BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada tahap ini akan dijelaskan mengenai implementasi dari aplikasi pengelolaan informasi kegiatan ionosfer dan telekomunikasi yang dibuat. Implementasi dari perancangan aplikasi ini menggunakan eclipse sebagai software aplikasi program dan bahasa pemrograman Java untuk logic android, XML untuk layout, My SQL untuk database, PHP untuk komunikasi android ke My SQL,.

5.1. Implementasi

Implementasi adalah tahap penerapan sistem yang telah dibuat pada tahap perancangan sistem supaya bisa untuk dioperasikan. Pada tahap ini akan dijelaskan mengenai implementasi perangkat lunak, implementasi perangkat keras, implementasi basis data, implementasi instalasi program dan implementasi antar muka.

Aplikasi tersebut dapat dijalankan hanya pada platform android, dengan implementasi dan pengujian yang akan dilakukan menggunakan perangkat keras Galaxy Tab P3100 (Galaxy Tab 2 7.0 inci). Aplikasi ini dapat berjalan, apabila memenuhi standar minimal dari perangkat keras *(hardware).*

5.1.1. Batasan Implementasi (Optional)

Dalam mengimplementasikan perangkat lunak pemantauan dan pengendalian perangkat lunak ini ada beberapa hal yang menjadi batasan implementasi, yaitu :

1. Aplikasi ini difokuskan pada approval hasil penelitian kegiatan ionosfer dan telekomunikasi.
2. Basis data yang digunakan dalam mengimplementasikan aplikasi tersebut adalah MySQL.
3. Proses menuliskan kode program dengan menggunakan Eclipse Helios (eclipse versi 3.6) sebagai editor utama dengan menggunakan bahasa pemrograman java untuk logic androidnya.
4. Implementasi aplikasi ini dilakukan pada Tablet Android dengan resolusi minimum 3.3" WQVGA (240 x 400: ldpi).

5.1.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada sistem komputer yang digunakan untuk membangun Aplikasi Pengelolaan Informasi Kegiatan Ionosfer dan Telekomunikasi berbasis *Android* di LAPAN ini adalah sebagai berikut:

1. Eclipse Helios (eclipse versi 3.6)
2. JDK (*Java Development Kit)* dan JRE *(Java Runtime Environment).*
3. Android SDK 8.0 *(Software Development Kit).*
4. ADT (Android Development Tools).
5. Sistem Operasi Windows 7
6. My SQL sebagai *Database*
7. PHP sebagai komunikasi Android ke My SQL dengan format JSON
8. Notepad ++

5.1.3. Implementasi Perangkat Keras

Dalam menerapkan dari rancangan yang telah dijelaskan sebelumnya dibutuhkan beberapa perangkat keras untuk menyajikan aplikasi ini. Adapun alat-alat yang dibutuhkan adalah :

1. Laptop
    a. Processor : Genuine Intel(R) CPU U7300 @1.30GHz
    b. Memory / RAM : 2 GB
    c. Harddisk : 320 Gb
    d. Sistem : 32 bit
    e. Layar : 14.1 inchi

2. Tablet
    a. Processor : Dual core 1GHz, PowerVR SGX540 GPU, TI OMAP 4430 chipset
    b. Memory / RAM : 1 GB
    c. eksternal MicroSD : 8 GB
    d. Jenis Layar : *tourchscreen* Kapasitif PLS LCD, 600 x 1024 piksel, 7.0 inci (~170 ppi pixel density)
    e. Konektivitas : GPRS, EDGE, HSDPA, 21 Mbps, Wi-Fi 802.11, Wi-Fi Direct, dual-band, Wi-Fi hotspot, Bluetoothv3.0 with A2DP, HS, microUSB v3.0 with A2DP

5.1.4. Implementasi Basis Data (Sintaks SQL)

Implementasi basis data dilakukan dengan menggunakan DBMS yang digunakan adalah MySQL. Adapun implementasi basis data dan rss feed dengan format data xml adalah sebagai berikut :

a. Implementasi Basis Data Konfigurasi ke database yang dituju

```
<?php

/*
 * All database connection variables
 */

define('DB_USER', "root"); // db user
define('DB_PASSWORD', ""); // db password (mention your db password here)
define('DB_DATABASE', "lapan_ion"); // database name
define('DB_SERVER', "localhost"); // db server
?>
```

b. Implementasi Basis Data Penyediaan Fungsi untuk membuat Koneksi

```
<?php

/**
 * A class file to connect to database
```

```
 */
class DB_CONNECT {

    // constructor
    function __construct() {
        // connecting to database
        $this->connect();
    }

    // destructor
    function __destruct() {
        // closing db connection
        $this->close();
    }

    /**
     * Function to connect with database
     */
    function connect() {
        // import database connection variables
        require_once 'db_config.php';
```

```
        // Connecting to mysql database
        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or
    die(mysql_error());

        // Selecing database
        $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or
    die(mysql_error());

        // returing connection cursor
        return $con;
    }

    /**
     * Function to close db connection
     */
    function close() {
        // closing db connection
        mysql_close();
    }
}
```

c. Implementasi Basis Data Get Approval

```
<?php
```

```
/*
 * Following code will list all the products
 */

// array for JSON response
$response = array();

// include db connect class
require_once 'db_connect.php';

// connecting to db
$db = new DB_CONNECT();
// get all products from products table
$result = mysql_query("SELECT *FROM ion where status = 0") or die(mysql_error());

// check for empty result
if (mysql_num_rows($result) > 0) {
    // looping through all results
    // products node
    $response["ion"] = array();
```

```
    while ($row = mysql_fetch_array($result)) {
        // temp user array
        $product = array();
        $product["id"] = $row["id"];
        $product["title"] = $row["title"];
        $product["date"] = $row["date"];
                    $product["description"] = $row["description"];
                    $product["status"] = $row["status"];

        // push single product into final response array
        array_push($response["ion"], $product);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
} else {
    // no products found
    $response["success"] = 0;
    $response["message"] = "No products found";
```

```
    // echo no users JSON
    echo json_encode($response);
}
?>
```

d. Implementasi Basis Data Get Ionosfer

```
<?php

/*
 * Following code will list all the products
 */

// array for JSON response
$response = array();

// include db connect class
require_once 'db_connect.php';

// connecting to db
$db = new DB_CONNECT();
// get all products from products table
$result = mysql_query("SELECT *FROM ion where status=1") or
die(mysql_error());
```

```
// check for empty result
if (mysql_num_rows($result) > 0) {
    // looping through all results
    // products node
    $response["matahari"] = array();

    while ($row = mysql_fetch_array($result)) {
        // temp user array
        $product = array();
        $product["id"] = $row["id"];
        $product["title"] = $row["title"];
        $product["date"] = $row["date"];
                $product["description"] = $row["description"];
                $product["status"] = $row["status"];

        // push single product into final response array
        array_push($response["matahari"], $product);
    }
    // success
    $response["success"] = 1;
```

```
    // echoing JSON response
    echo json_encode($response);
} else {
    // no products found
    $response["success"] = 0;
    $response["message"] = "No products found";

    // echo no users JSON
    echo json_encode($response);
}
?>
```

