

Soal

1. Apakah pengertian dari aplikasi enterprise?
2. Jelaskan rancangan salah satu arsitektur sistem informasi terdistribusi dan berikan penjelasan latar belakang/alasan kenapa di desain arsitektur tersebut?
3. Jelaskan middleware pada arsitektur integrasi aplikasi enterprise ?
 - a. Kegunaan dan
 - b. kelebihan
4. Apa manfaat Rest pada aplikasi yang akan anda kembangkan pada kuliah enterprise? Jelaskan manfaat dari sisi client (Rest Client)
5. Jelaskan aturan XML yang harus dipenuhi jika akan digunakan pada Rest ? (minimal 3)

Jawab :

1. Sebuah software yang digunakan untuk memenuhi kebutuhan organisasi agar saling terhubung atau terintegrasi, seperti bisnis, perkantoran, sekolah, kelompok pengguna berbasis bunga, klub, amal, atau pemerintah.

2. Arsitektur SI terdistribusi : Peer-to-peer

Arsitektur peer-to-peer ada dimana masing-masing komputer memiliki kontrol terhadap *resource* (misalnya data, printer, atau CD-ROM), tetapi memungkinkan komputer lain menggunakan sumber (*resource*) tersebut. Ide mengenai konsep ini muncul kira-kira pada akhir dekade 1980-an, ketika jaringan komputer mulai menjadi salah satu barang wajib dalam perusahaan, baik itu perusahaan kecil maupun besar.

Jaringan *peer-to-peer* mulai banyak digemari ketika Microsoft merilis sistem operasi *Windows for Workgroups*, meski sebelumnya sistem operasi MS-DOS (atau IBM PC-DOS) dengan perangkat MS-NET (atau PC-NET) juga dapat digunakan untuk tujuan ini. Karakteristik utama jaringan tersebut adalah dalam jaringan ini tidak terdapat sebuah server pusat yang mengatur klien-klien, karena memang setiap komputer bertindak sebagai server untuk komputer klien lainnya. Sistem keamanan yang ditawarkan oleh metode ini terbilang lebih rendah dibandingkan dengan metode klien/server dan manajemen terhadapnya pun menjadi relatif lebih rumit.

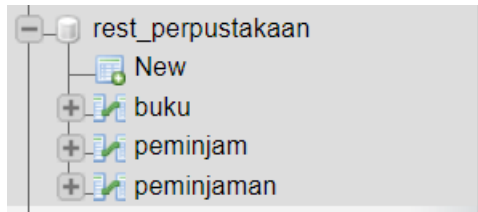
3. Middleware adalah software yang berada diantara lapisan aplikasi (application layer) dan lapisan data dari sebuah arsitektur layer-layer TCP/IP yang menghubungkan sebuah aplikasi atau rangkaian aplikasi.

Kegunaan dan kelebihan :

- a. Sebagai protokol yang menyediakan layanan pesan (messaging services) sehingga aplikasi-aplikasi yang berbeda-beda dapat berkomunikasi.
 - b. Sebagai Integrator yang memungkinkan beberapa proses dapat berjalan pada satu atau lebih mesin untuk saling berinteraksi pada suatu jaringan.
-
4. REST API membantu dalam menyediakan resources (data) untuk penggunaan selanjutnya. Bermanfaat untuk pengembangan aplikasi webi, dekstop, atau mobile yang menjadi client dari REST API untuk mengakses dan menampilkan resources yang sudah ada.
 5. Aturan,
 - a. Deklarasi dokumen XML harus diawali dengan deklarasi XML
 - b. Semua elemen XML harus mempunyai tag penutup “<...> </ ...> ”
 - c. Elemen-elemen XML harus berpasangan dengan benar “<a> ... ”dokumen XML harus mempunyai elemen utama (Root Element).

