4.6.4 Proses Dekripsi Key AES dengan private key

Key AES yang didapatkan dari dalam server akan didekripsi terlebih dahulu dengan private key yang dimiliki oleh user sesuai dengan device yang digunakan oleh user.

Segmen Program 4.29 Fungsi untuk endekripsi key AES di aplikasi

```

private byte[] decryptAESwithRSA(String key) {
    File privatekeyfile = new
File(getContext().getFilesDir(), "privatekey-
"+MainActivity.mainclient.getCredentials().getUsername()+"-"+
Settings.Secure.getString(getContext().getContentResolver(),
Settings.Secure.ANDROID_ID)+".bin");
    //File aeskey = new File(key);
    //FileInputStream aeskeyinput = new
FileInputStream(aeskey);
    //byte[] utf8 = readFromStream(aeskeyinput);
    byte[] res = null;
    try{
        byte[] utf8 = Base64.decode(key, Base64.NO_WRAP);
        FileInputStream fis = new
FileInputStream(privatekeyfile);
        byte k[] = readFromStream(fis);
        PKCS8EncodedKeySpec pkcs = new PKCS8EncodedKeySpec(k);
        KeyFactory kf = KeyFactory.getInstance("RSA");
        PrivateKey seckey = kf.generatePrivate(pkcs);
        Cipher enc = Cipher.getInstance("RSA");
        enc.init(Cipher.DECRYPT_MODE, seckey);
        // Encrypt
        res = enc.doFinal(utf8);
        // Encode bytes to base64 to get a string
    }
    catch (Exception e){
        e.printStackTrace();
        Toast.makeText(getContext(), "Wrong or Invalid
private key!", Toast.LENGTH_SHORT).show();
    }
    return res;
}

```

4.6.5 Proses Dekripsi File di dalam device

File sementara yang didapatkan dari server akan didekripsi dengan key AES yang telah didekripsi dengan private key user setelah proses unduh file sementara selesai. File hasil dekripsi akan dimasukkan ke dalam folder Unduh dan file sementara akan dihapus.

Segmen Program 4.30 Fungsi untuk mendekripsi file oleh key AES

```
public void decrypt(byte[] k, File newfileunggah, File
afterencrypt)
{
    try{
        FileInputStream input = new
FileInputStream(newfileunggah);
        FileOutputStream output = new
FileOutputStream(afterencrypt);
        //byte k[]=key.getBytes("UTF-8");
        SecretKeySpec seckey = new SecretKeySpec(k,"AES");
        Cipher enc = Cipher.getInstance("AES");
        enc.init(Cipher.DECRYPT_MODE,seckey);
        CipherOutputStream cos = new
CipherOutputStream(output, enc);
        byte[] buf = new byte[16];
        int read;
        while((read=input.read(buf))!=-1)
        {
            cos.write(buf,0,read);
        }
        input.close();
        output.flush();
        cos.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
        Toast.makeText(getApplicationContext(),"Wrong or Invalid
private key!",Toast.LENGTH_SHORT).show();
    }
}
```

4.7 Implementasi Share pada Aplikasi Mobile

Proses Share dimulai ketika user menekan tombol share di dalam dialog yang muncul saat user memilih file.

4.7.1 Proses Pengambilan Data Share

Pada saat dialog share pertama kali dibuka, maka akan muncul list dari kepada siapa file ini telah dishare. Pengambilan data ini menggunakan API Owncloud.

Segmen Program 4.31 Fungsi untuk mengambil list data share

```
public void onRemoteOperationFinish(RemoteOperation caller,
RemoteOperationResult result) throws Exception {
    if (caller instanceof GetRemoteSharesForFileOperation)
    {
        shareadapter.clear();
        if (result.isSuccess()) {
            ArrayList< OCShare > shares = new ArrayList<
OCShare >();
            for (Object obj: result.getData()) {
                shares.add(( OCShare) obj);
            }
            TextView nouters =
(TextView)findViewById(R.id.shareNoUsers);

            if (shares.size()>0)
            {
                sharelist.setVisibility(View.VISIBLE);
                nouters.setVisibility(View.INVISIBLE);
            }
            else
            {
                sharelist.setVisibility(View.GONE);
                nouters.setVisibility(View.VISIBLE);
            }
            for (int i=0;i<shares.size();i++)
            {
                shareadapter.add(shares.get(i));
            }
            shareadapter.notifyDataSetChanged();
            // do your stuff here
        }
    }
}
```