e. Implementasi Basis Data Get Schedule

```
<?php

/*
 * Following code will list all the products
 */

// array for JSON response
$response = array();

// include db connect class
```

```
require_once 'db_connect.php';

// connecting to db
$db = new DB_CONNECT();
// get all products from products table
$result = mysql_query("SELECT *FROM schedule") or die(mysql_error());

// check for empty result
if (mysql_num_rows($result) > 0) {
    // looping through all results
    // products node
    $response["ion"] = array();

    while ($row = mysql_fetch_array($result)) {
        // temp user array
        $product = array();
        $product["id"] = $row["id"];
        $product["title"] = $row["title"];
        $product["descr"] = $row["descr"];
                    $product["tanggal"] = $row["tanggal"];

        // push single product into final response array
```

```
        array_push($response["ion"], $product);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
} else {
    // no products found
    $response["success"] = 0;
    $response["message"] = "No products found";

    // echo no users JSON
    echo json_encode($response);
}
?>
```

f. Implementasi Basis Data Insert Schedule

```
<?php

/*
 * Following code will create a new product row
 * All product details are read from HTTP Post Request
```

```
 */

// array for JSON response
$response = array();

// check for required fields
if (isset($_GET['title']) && isset($_GET['descr'])&& isset($_GET['tanggal']))
{

   $title = $_GET['title'];
   $desc = $_GET['descr'];
         $date = $_GET['tanggal'];

   // include db connect class
   require_once 'db_connect.php';

   // connecting to db
   $db = new DB_CONNECT();

   // mysql inserting a new row
   $result = mysql_query("Insert into schedule(title,descr,tanggal)
Values('$title','$desc','$date')");
```

```
    // check if row inserted or not
    if ($result) {
        // successfully inserted into database
        $response["success"] = 1;
        $response["message"] = "Product successfully created.";

        // echoing JSON response
        echo json_encode($response);
    } else {
        // failed to insert row
        $response["success"] = 0;
        $response["message"] = "Oops! An error occurred.";

        // echoing JSON response
        echo json_encode($response);
    }
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
```

```
    // echoing JSON response
    echo json_encode($response);
}
?>
```

g. Implementasi Basis Data Update Approval

```
<?php

/*
 * Following code will create a new product row
 * All product details are read from HTTP Post Request
 */

// array for JSON response
$response = array();

// check for required fields
if (isset($_GET['id'])) {

    $id = $_GET['id'];
        $kat = $_GET['kat'];
    // include db connect class
    require_once 'db_connect.php';
```

```
    // connecting to db
    $db = new DB_CONNECT();

    // mysql inserting a new row
        if($kat==1)
            $result = mysql_query("Update ion set status = 1 where
id='$id'");
        else
            $result = mysql_query("Update ion set status = 2 where
id='$id'");
    // check if row inserted or not
    if ($result) {
        // successfully inserted into database
        $response["success"] = 1;
        $response["message"] = "Product successfully created.";

        // echoing JSON response
        echo json_encode($response);
    } else {
        // failed to insert row
        $response["success"] = 0;
```

```
        $response["message"] = "Oops! An error occurred.";

        // echoing JSON response
        echo json_encode($response);
    }
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";

    // echoing JSON response
    echo json_encode($response);
}
?>
```

h. Implementasi Basis Data Validate Login

```
<?php

/*
 * Following code will list all the products
 */

// array for JSON response
```

```
$response = array();

// include db connect class
require_once 'db_connect.php';

// connecting to db
$db = new DB_CONNECT();
// get all products from products table
$user = $_GET['user'];
$pass = md5($_GET['pass']);
$result = mysql_query("SELECT *FROM user where username = '$user' and
password='$pass'") or die(mysql_error());

// check for empty result
if (mysql_num_rows($result) > 0) {
   $response["user"] = array();
          while ($row = mysql_fetch_array($result)) {
       // temp user array
       $product = array();
       $product["id"] = $row["id"];
                  $product["phone"] = $row["phone"];
                  $product["nama"] = $row["nama"];
```

```
        // push single product into final response array
        array_push($response["user"], $product);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
} else {
    // no products found
    $response["success"] = 0;
    $response["message"] = "No products found";

    // echo no users JSON
    echo json_encode($response);
}
?>
```

5.1.5. Implementasi Antar Muka

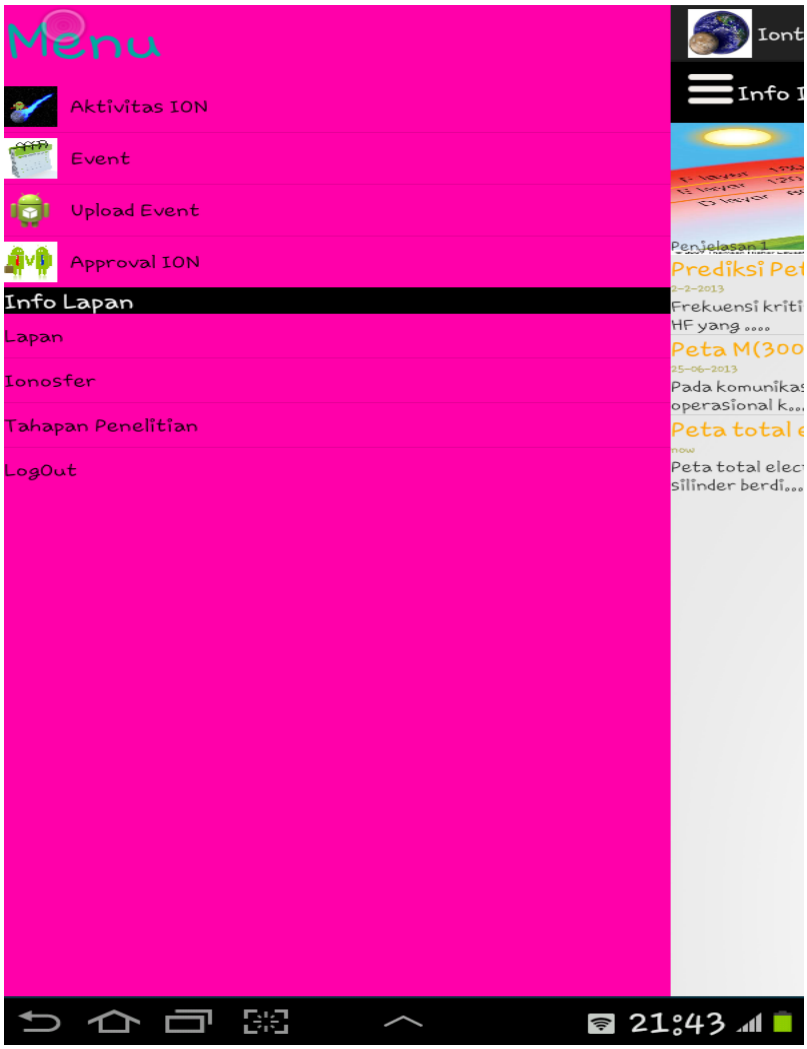
Implementasi antarmuka merupakan tahapan dalam memenuhi kebutuhan pengguna (*user*) dalam berinteraksi dengan komputer. Fasilitas antar muka yang baik sangat membantu pemakai dalam memahami proses yang sedang dilakukan oleh sistem tersebut dan dapat meningkatkan kinerja sistem. Berikut beberapa penjelasan dari bagaimana aplikasi ini bekerja dengan tampilan aplikasi yang dibuat.