DataBase

1. Membuat database perpustakaan “rest_perpustakaan”



2. Table buku

	id_buku	judul	pengarang	tahun_terbit
<input type="checkbox"/>	B01	Rekayasa Sosial	Jalaluddin Rakhmat	2014
<input type="checkbox"/>	B02	Dilan 1990	Pldi Baiq	2015

Options: ☐ Pilih Semua, ☐ Dengan pilihan: Ubah, Salin, Hapus, Ekspor. ☐ Tampilkan semua, Jumlah baris: 25, Saring baris: Cari di tabel ini, Urut berdasarkan kunci: Tidak ada

3. Pada table peminjam

	id_peminjam	nama	alamat	telpn
<input type="checkbox"/>	001	beby novia	malang	8199980801
<input type="checkbox"/>	002	ayang evangga	kediri	8911178368

Options: ☐ Pilih Semua, ☐ Dengan pilihan: Ubah, Salin, Hapus, Ekspor. ☐ Tampilkan semua, Jumlah baris: 25, Saring baris: Cari di tabel ini, Urut berdasarkan kunci: Tidak ada

4. Pada table peminjaman

	id_peminjaman	id_user	tanggal_pinjam	tanggal_kembali	id_buku
<input type="checkbox"/>	ID01	001	2018-10-10	2019-10-11	B01
<input type="checkbox"/>	ID02	002	2018-10-14	2018-10-20	B02
<input type="checkbox"/>	ID03	002	2018-10-11	2018-10-15	B01

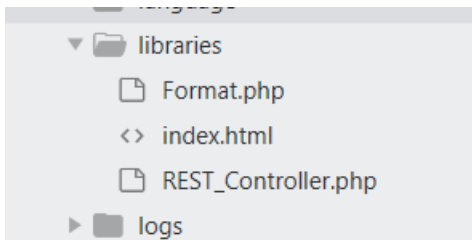
Options: ☐ Pilih Semua, ☐ Dengan pilihan: Ubah, Salin, Hapus, Ekspor. ☐ Tampilkan semua, Jumlah baris: 25, Saring baris: Cari di tabel ini, Urut berdasarkan kunci: Tidak ada

Mengatur Rest pada CodeIgniter

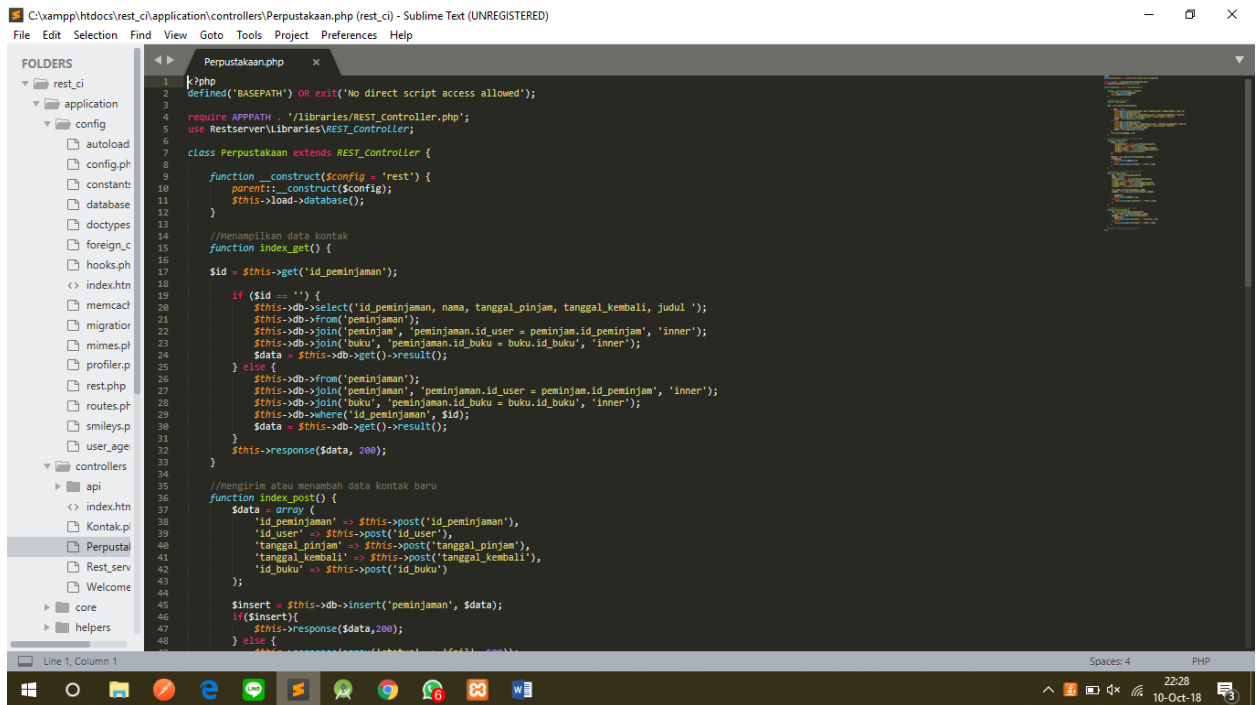
1. Mengatur config database pada codeigniter config/databae.php

```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 ▼ $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'rest_perpustakaan',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
```

