

LISTING PROGRAM APLIKASI ANDROID

AboutFragment.java

```

package tia.restodj;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import tia.restodj.R;
public class AboutFragment extends Fragment {
    public AboutFragment() {}
    String content;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        // -- inflate the layout for this fragment
        View myInflatedView =
inflater.inflate(R.layout.fragment_about, container, false);
        WebView wv = (WebView)
myInflatedView.findViewById(R.id.webViewMain);
        content = "file:///android_asset/index.html";
        wv.loadUrl(content);
        return myInflatedView;
    }
}

```

AppController.java

```

package tia.restodj;
import android.app.Application;
import android.text.TextUtils;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;
public class AppController extends Application {
    public static final String TAG =
AppController.class.getSimpleName();
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    LruBitmapCache mLruBitmapCache;
    private static AppController mInstance;
    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }
    public static synchronized AppController getInstance() {
        return mInstance;
    }
    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue =
Volley.newRequestQueue(getApplicationContext());
        }
    }
}

```

```

        return mRequestQueue;
    }
    public ImageLoader getImageLoader() {
        getRequestQueue();
        if (mImageLoader == null) {
            getLruBitmapCache();
            mImageLoader = new
ImageLoader(this.mRequestQueue, mLruBitmapCache);
        }
        return this.mImageLoader;
    }
    public LruBitmapCache getLruBitmapCache() {
        if (mLruBitmapCache == null)
            mLruBitmapCache = new LruBitmapCache();
        return this.mLruBitmapCache;
    }
    public <T> void addToRequestQueue(Request<T> req, String
tag) {
        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }
    public <T> void addToRequestQueue(Request<T> req) {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }
    public void cancelPendingRequests(Object tag) {
        if (mRequestQueue != null) {
            mRequestQueue.cancelAll(tag);
        }
    }
}

```

AppStatus.java

```

package tia.restodj;
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.util.Log;
public class AppStatus {
    private static AppStatus instance = new AppStatus();
    static Context context;
    ConnectivityManager connectivityManager;
    NetworkInfo wifiInfo, mobileInfo;
    boolean connected = false;
    public static AppStatus getInstance(Context ctx) {
        context = ctx;
        return instance;
    }
    public boolean isOnline(Context con) {
        try {
            connectivityManager = (ConnectivityManager) con
.getSystemService(Context.CONNECTIVITY_SERVICE);
            NetworkInfo networkInfo =
connectivityManager.getActiveNetworkInfo();
            connected = networkInfo != null &&
networkInfo.isAvailable() &&

```

```

        networkInfo.isConnected();
    return connected;
    } catch (Exception e) {
        System.out.println("CheckConnectivity Exception: " +
e.getMessage());
        Log.v("connectivity", e.toString());
    }
    return connected;
}
private static NetworkInfo networkInfo;
public static boolean isConnected(Context context) {
    ConnectivityManager cm =
(ConnectivityManager) context.getSystemService(Context.CONNECTIVITY
_SERVICE);
    try{
        networkInfo =
cm.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
    } catch (Exception e) {
        e.printStackTrace();
    }

    if (networkInfo != null
        && networkInfo.isAvailable()
        && networkInfo.isConnected()) {
        return true;
    }
    networkInfo =
cm.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
    if (networkInfo != null
        && networkInfo.isAvailable()
        && networkInfo.isConnected()) {
        return true;
    }
    return false;
}
}

```

Gallery Activity.java

```

package tia.restodj;
import android.content.Context;
import android.util.AttributeSet;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.widget.Gallery;
@SuppressWarnings("deprecation")
public class Gallery_Activity extends Gallery{
    private boolean stuck = false;
    public Gallery_Activity(Context context, AttributeSet attrs,
int defStyle) {
        super(context, attrs, defStyle);
    }
    public Gallery_Activity(Context context, AttributeSet attrs) {
        super(context, attrs);
    }
    public Gallery_Activity(Context context) {
        super(context);
    }
}

```

```

    }
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        return stuck || super.onTouchEvent(event);
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        switch (keyCode) {
            case KeyEvent.KEYCODE_DPAD_LEFT:
            case KeyEvent.KEYCODE_DPAD_RIGHT:
                return stuck || super.onKeyDown(keyCode, event);
        }
        return super.onKeyDown(keyCode, event);
    }
    public void setScrollingEnabled(boolean enabled) {
        stuck = !enabled;
    }
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX, float velocityY){
        return super.onFling(e1, e2, 0, velocityY);
    }
}