1. Tampilan Form Login

   Form Login ini adalah tampilan saat pertama kali aplikasi dijalankan untuk masuk menu utama admin. Untuk masuk menu utama, admin harus mengisi *username* dan *password* kemudian klik button *login.*

1. Implementasi Tampilan Menu Utama Admin

Dibawah ini merupakan implementasi tampilan utama sistem yang diusulkan



Gambar 5.1 Implementasi Menu Utama

2. Implementasi Tampilan Menu Aktivitas Ionosfer

Dibawah ini merupakan implementasi tampilan menu aktivitas ionosfer sistem yang diusulkan



Gambar 5.2 Implementasi Tampilan Menu Aktivitas Ionosfer

3. Implementasi Tampilan Menu Approval

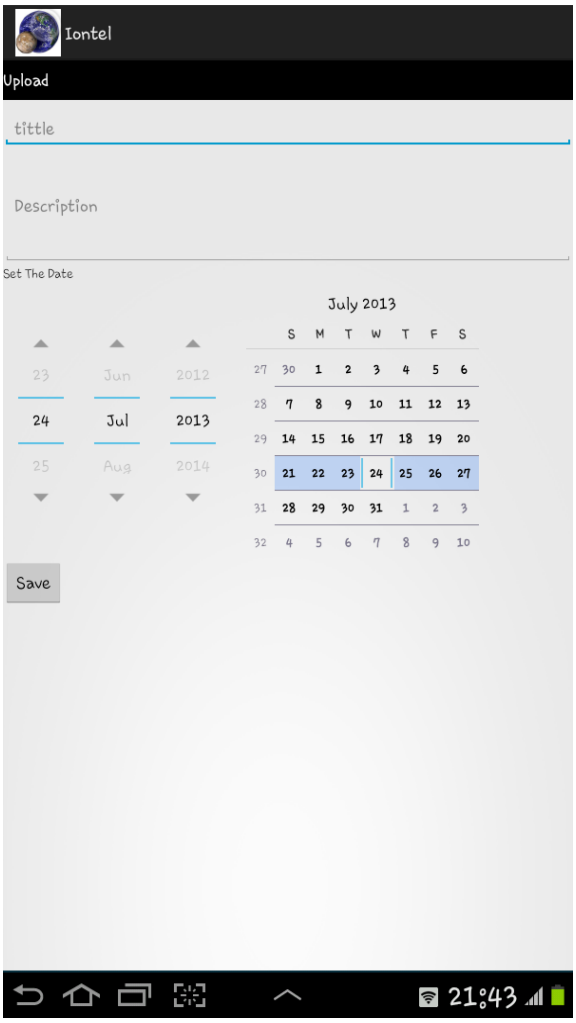
Dibawah ini merupakan implementasi menu approval sistem yang diusulkan



Gambar 5.3 Implementasi Tampilan Menu Approval

4. Implementasi Tampilan Menu Upload Event

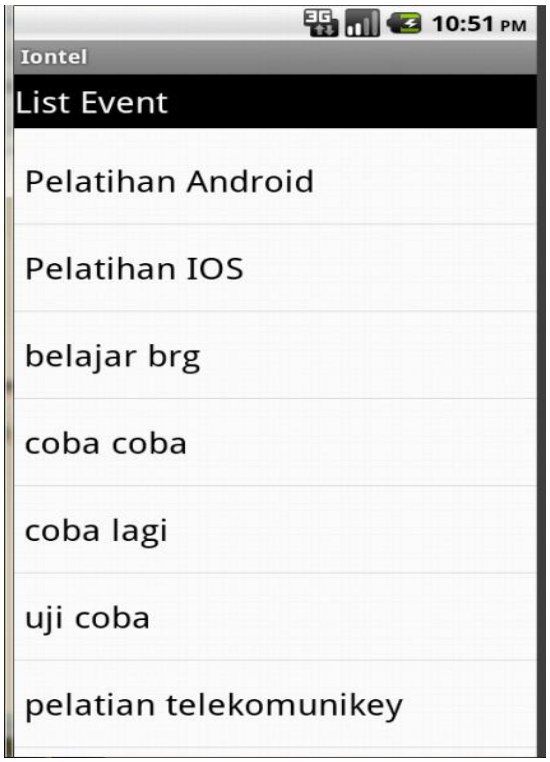
Dibawah ini merupakan implementasi menu upload Event sistem yang diusulkan



Gambar 5.4 Implementasi Tampilan Menu Upload Schedule

5. Implementasi Tampilan Menu List Event

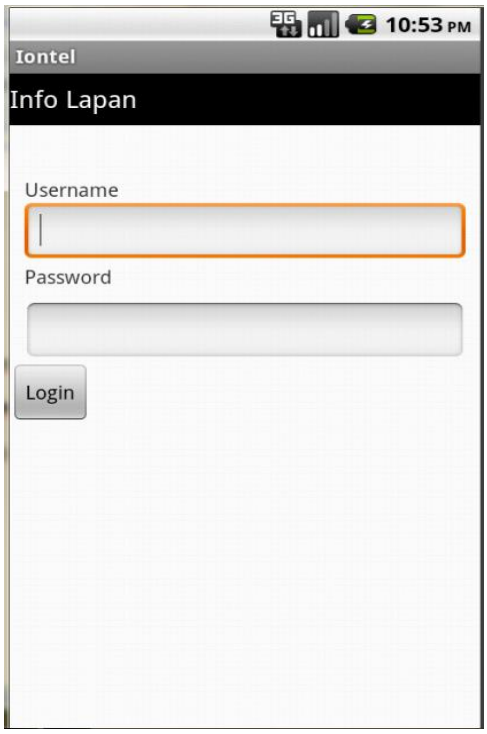
Dibawah ini merupakan implementasi menu list schedule sistem yang diusulkan



Gambar 5.5 Implementasi Tampilan Menu List Schedule

6. Implementasi Tampilan *Form* Login Admin

Dibawah ini merupakan implementasi tampilan *form* Login Admin sistem yang diusulkan