2. Menambahkan Libraries Rest_controler pada application/libraries/rest_controler.php



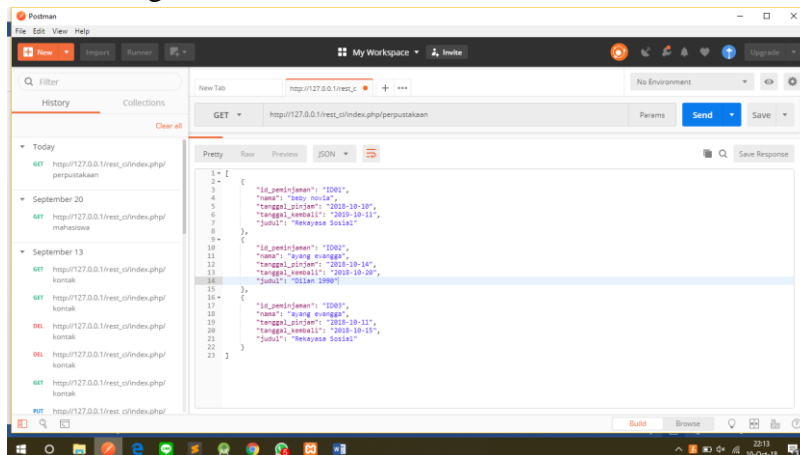
3. Membuat controller baru



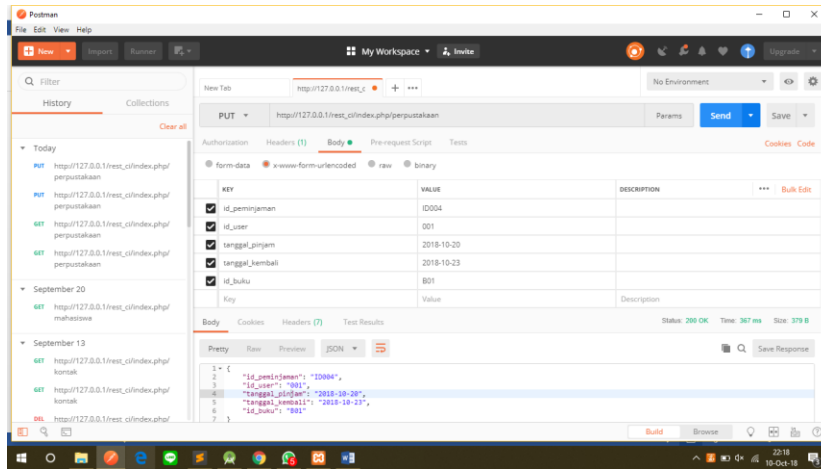
```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 require APPPATH . '/libraries/REST_Controller.php';
5 use Restserver\Libraries\REST_Controller;
6
7 class Perustakaan extends REST_Controller {
8
9     function __construct($config = 'rest') {
10         parent::__construct($config);
11         $this->load->database();
12     }
13
14     //Menampilkan data kontak
15     function index_get() {
16
17         $id = $this->get('id_peminjaman');
18
19         if ($id == '') {
20             $this->db->select('id_peminjaman, nama, tanggal_pinjam, tanggal_kembali, judul ');
21             $this->db->from('peminjaman');
22             $this->db->join('peminjaman', 'peminjaman.id_user = peminjam.id_peminjam', 'inner');
23             $this->db->join('buku', 'peminjaman.id_buku = buku.id_buku', 'inner');
24             $data = $this->db->get()->result();
25         } else {
26             $this->db->from('peminjaman');
27             $this->db->join('peminjaman', 'peminjaman.id_user = peminjam.id_peminjam', 'inner');
28             $this->db->join('buku', 'peminjaman.id_buku = buku.id_buku', 'inner');
29             $this->db->where('id_peminjaman', $id);
30             $data = $this->db->get()->result();
31         }
32         $this->response($data, 200);
33     }
34
35     //mengirim atau menambah data kontak baru
36     function index_post() {
37         $data = array (
38             'id_peminjaman' => $this->post('id_peminjaman'),
39             'id_user' => $this->post('id_user'),
40             'tanggal_pinjam' => $this->post('tanggal_pinjam'),
41             'tanggal_kembali' => $this->post('tanggal_kembali'),
42             'id_buku' => $this->post('id_buku')
43         );
44
45         $insert = $this->db->insert('peminjaman', $data);
46         if($insert){
47             $this->response($data,200);
48         } else {
49             $this->response($data,400);
50         }
51     }
52 }
```