4.7.2 Proses Share

Proses Share dilakukan menekan tombol Add Button, akan muncul list dari user – user dan group yang belum di share. Setelah user memilih data yang akan di share, maka proses share akan dimulai dengan menggunakan

API Owncloud. Jika proses selesai, maka list user yang ter-share akan di refresh.

Segmen Program 4.32 Fungsi untuk mengambil user dan group di server

```
<?php
$con =
mysqli_connect("localhost","owncloud","owncloud","owncloud_ownc
louddb2");
if (isset($_POST['getusers']))
{
    $filepath = $_POST['filepath'];
    $user = $_POST['user'];
    $sql = 'select uid from oc_users where uid not in(SELECT
uid FROM `oc_users` WHERE uid in (select share_with from
oc_share where uid_initiator = "'. $user.'" and file_target =
"'. $filepath.'" and share_type = "0")) and uid != "'. $user.'"
and uid not in (select uid_owner from oc_share where
uid_initiator = "'. $user.'" and file_target =
"'. $filepath.'"');
    $res = mysqli_query($con,$sql);
    while($row = mysqli_fetch_array($res))
    {
        echo $row['uid'] . ' ';
    }
}
if (isset($_POST['getgroups']))
{
    $filepath = $_POST['filepath'];
    $user = $_POST['user'];
    $sql = 'select gid from oc_groups where gid not in(SELECT
gid FROM `oc_users` WHERE gid in (select share_with from
oc_share where uid_initiator = "'. $user.'" and file_target =
"'. $filepath.'" and share_type = "1"))';
    $res = mysqli_query($con,$sql);
    while($row = mysqli_fetch_array($res))
    {
        echo $row['gid'] . ' ';
    }
}
?>
```

```
groupList.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
        CreateRemoteShareOperation share = new
CreateRemoteShareOperation(filepath,
ShareType.GROUP,useradapter.getItem(position),false, "", 31);
share.execute(MainActivity.mainclient,ShareUsersDialog.this,Mai
nActivity.mainhandler);
    }
});
```



4.7.3 Proses Pemberhentian Share

Proses ini dimulai ketika user menekan tombol tong sampah di sebelah list user. Setelah tombol dipencet, maka aplikasi akan memanggil API Owncloud yang digunakan di dalam penghapusan Share.

Segmen Program 4.33 Fungsi untuk memberhentikan share

```
public void onClick(View v) {
    RemoveRemoteShareOperation operation = new
    RemoveRemoteShareOperation((int)getItem(position).getRemoteId()
    );
    operation.execute( MainActivity.mainclient,
    ShareAdapter.this, MainActivity.mainhandler);
}

@Override
public void onRemoteOperationFinish(RemoteOperation caller,
RemoteOperationResult result) throws Exception {
    if (caller instanceof RemoveRemoteShareOperation) {
        sharedialog.refreshsharelist();
    }
}
```

4.8 Implementasi Delete pada Aplikasi Mobile

Proses Delete dilakukan setelah user menekan file atau menekan dengan lama folder yang akan dihapus dan memilih pilihan delete pada Dialog yang muncul. Proses Delete akan menggunakan API dari Owncloud untuk menghapus file tersebut. Setelah proses delete dilakukan, maka direktori akan di-refresh.

Segmen Program 4.34 Fungsi untuk menghapus file

```
private void startDelete(String remotePath){
    RemoveRemoteFileOperation removeOperation = new
    RemoveRemoteFileOperation(remotePath);
    removeOperation.execute(MainActivity.mainclient, this,
    MainActivity.mainhandler);
}
```

```
    }

    else if (caller instanceof RemoveRemoteFileOperation ) {
        Toast.makeText (getContext () , "File
Deleted", Toast.LENGTH_SHORT).show ();
        StartRefresh (currentpath);
    }
```