```

HomeFragment.java

```

package tia.restodj;
import android.app.Fragment;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.LinearLayout;
import android.widget.TextView;
import tia.restodj.R;
public class HomeFragment extends Fragment {
    public HomeFragment(){}
    LinearLayout count_layout;
    TextView page_text[];
    int count = 0;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState){
        View myInflatedView =
inflater.inflate(R.layout.fragment_home, container, false);
        Gallery_Activity mGallery = (Gallery_Activity)
myInflatedView.findViewById(R.id.mygallery01);
        mGallery.setAdapter(new ImageAdapter(getActivity()));
        mGallery.setScrollingEnabled(true);

        count_layout = (LinearLayout)
myInflatedView.findViewById(R.id.image_count);
        count = mGallery.getAdapter().getCount();
    }
}

```

```

        System.out.println("Gallery Image Count=====>>>" +
count);
        page_text = new TextView[count];
        for (int i = 0; i < count; i++)
        {
            page_text[i] = new TextView(getActivity());
            page_text[i].setText(".");
            page_text[i].setTextSize(45);
            page_text[i].setTypeface(null, Typeface.BOLD);
page_text[i].setTextColor(android.graphics.Color.GRAY);
            count_layout.addView(page_text[i]);
        }
        mGallery.setOnItemClickListener(new
OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
view,
                                int pos, long id) {
                System.out.println("Item Selected
Position=====>>>" + pos);
                for (int i = 0; i < count; i++) {
                    page_text[i]
.setTextColor(android.graphics.Color.GRAY);
                }
                page_text[pos]
.setTextColor(getResources().getColor(R.color.colorBacround));
            }
            @Override
            public void onNothingSelected(AdapterView<?> arg0) {
            }
        });
        return myInflatedView;
    }
}

```

ImageAdapter.java

```

package tia.restodj;
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import tia.restodj.R;
public class ImageAdapter extends BaseAdapter
{
    Context mycontext = null;
    int galitembg = 0;
    public int[] IMAGE_IDS = {
        R.drawable.slide_one,
        R.drawable.slide_two,
        R.drawable.slide_three
    };
    public ImageAdapter(Context c)
    {
        mycontext = c;
    }
}

```

```

        TypedArray typArray =
mycontext.obtainStyledAttributes(R.styleable.GalleryTheme);
        galitembg =
typArray.getResourceId(R.styleable.GalleryTheme_android_galleryIte
mBackground, 0);
        typArray.recycle();
    }
    @Override
    public int getCount()
    {
        return IMAGE_IDS.length;
    }
    @Override
    public Object getItem(int position)
    {
        return position;
    }
    @Override
    public long getItemId(int position)
    {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup
parent)
    {
        ImageView imageview = new ImageView(mycontext);
        imageview.setImageResource(IMAGE_IDS[position]);
        imageview.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
        return imageview;
    }
}

```

JSONParser.java

```

package tia.restodj;
import android.util.Log;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;
public class JSONParser {
    static InputStream is = null;

```

```

static JSONObject jsonObj = null;
static String json = "";
public JSONParser() {
}
public JSONObject getJSONFromUrl(String url) {
    try {
        DefaultHttpClient httpClient = new
DefaultHttpClient();
        HttpPost httpPost = new HttpPost(url);
        HttpResponse httpResponse =
httpClient.execute(httpPost);
        HttpEntity httpEntity =
httpResponse.getEntity();
        is = httpEntity.getContent();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        BufferedReader reader = new BufferedReader(new
InputStreamReader(
            is, "iso-8859-1"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting result "
+ e.toString());
    }
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " +
e.toString());
    }
    return jsonObj;
}
public JSONObject makeHttpRequest(String url, String method,
List<NameValuePair> params) {
    try {
        if (method == "POST") {
            DefaultHttpClient httpClient = new
DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new
UrlEncodedFormEntity(params));
            HttpResponse httpResponse =
httpClient.execute(httpPost);

```

```

        HttpEntity httpEntity =
httpResponse.getEntity();
        is = httpEntity.getContent();
    } else if (method == "GET") {
        DefaultHttpClient httpClient = new
DefaultHttpClient();
        String paramString =
URLEncodedUtils.format(params, "utf-8");
        url += "?" + paramString;
        HttpGet httpGet = new HttpGet(url);
        HttpResponse httpResponse =
httpClient.execute(httpGet);
        HttpEntity httpEntity =
httpResponse.getEntity();
        is = httpEntity.getContent();
    }
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
try {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result "
+ e.toString());
}
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " +
e.toString());
}
return jsonObj;
}
}