Mencoba akses dengan Postman

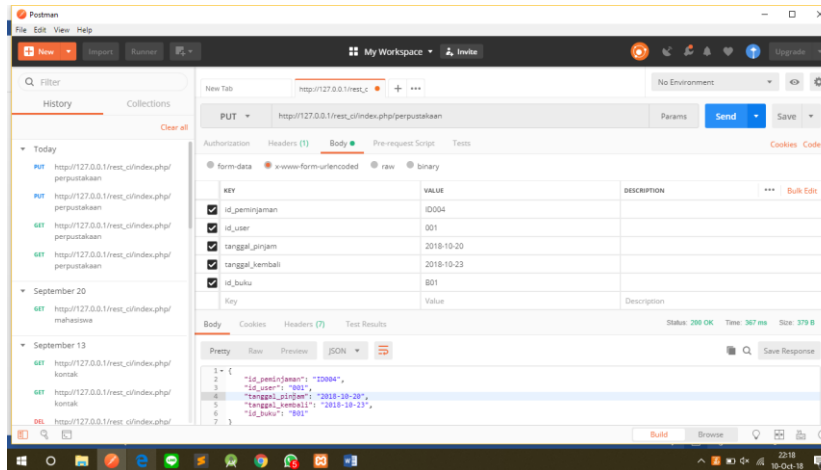
1. Percobaan get



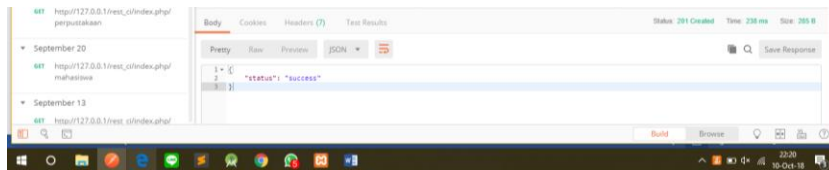
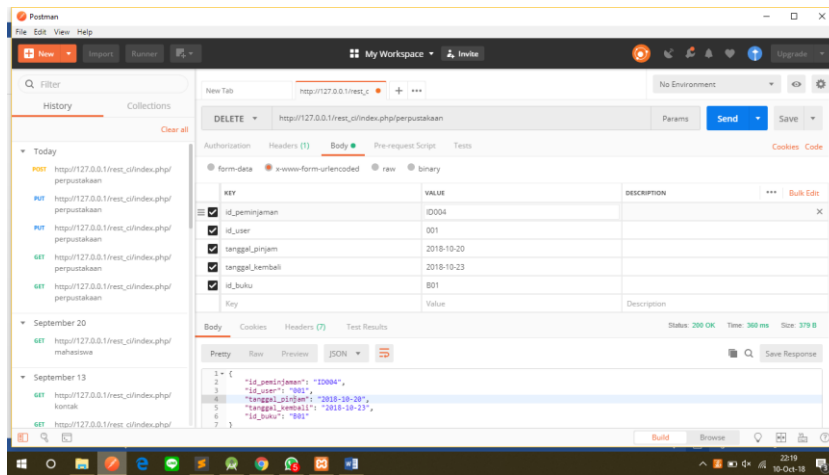
2. Percobaan put