Gambar 5.6 Implementasi *Form* Login Admin

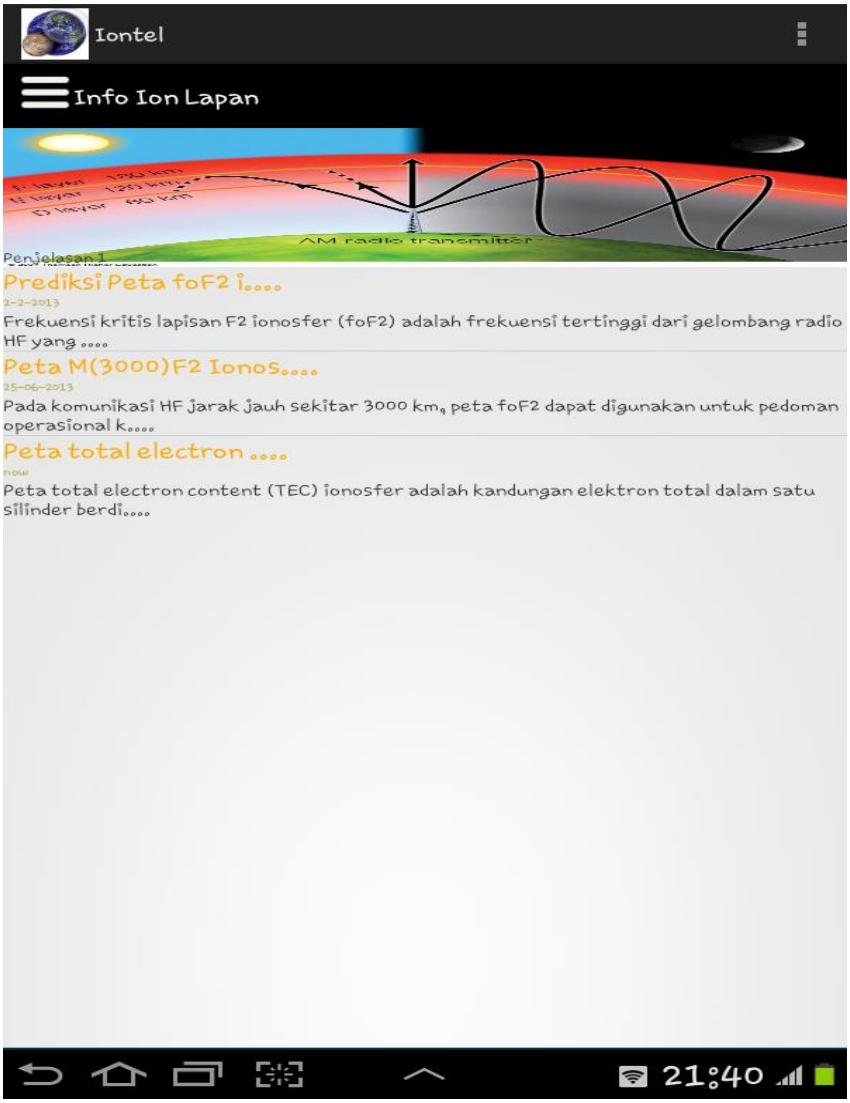
7. Implementasi Tampilan Menu Utama *User*

Dibawah ini merupakan implementasi menu utama *User* sistem yang diusulkan



Gambar 5.7 Implementasi Tampilan Menu *User*

8. Implementasi Menu Home



Gambar 5.8 Implementasi Tampilan Menu *User*

9. Implementasi Menu Info Lapan



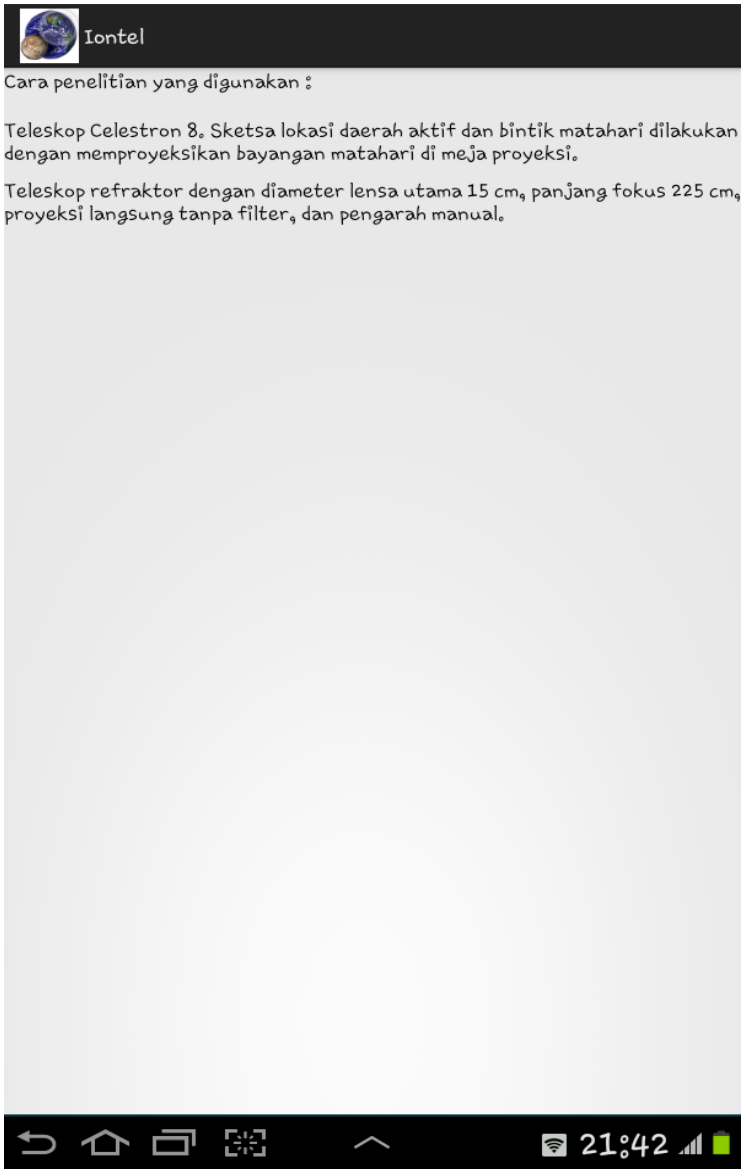
Gambar 5.9 Implementasi Tampilan Menu Info Lapan