```

LruBitmapCache.java

```

package tia.restodj;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;
import com.android.volley.toolbox.ImageLoader.ImageCache;

public class LruBitmapCache extends LruCache<String, Bitmap>
implements

```



```

        ImageCache {
            public static int getDefaultLruCacheSize() {
                final int maxMemory = (int)
(Runtime.getRuntime().maxMemory() / 1024);
                final int cacheSize = maxMemory / 8;
                return cacheSize;
            }
            public LruBitmapCache() {
                this(getDefaultLruCacheSize());
            }
            public LruBitmapCache(int sizeInKiloBytes) {
                super(sizeInKiloBytes);
            }
            @Override
            protected int sizeof(String key, Bitmap value) {
                return value.getRowBytes() * value.getHeight() / 1024;
            }

            @Override
            public Bitmap getBitmap(String url) {
                return get(url);
            }
            @Override
            public void putBitmap(String url, Bitmap bitmap) {
                put(url, bitmap);
            }
        }
    }

```

MainActivity.java

```

package tia.restodj;
import android.app.ActionBar;
import android.app.ActionBar.Tab;
import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.res.Configuration;
import android.content.res.TypedArray;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.support.v4.app.ActionBarDrawerToggle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.view.ViewPager;
import android.support.v4.widget.DrawerLayout;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ImageView;
import android.widget.ListView;
import tia.restodj.R;
import java.util.ArrayList;
public class MainActivity extends FragmentActivity implements
        ActionBar.TabListener {
    private ViewPager viewPager;

```

```

private DrawerLayout mDrawerLayout;
private ListView mDrawerList;
private ActionBarDrawerToggle mDrawerToggle;
private CharSequence mDrawerTitle;
private CharSequence mTitle;
private String[] navMenuTitles;
private TypedArray navMenuIcons;
private ArrayList<NavDrawerItem> navDrawerItems;
private NavDrawerListAdapter adapter;
ImageView Pic;
RoundedImageView RoundedImageView;
@Override
@SuppressWarnings("ResourceType")
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mTitle = mDrawerTitle = getTitle();
    navMenuTitles =
getResources().getStringArray(R.array.nav_drawer_items);
    navMenuIcons = getResources()
.obtainTypedArray(R.array.nav_drawer_icons);
    mDrawerLayout = (DrawerLayout)
findViewById(R.id.drawer_layout);
    mDrawerList = (ListView)
findViewById(R.id.list_slidermenu);
    View header =
getLayoutInflater().inflate(R.layout.nav_header_main, null,
false);
    mDrawerList.addHeaderView(header, null, false);
    Pic = (ImageView) findViewById(R.id.profile_header);
    Bitmap bm =
BitmapFactory.decodeResource(getResources(), R.drawable.fd);
    RoundedImageView = new RoundedImageView(bm);
    Pic.setImageDrawable(RoundedImageView);
    navDrawerItems = new ArrayList<NavDrawerItem>();
    navDrawerItems.add(new NavDrawerItem(navMenuTitles[0],
navMenuIcons.getResourceId(0, -1)));
    navDrawerItems.add(new NavDrawerItem(navMenuTitles[1],
navMenuIcons.getResourceId(1, -1)));
    navDrawerItems.add(new NavDrawerItem(navMenuTitles[2],
navMenuIcons.getResourceId(2, -1)));
    navDrawerItems.add(new NavDrawerItem(navMenuTitles[3],
navMenuIcons.getResourceId(3, -1)));
    navMenuIcons.recycle();
    mDrawerList.setOnItemClickListener(new
SlideMenuClickListener());
    adapter = new
NavDrawerListAdapter(getApplicationContext(),
navDrawerItems);
    mDrawerList.setAdapter(adapter);
    getActionBar().setDisplayHomeAsUpEnabled(true);
    getActionBar().setHomeButtonEnabled(true);
    mDrawerToggle = new ActionBarDrawerToggle(this,
mDrawerLayout,
R.drawable.ic_drawer,
R.string.app_name,