3. Percobaan post



4. Percobaan delete

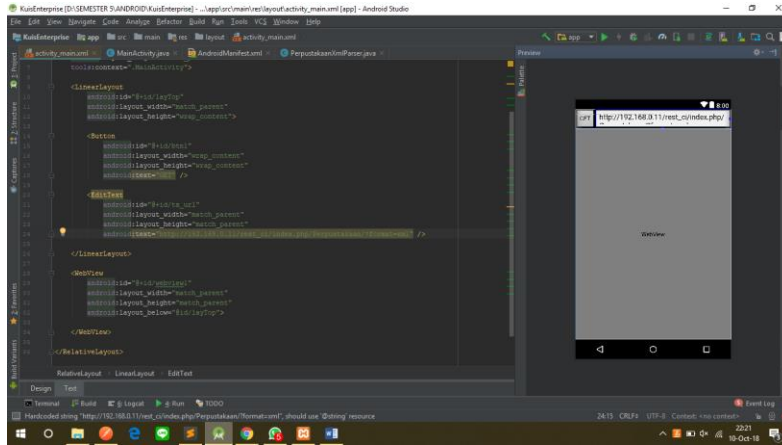


BEBY NOVIA PUTRI LOVENIDA

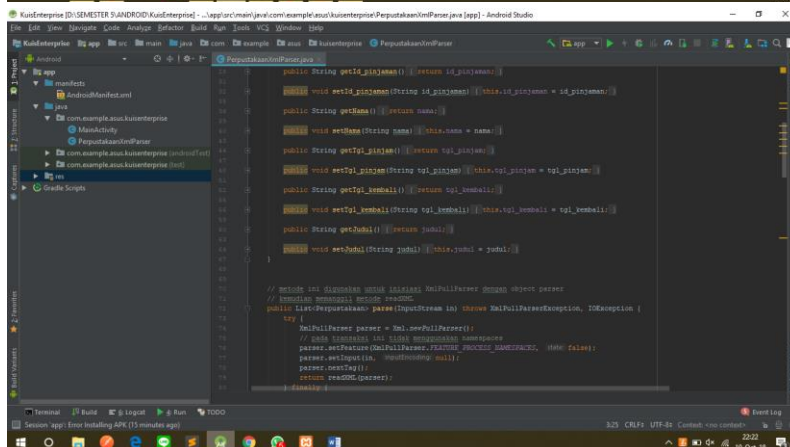
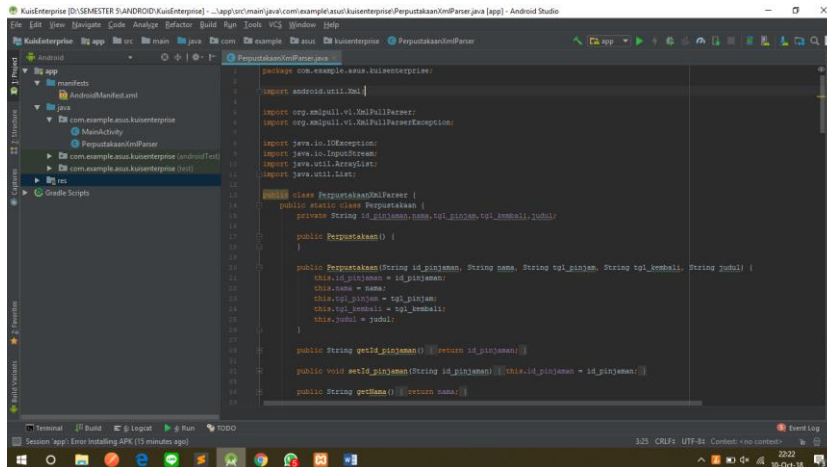
T13B- 16417200073