10. Implementasi Tampilan Menu Tahapan Penelitian



Gambar 5.10 Implementasi Tampilan Menu Tahapan Penelitian

11. Implementasi Tampilan Menu Cara Penelitian



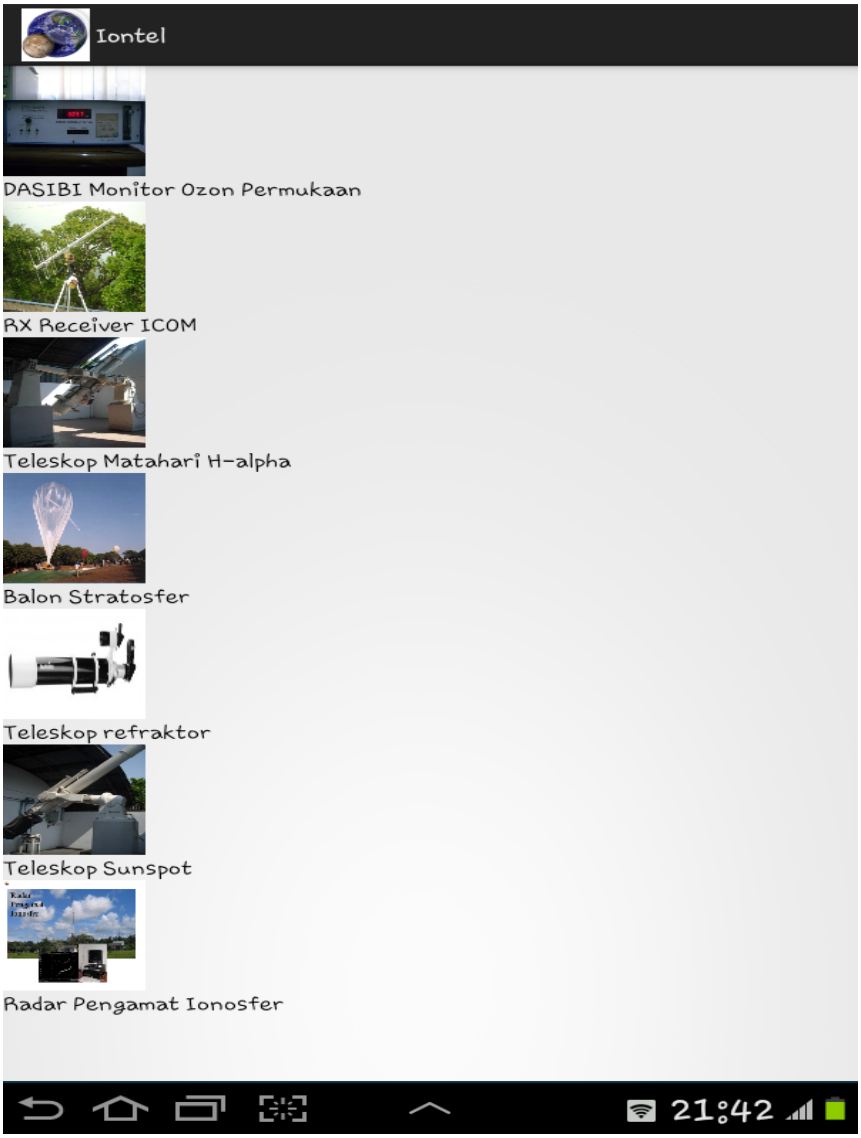
Gambar 5.11 Implementasi Tampilan Menu Cara Penelitian

12. Implementasi Tampilan Menu Tentang Penelitian



Gambar 5.12 Implementasi Tampilan Menu Tentang Penelitian

13. Implementasi Tampilan Menu Alat Penelitian

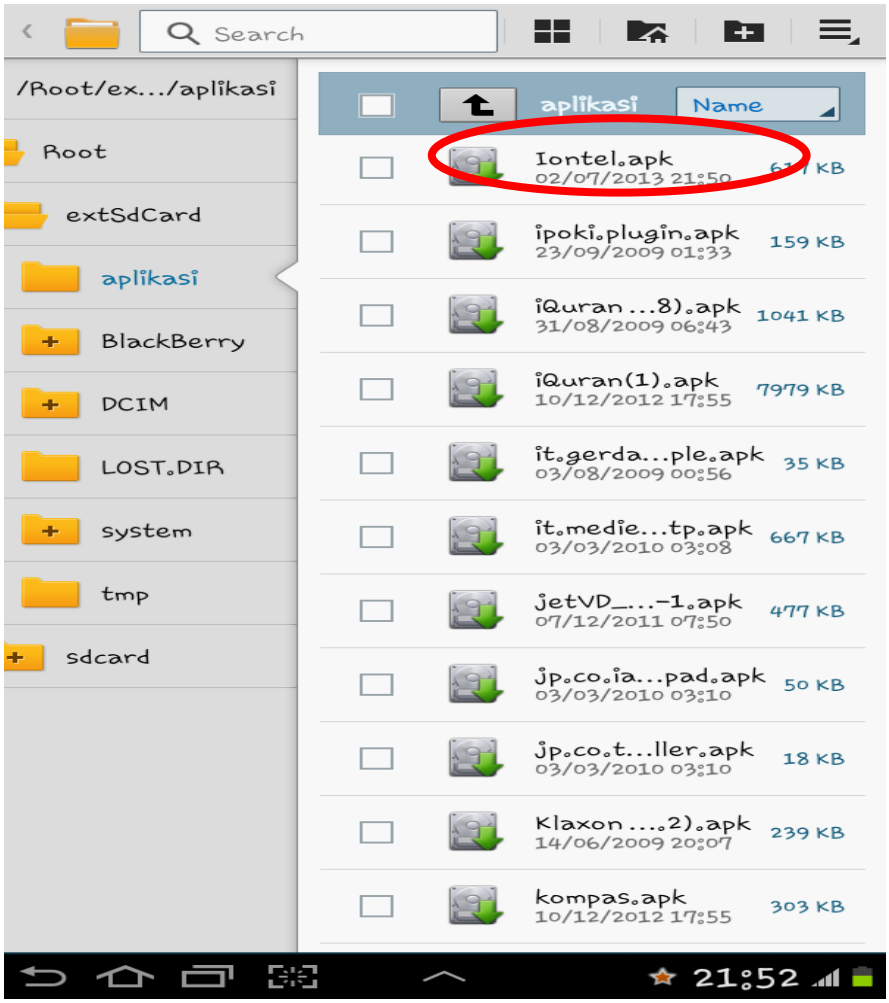


Gambar 5.13 Implementasi Tampilan Menu Alat Penelitian

5.1.6 Implementasi Instalasi Program

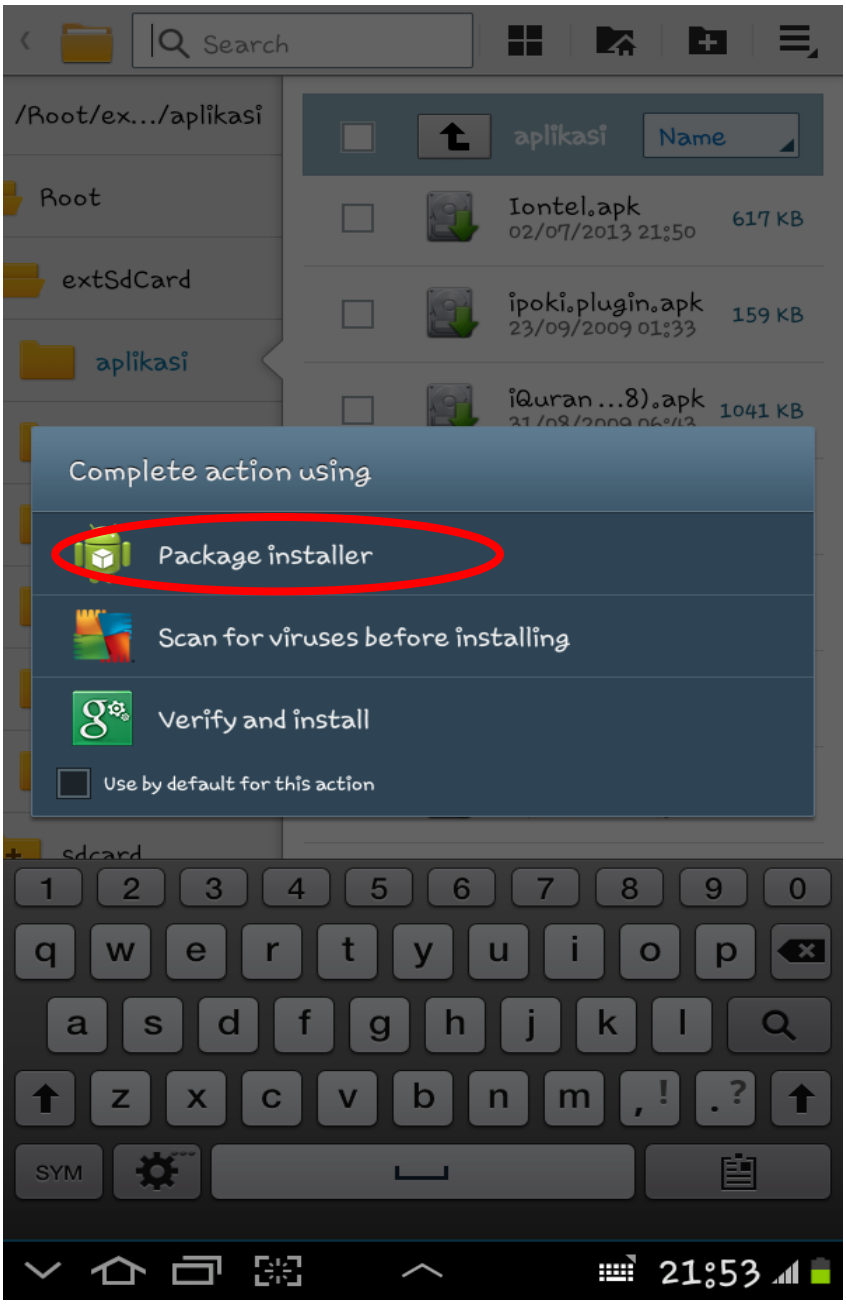
Untuk Proses instalasi program dilakukan seperti biasa tanpa ada perbedaan ketika mengistalasi program-program pada umumnya. Berikut ini adalah implementasi instalasi di Samsung Galaxy Tab untuk menggunakan Aplikasi Pengelolan Informasi Berbasis Android di Lembaga Penerbangan dan Antariksa Nasional (LAPAN). Langkah-langkahnya adalah sebagai berikut :