```

```

        R.string.app_name
    ) {
        public void onDrawerClosed(View view) {
            getActionBar().setTitle(mTitle);
            invalidateOptionsMenu();
        }
        public void onDrawerOpened(View drawerView) {
            getActionBar().setTitle(mDrawerTitle);
            invalidateOptionsMenu();
        }
    };
    mDrawerLayout.setDrawerListener(mDrawerToggle);
    if (savedInstanceState == null) {
        displayView(0);
    }
}
/**
 * Slide menu item click listener
 * */
private class SlideMenuClickListener implements
    ListView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
int position,
        long id) {
        displayView(position-1);
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return false;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }
    switch (item.getItemId()) {
    case R.id.action_favorite:
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
}
/**
 * Called when invalidateOptionsMenu() is triggered
 * */
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    boolean drawerOpen =
mDrawerLayout.isDrawerOpen(mDrawerList);
    menu.findItem(R.id.action_favorite).setVisible(!drawerOpen);
    return super.onPrepareOptionsMenu(menu);
}
}
/**

```

```

    * Diplaying fragment view for selected nav drawer list item
    * */
private void displayView(int position) {
    Fragment fragment = null;
    switch (position) {
        case 0:
            fragment = new HomeFragment();
            break;
        case 1:
            fragment = new TAMenuOrderFragment();
            break;
        case 2:
            fragment = new TOOrderFragment();
            break;
        case 3:
            fragment = new AboutFragment();
            break;
        default:
            break;
    }
    if (fragment != null) {
        FragmentManager fragmentManager =
getFragmentManager();
        fragmentManager.beginTransaction()
            .replace(R.id.frame_container,
fragment).commit();
        mDrawerList.setItemChecked(position+1, true);
        mDrawerList.setSelection(position);
        setTitle(navMenuTitles[position]);
        mDrawerLayout.closeDrawer(mDrawerList);
    } else {
        Log.e("MainActivity", "Error in creating fragment");
    }
}
@Override
public void setTitle(CharSequence title) {
    mTitle = title;
    getActionBar().setTitle(mTitle);
}
/**
 * When using the ActionBarDrawerToggle, you must call it
during
 * onCreate() and onConfigurationChanged()...
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mDrawerToggle.syncState();
}
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    mDrawerToggle.onConfigurationChanged(newConfig);
}
@Override
public void onTabReselected(Tab tab, FragmentTransaction ft) {

```

```

    }
    @Override
    public void onTabSelected(Tab tab, FragmentTransaction ft) {
        viewPager.setCurrentItem(tab.getPosition());
    }
    @Override
    public void onTabUnselected(Tab tab, FragmentTransaction ft) {
    }
}

```

NavDrawerItem.java

```

package tia.restodj;
public class NavDrawerItem {
    private String title;
    private int icon;
    private String count = "0";
    private boolean isCounterVisible = false;
    public NavDrawerItem(){}
    public NavDrawerItem(String title, int icon){
        this.title = title;
        this.icon = icon;
    }
    public NavDrawerItem(String title, int icon, boolean
isCounterVisible, String count){
        this.title = title;
        this.icon = icon;
        this.isCounterVisible = isCounterVisible;
        this.count = count;
    }
    public String getTitle(){
        return this.title;
    }
    public int getIcon(){
        return this.icon;
    }
    public String getCount(){
        return this.count;
    }
    public boolean getCounterVisibility(){
        return this.isCounterVisible;
    }
    public void setTitle(String title){
        this.title = title;
    }
    public void setIcon(int icon){
        this.icon = icon;
    }
    public void setCount(String count){
        this.count = count;
    }
    public void setCounterVisibility(boolean isCounterVisible){
        this.isCounterVisible = isCounterVisible;
    }
}

```

NavDrawerListAdapter.java

```

package tia.restodj;
import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import tia.restodj.R;
import java.util.ArrayList;
public class NavDrawerListAdapter extends BaseAdapter {
    private Context context;
    private ArrayList<NavDrawerItem> navDrawerItems;
    public NavDrawerListAdapter(Context context,
ArrayList<NavDrawerItem> navDrawerItems){
        this.context = context;
        this.navDrawerItems = navDrawerItems;
    }
    @Override
    public int getCount() {
        return navDrawerItems.size();
    }
    @Override
    public Object getItem(int position) {
        return navDrawerItems.get(position);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        if (convertView == null) {
            LayoutInflater mInflater = (LayoutInflater)
context.getSystemService(Activity.LAYOUT_INFLATER_SERVICE);
            convertView =
mInflater.inflate(R.layout.drawer_list_item, null);
        }
        ImageView imgIcon = (ImageView)
convertView.findViewById(R.id.icon);
        TextView txtTitle = (TextView)
convertView.findViewById(R.id.title);
        TextView txtCount = (TextView)
convertView.findViewById(R.id.counter);
        imgIcon.setImageResource(navDrawerItems.get(position).getIcon());
        txtTitle.setText(navDrawerItems.get(position).getTitle());
        if(navDrawerItems.get(position).getCounterVisibility()){
            txtCount.setText(navDrawerItems.get(position).getCount());
        }else{
            txtCount.setVisibility(View.GONE);
        }
        return convertView;
    }
}