Android Studio RestClient

1. Tampilan utama pada activity_main.xml

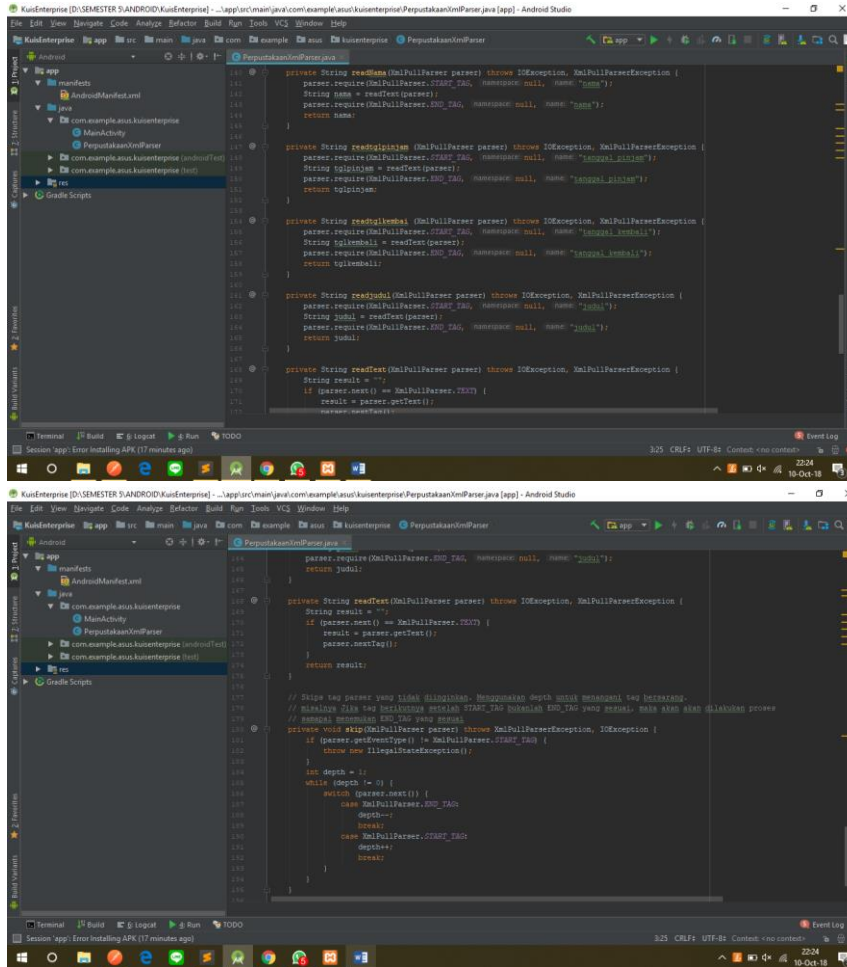


2. Coding pada perpustakaanxmlParser

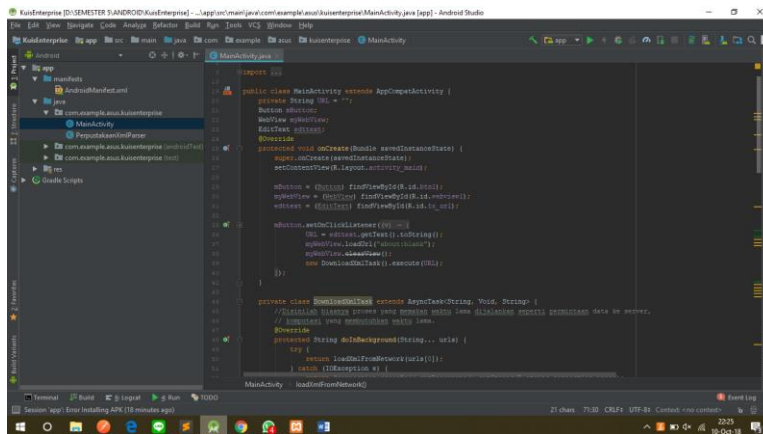


BEBY NOVIA PUTRI LOVENIDA
T13B- 16417200073

```
// Screenshot 1: Initial setup of PerputaanOnlineParser class
// Screenshot 2: Implementation of readData method, including the creation of a Perputaan object and the use of a while loop to read data from a server
// Screenshot 3: Implementation of readData method, including the use of a while loop to read data from a server and the use of a while loop to read data from a server
```