1. Buka file Iontel.apk



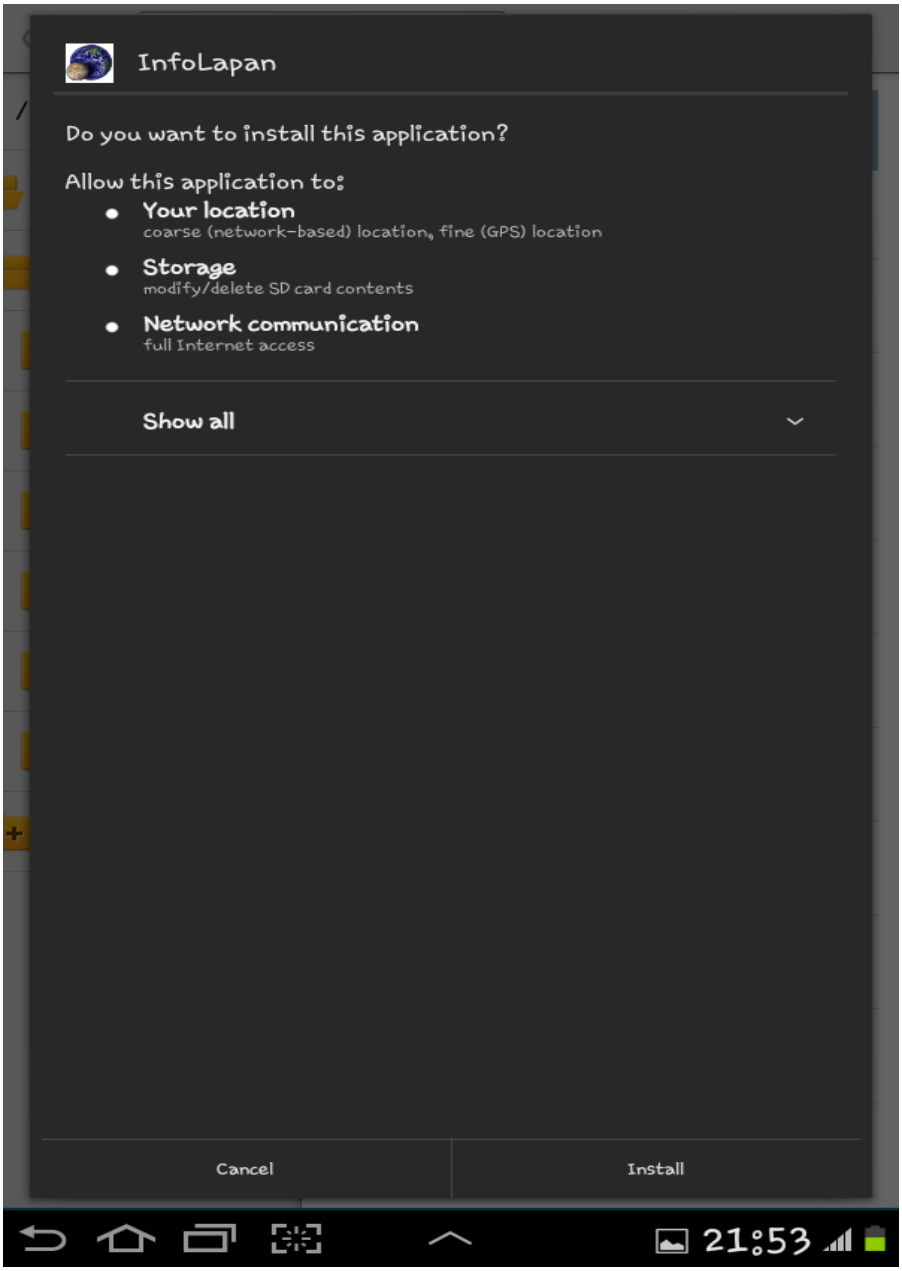
Gambar 5.14 Tampilan file iontel.apk

2. Klik file Iontel.apk tadi maka akan muncul tampilan seperti dibawah ini.



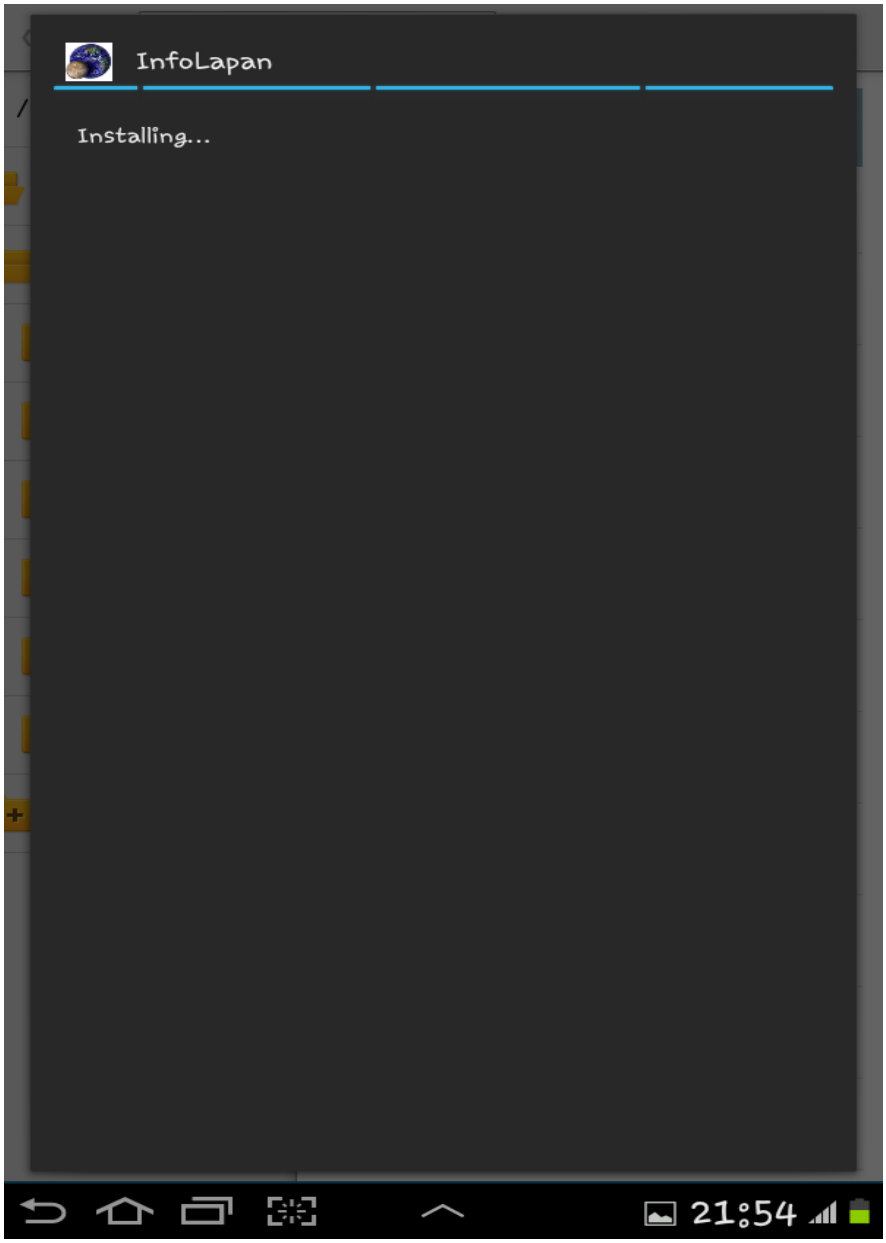