```

RoundedImageView.java

```

package tia.restodj;
import android.graphics.Bitmap;
import android.graphics.BitmapShader;
import android.graphics.Canvas;
import android.graphics.ColorFilter;
import android.graphics.Paint;
import android.graphics.PixelFormat;
import android.graphics.Rect;
import android.graphics.RectF;
import android.graphics.Shader;
import android.graphics.drawable.Drawable;
public class RoundedImageView extends Drawable {
    private final Bitmap mBitmap;
    private final Paint mPaint;
    private final RectF mRectF;
    private final int mBitmapWidth;
    private final int mBitmapHeight;
    public RoundedImageView(Bitmap bitmap) {
        mBitmap = bitmap;
        mRectF = new RectF();
        mPaint = new Paint();
        mPaint.setAntiAlias(true);
        mPaint.setDither(true);
        final BitmapShader shader = new BitmapShader(bitmap,
Shader.TileMode.CLAMP, Shader.TileMode.CLAMP);
        mPaint.setShader(shader);
        mBitmapWidth = mBitmap.getWidth();
        mBitmapHeight = mBitmap.getHeight();
    }
    @Override
    public void draw(Canvas canvas) {
        canvas.drawOval(mRectF, mPaint);
    }
    @Override
    protected void onBoundsChange(Rect bounds) {
        super.onBoundsChange(bounds);
        mRectF.set(bounds);
    }
    @Override
    public void setAlpha(int alpha) {
        if (mPaint.getAlpha() != alpha) {
            mPaint.setAlpha(alpha);
            invalidateSelf();
        }
    }
    @Override
    public void setColorFilter(ColorFilter cf) {
        mPaint.setColorFilter(cf);
    }
    @Override
    public int getOpacity() {
        return PixelFormat.TRANSLUCENT;
    }
    @Override

```

```

public int getIntrinsicWidth() {
    return mBitmapWidth;
}
@Override
public int getIntrinsicHeight() {
    return mBitmapHeight;
}
public void setAntiAlias(boolean aa) {
    mPaint.setAntiAlias(aa);
    invalidateSelf();
}
@Override
public void setFilterBitmap(boolean filter) {
    mPaint.setFilterBitmap(filter);
    invalidateSelf();
}
@Override
public void setDither(boolean dither) {
    mPaint.setDither(dither);
    invalidateSelf();
}
public Bitmap getBitmap() {
    return mBitmap;
}
}

```

SplashScreen.java

```

package tia.restodj;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import tia.restodj.R;
public class SplashScreen extends Activity {
    private static String TAG = SplashScreen.class.getName();
    private static long SLEEP_TIME = 1;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        /*
this.requestWindowFeature(Window.FEATURE_NO_TITLE);
this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
// Remove the notification bar
        */
        setContentView(R.layout.splash);
        IntentLauncher launcher = new IntentLauncher();
        launcher.start();
    }
    private class IntentLauncher extends Thread {
        public void run() {
            try{
                Thread.sleep(SLEEP_TIME*500);
            } catch (Exception e){
                Log.e(TAG, e.getMessage());
            }
        }
    }
}

```



```

        Intent intent = new Intent(SplashScreen.this,
MainActivity.class);
        SplashScreen.this.startActivity(intent);
        SplashScreen.this.finish();
    }
}
}

```

TACheckOut.java

```

package tia.restodj;
import android.app.ActionBar;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.PorterDuff;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
public class TACheckOut extends Activity {
    private ActionBar actionBar;
    private ListView lv;
    public static ArrayList<TACheckOutData> modelArrayList;
    private TACheckOutAdapter TACheckOutAdapter;
    private Button btnnext;
    String dipilih;
    TextView txt_tabel;
    EditText nama_pelanggan;
    ProgressDialog pDialog;
    NumberFormat rupiahFormat;
    private static final String TAG =
MainActivity.class.getSimpleName();
    private static final String REGISTER_URL =
"http://dapurjawarestosa.16mb.com/api/order";
    public static final String NAMA_PEMESAN = "nama_pemesan";
    public static final String ID_MEJA = "id_meja";
    public static final String ID_MENU = "id_menu";