3. Codigan Pada MainActivity.java



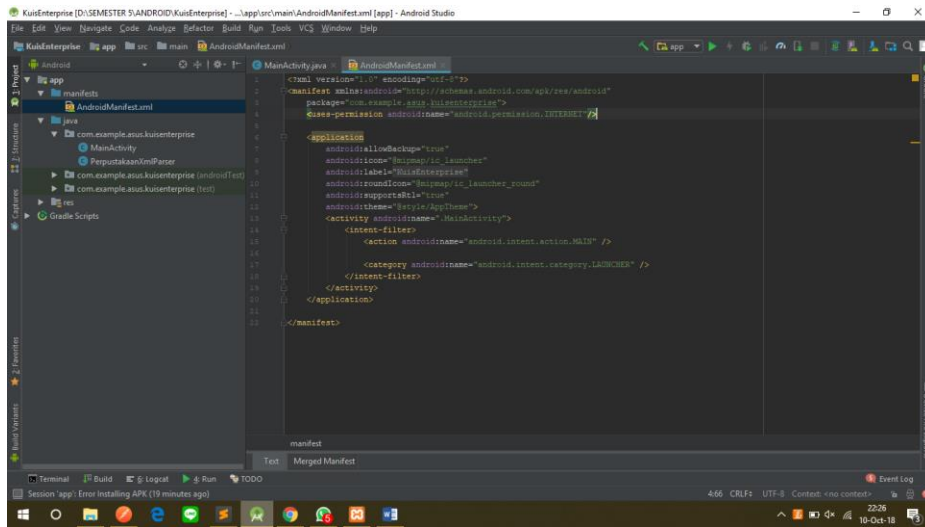
BEBY NOVIA PUTRI LOVENIDA
T13B- 16417200073

```
1: package com.example.asus.kuisenterprise;
2:
3: import android.os.AsyncTask;
4: import android.os.Bundle;
5: import android.view.View;
6: import android.view.View.OnClickListener;
7: import android.widget.Button;
8: import android.widget.TextView;
9: import androidx.appcompat.app.AppCompatActivity;
10: import java.io.IOException;
11: import java.io.InputStream;
12: import java.net.HttpURLConnection;
13: import java.net.URL;
14:
15: public class MainActivity extends AppCompatActivity {
16:
17:     private TextView textView;
18:     private Button button;
19:
20:     @Override
21:     protected void onCreate(Bundle savedInstanceState) {
22:         super.onCreate(savedInstanceState);
23:         setContentView(R.layout.activity_main);
24:         textView = findViewById(R.id.textView);
25:         button = findViewById(R.id.button);
26:         button.setOnClickListener(new OnClickListener() {
27:             @Override
28:             public void onClick(View view) {
29:                 new DownloadUrl().execute();
30:             }
31:         });
32:     }
33:
34:     private class DownloadUrl extends AsyncTask<String, Void, String> {
35:
36:         @Override
37:         protected String doInBackground(String... url) {
38:             try {
39:                 return loadFromNetwork(url[0]);
40:             } catch (IOException e) {
41:                 return "Connection error";
42:             }
43:         }
44:
45:         @Override
46:         protected void onPostExecute(String result) {
47:             textView.setText(result);
48:         }
49:     }
50:
51:     private String loadFromNetwork(String url) throws IOException {
52:         InputStream stream = null;
53:         HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
54:         conn.setReadTimeout(10000);
55:         conn.setConnectTimeout(10000);
56:         conn.setRequestMethod("GET");
57:         conn.setDoInput(true);
58:         conn.connect();
59:         stream = conn.getInputStream();
60:         return readFromStream(stream);
61:     }
62:
63:     private String readFromStream(InputStream stream) throws IOException {
64:         StringBuilder sb = new StringBuilder();
65:         byte[] buffer = new byte[1024];
66:         int len;
67:         while ((len = stream.read(buffer)) != -1) {
68:             sb.append(new String(buffer, 0, len));
69:         }
70:         return sb.toString();
71:     }
72: }
```