Gambar 5.15 Tampilan untuk package installer.apk

3. **Klik tombol *package installer* lalu akan muncul seperti pada tampilan di bawah ini.**



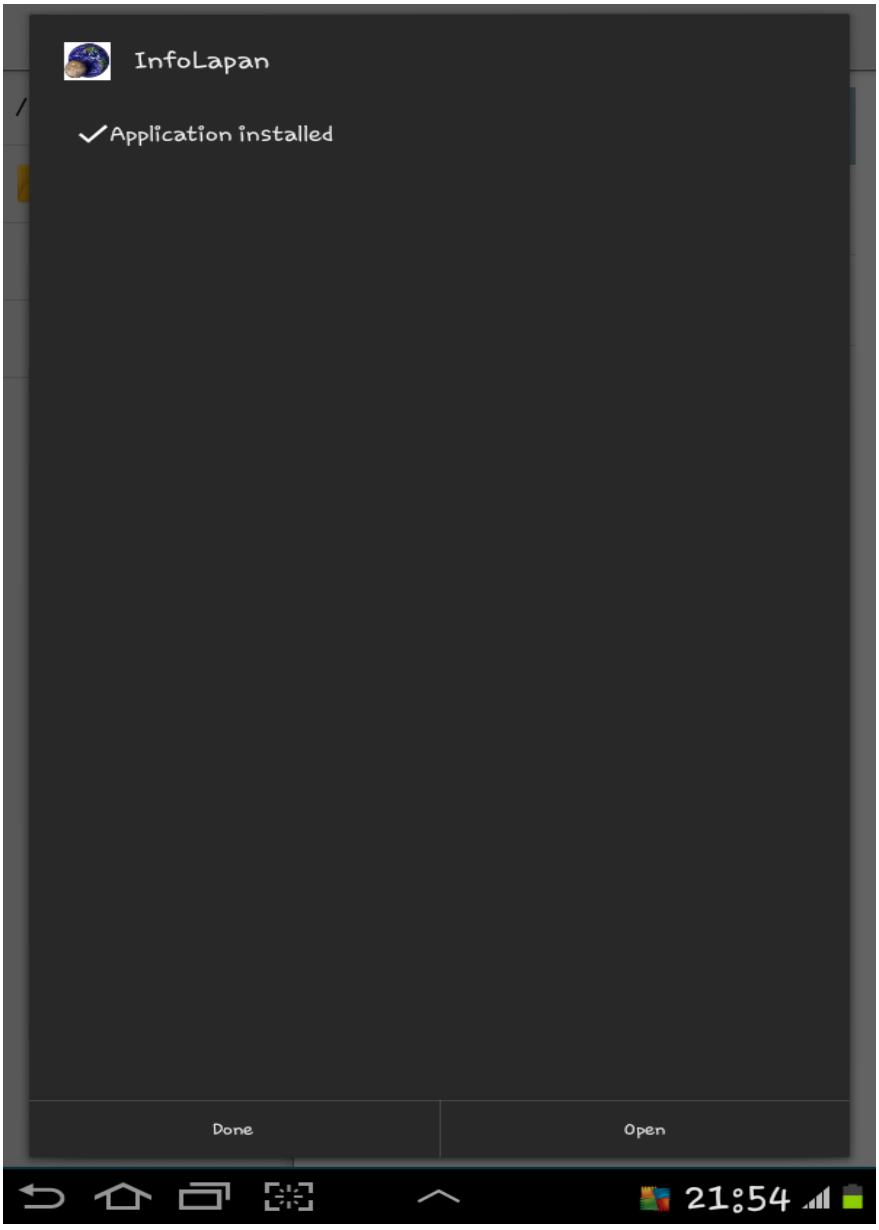
Gambar 5.16 Tampilan proses mengistall iontel.apk