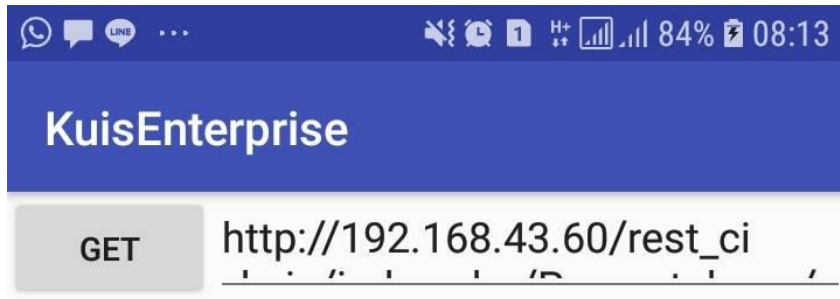
```
1: package com.example.asus.kuisenterprise;
2:
3: import android.os.AsyncTask;
4: import android.os.Bundle;
5: import android.view.View;
6: import android.view.View.OnClickListener;
7: import android.widget.Button;
8: import android.widget.TextView;
9: import androidx.appcompat.app.AppCompatActivity;
10: import java.io.IOException;
11: import java.io.InputStream;
12: import java.net.HttpURLConnection;
13: import java.net.URL;
14:
15: public class MainActivity extends AppCompatActivity {
16:
17:     private TextView textView;
18:     private Button button;
19:
20:     @Override
21:     protected void onCreate(Bundle savedInstanceState) {
22:         super.onCreate(savedInstanceState);
23:         setContentView(R.layout.activity_main);
24:         textView = findViewById(R.id.textView);
25:         button = findViewById(R.id.button);
26:         button.setOnClickListener(new OnClickListener() {
27:             @Override
28:             public void onClick(View view) {
29:                 new DownloadUrl().execute();
30:             }
31:         });
32:     }
33:
34:     private class DownloadUrl extends AsyncTask<String, Void, String> {
35:
36:         @Override
37:         protected String doInBackground(String... url) {
38:             try {
39:                 return loadFromNetwork(url[0]);
40:             } catch (IOException e) {
41:                 return "Connection error";
42:             }
43:         }
44:
45:         @Override
46:         protected void onPostExecute(String result) {
47:             textView.setText(result);
48:         }
49:     }
50:
51:     private String loadFromNetwork(String url) throws IOException {
52:         InputStream stream = null;
53:         HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
54:         conn.setReadTimeout(10000);
55:         conn.setConnectTimeout(10000);
56:         conn.setRequestMethod("GET");
57:         conn.setDoInput(true);
58:         conn.connect();
59:         stream = conn.getInputStream();
60:         return readFromStream(stream);
61:     }
62:
63:     private String readFromStream(InputStream stream) throws IOException {
64:         StringBuilder sb = new StringBuilder();
65:         byte[] buffer = new byte[1024];
66:         int len;
67:         while ((len = stream.read(buffer)) != -1) {
68:             sb.append(new String(buffer, 0, len));
69:         }
70:         return sb.toString();
71:     }
72: }
```

```
1: package com.example.asus.kuisenterprise;
2:
3: import android.os.AsyncTask;
4: import android.os.Bundle;
5: import android.view.View;
6: import android.view.View.OnClickListener;
7: import android.widget.Button;
8: import android.widget.TextView;
9: import androidx.appcompat.app.AppCompatActivity;
10: import java.io.IOException;
11: import java.io.InputStream;
12: import java.net.HttpURLConnection;
13: import java.net.URL;
14:
15: public class MainActivity extends AppCompatActivity {
16:
17:     private TextView textView;
18:     private Button button;
19:
20:     @Override
21:     protected void onCreate(Bundle savedInstanceState) {
22:         super.onCreate(savedInstanceState);
23:         setContentView(R.layout.activity_main);
24:         textView = findViewById(R.id.textView);
25:         button = findViewById(R.id.button);
26:         button.setOnClickListener(new OnClickListener() {
27:             @Override
28:             public void onClick(View view) {
29:                 new DownloadUrl().execute();
30:             }
31:         });
32:     }
33:
34:     private class DownloadUrl extends AsyncTask<String, Void, String> {
35:
36:         @Override
37:         protected String doInBackground(String... url) {
38:             try {
39:                 return loadFromNetwork(url[0]);
40:             } catch (IOException e) {
41:                 return "Connection error";
42:             }
43:         }
44:
45:         @Override
46:         protected void onPostExecute(String result) {
47:             textView.setText(result);
48:         }
49:     }
50:
51:     private String loadFromNetwork(String url) throws IOException {
52:         InputStream stream = null;
53:         HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
54:         conn.setReadTimeout(10000);
55:         conn.setConnectTimeout(10000);
56:         conn.setRequestMethod("GET");
57:         conn.setDoInput(true);
58:         conn.connect();
59:         stream = conn.getInputStream();
60:         return readFromStream(stream);
61:     }
62:
63:     private String readFromStream(InputStream stream) throws IOException {
64:         StringBuilder sb = new StringBuilder();
65:         byte[] buffer = new byte[1024];
66:         int len;
67:         while ((len = stream.read(buffer)) != -1) {
68:             sb.append(new String(buffer, 0, len));
69:         }
70:         return sb.toString();
71:     }
72: }
```

4. Codigan Pada Manifest



BEBY NOVIA PUTRI LOVENIDA
TI3B- 16417200073



Id :**ID01**

Nama : beby novia
pinjam :2018-10-10
kembali :2019-10-11
Judul :Rekayasa Sosial

Id :**ID02**

Nama : ayang evangga
pinjam :2018-10-14
kembali :2018-10-20
Judul :Dilan 1990

Id :**ID03**

Nama : ayang evangga
pinjam :2018-10-11
kembali :2018-10-15
Judul :Rekayasa Sosial