4. Klik tombol intall kemudian akan muncul tampilan dibawah ini.



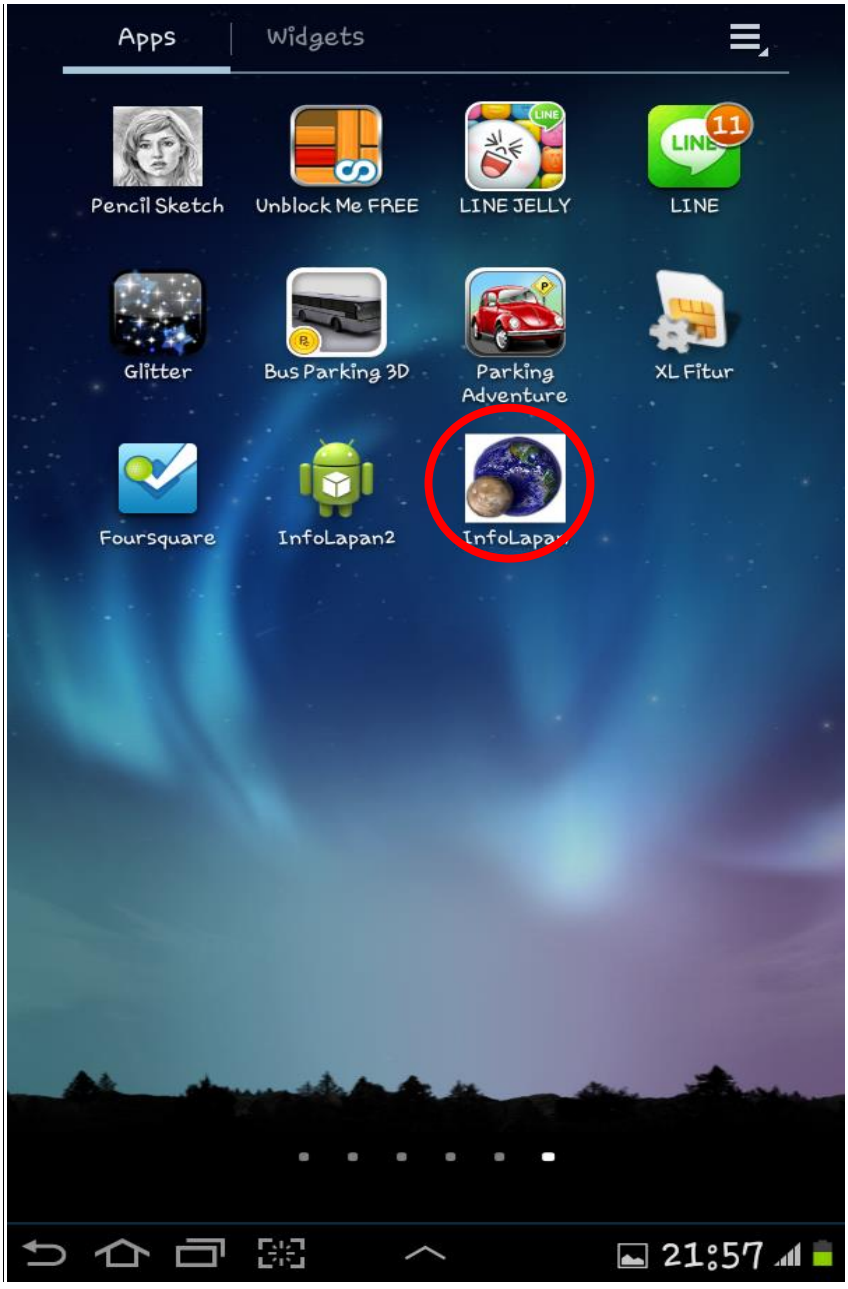
Gambar 5.17 Tampilan proses mengistall iontel.apk

5. **Setelah installasi selesai, *user* dapat memilih mau langsung membuka aplikasi atau tidak.**



Gambar 5.18 Tampilan setelah selesai menginstall iontel.apk

6. **Jika sudah terinstall akan ada didaftar menu di *tablet* seperti yang terlihat pada gambar dibawah ini (dilingkari warna merah)**



Gambar 5.19 Tampilan Aplikasi iontel ada *tablet* Android

5.2 Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu merepresentasikan kajian pokok dari spesifikasi, analis perancangan dan pengkodean dari perangkat lunak itu sendiri.

5.2.1 Rencana Pengujian

Pengujian Aplikasi Pengelolaan Informasi Berbasis Android di Lembaga Penerbangan dan Antariksa Nasional (LAPAN) ini menggunakan metode *Black Box* . Pengujian *Black Box* ini berfokus pada persyaratan fungsional dari aplikasi yang dibuat.

Tabel 5.1 Rencana Pengujian Pada Bagian Admin (Android)

Item Pengujian	Deskripsi	Jenis Pengujian
Menu Login	Melakukan pengujian apakah proses *Login* sudah benar dan sesuai dengan yang dirancang.	*Black Box*

Item Pengujian	Deskripsi	*Jenis Pengujian*
Menu Aktivitas Ionosfer	Melakukan pengujian apakah proses menampilkan data sudah benar dan sesuai dengan yang dirancang	*Black Box*
Menu Approval	Melakukan pengujian apakah proses mempublish atau mensyahkan informasi sudah benar dan sesuai dengan yang dirancang	*Black Box*
Menu Upload Schedule	Melakukan pengujian apakah proses Upload schedule sudah benar dan sesuai dengan yang dirancang	*Black Box*
Menu List Schedule	Melakukan pengujian apakah proses list schedule sudah benar dan sesuai dengan yang dirancang	*Black Box*
Menu About us	Melakukan pengujian apakah proses menu about us sudah benar dan sesuai dengan yang dirancang	*Black Box*
Menu LogOut	Melakukan pengujian apakah proses LogOut sudah benar dan sesuai dengan yang dirancang	*Black Box*

5.2.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian dibuat untuk mengetahui Aplikasi Pengelolaan Informasi Kegiatan Ionosfer dan Telekomunikasi Berbasis Android di Lembaga Penerbngan dan Antariksa Nasional (LAPAN) ini dapat berjalan dengan baik atau tidak.

a. Kasus dan Hasil Uji Pengujian Pada Bagian *Admin* (Android)

1. Pengujian *Login*

Berikut ini adalah tabel pengujian *Login* Aplikasi Informasi Kegiatan Ionosfer dan Telekomunikasi Berbasis Android di Lembaga Penerbngan dan Antariksa Nasional (LAPAN)

Tabel 5.2 Pengujian Login

Kasus dan Hasil Uji (Data Normal)		
Data Masukan	**Yang Diharapkan**	**Hasil Uji**
Username dan *Password* benar	Masuk ke halaman admin dengan hak akses administrator	[√] Berhasil [ ] Gagal
Kasus Dan Hasil Uji (Data Salah)		
Data Masukan	**Yang Diharapkan**	**Hasil Uji**
Username dan *Password* kosong	Login gagal, ada keterangan bahwa *username* dan/ *password* not match.	[√] Berhasil [ ] Gagal
Kasus Dan Hasil Uji (Data Salah)		
Data Masukan	**Yang Diharapkan**	**Hasil Uji**

Username atau *Password* salah	*Login* gagal, ada keterangan bahwa *username* atau *password* not match	[√] Berhasil [ ] Gagal

2. Pengujian Menu Profil

Berikut ini adalah tabel pengujian Menu Profil Aplikasi Pengelolaan Informasi Berbasis Android di Lembaga Penerbangan dan Antariksa Nasional (LAPAN)

Tabel 5.3 Pengujian Menu Tambah Data

Kasus dan Hasil Uji (Data Normal)		
Data Masukan	**Yang Diharapkan**	**Hasil Uji**
Masukkan data yang akan dirubah	- Muncul keterangan insert succses - Data tersimpan kedalam *database*	[√] Berhasil [ ] Gagal
Kasus Dan Hasil Uji (Data Salah)		
Data Masukan	**Yang Diharapkan**	**Hasil Uji**
Data tidak di input	- Muncul not match - Data tidak tersimpan kedalam *database*	[√] Berhasil [ ] Gagal

5.2.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* diatas dapat ditarik kesimpulan bahwa perangkat lunak berjalan dengan baik dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan perusahaan.