```

```

public static final String QTY = "qty";
public interface Callbacks {
    public void onBackPressed();
}
@Override
public void onBackPressed() {
    super.onBackPressed();
    Intent intent = new Intent(TACheckOut.this,
MainActivity.class);
    startActivity(intent);
    finish();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.check_out);
    actionBar = getActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setTitle("KEMBALI");
    lv = (ListView) findViewById(R.id.lv);
    btnnext = (Button) findViewById(R.id.next);
    nama_pelanggan = (EditText)
findViewById(R.id.nama_pelanggan);
nama_pelanggan.setBackground().setColorFilter(getResources().getCo
lor(R.color.colorBacround), PorterDuff.Mode.ADD.SRC_IN);
modelArrayList = getModel();
TACheckOutAdapter = new TACheckOutAdapter(this);
lv.setAdapter(TACheckOutAdapter);
rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
btnnext.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
if(nama_pelanggan.getText().toString().length()==0)
    {
        nama_pelanggan.setError("Tolong Isi Nama
Anda");
        return;
    }
    else
    {
        String nama = "";
        int sum=0;
        for (TACheckOutData hold : modelArrayList) {
            /*
            int arr[]={19,20,20,20,20};
            int sum=0;

            for(int i:arr) {
                sum += i;
            }
            */
            sum
+=Integer.parseInt(hold.getPrice().toString());

```

```

                String hargatambahqty =
rupiahFormat.format(Double.parseDouble(hold.getPrice()));
                nama += "\n" + hold.getNama() +
                    " : " + hold.getNumber() +
                    " : " + hargatambahqty;
            }
            String pelanggan = "A.N : " +
nama_pelanggan.getText().toString() + "\n" ;
            String totalhargo =
rupiahFormat.format(Double.parseDouble(String.valueOf(sum)));
            String fakesum = "\n" + "\n" + "Total : " +
totalhargo ;

            if (!nama.isEmpty()) {
                dipilih = pelanggan + nama + fakesum;
            } else {
                dipilih = "ANDA BELUM MEMILIH MENU.";
            }
            formSubmit(dipilih);
        }
    });
}

private void formSubmit(String hasil){
    AlertDialog.Builder dialog = new
AlertDialog.Builder(this);
    LayoutInflater inflater = this.getLayoutInflater();
    View dialogView =
inflater.inflate(R.layout.tabel_form_submit, null);
    dialog.setView(dialogView);
    dialog.setIcon(R.mipmap.fd);
    dialog.setTitle("MENU YANG DIPILIH");
    dialog.setCancelable(true);
    txt_tabel = (TextView)
dialogView.findViewById(R.id.txt_tabel);
    txt_tabel.setText(hasil);
    final String go = txt_tabel.toString().trim();
    dialog.setCancelable(false).setPositiveButton("PESAN",new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            //to perform on ok
            if(go == "ANDA BELUM MEMILIH MENU."){
                dialog.cancel();
            }else{
                /*
                for (TACheckOutData hold : modelArrayList) {
                    String[] arr = {
nama_pelanggan.getText().toString(),
                        String.valueOf(0),
                        hold.getId(),
String.valueOf(hold.getNumber())
                    };
                    SendGoYo(arr);
                }
                */
                SendGoYo();
            }
        }
    });
}

```

```

    }
    }).setNegativeButton("BATAL", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        dialog.cancel();
    }
});
dialog.show();
}
public void SendGoYo() {
    pDialog = ProgressDialog.show(this, null, null);
    pDialog.setContentView(R.layout.progress_bar);
    pDialog.show();
    pDialog.setCancelable(false);
    //final String id_meja =
txt_tabel.getText().toString().trim();
    //final String name =
txt_tabel.getText().toString().trim();
    /*
    final String nama_pemesan = arr[0].toString().trim();
    final String id_meja = arr[1].toString().trim();
    final String id_menu = arr[2].toString().trim();
    final String qty = arr[3].toString().trim();
    */
    StringRequest stringRequest = new
StringRequest(Request.Method.POST, REGISTER_URL,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Intent intent = new
Intent(TACheckOut.this, MainActivity.class);
                startActivity(intent);
                hidePDialog();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error)
{
                Toast.makeText(TACheckOut.this, error.toString(), Toast.LENGTH_LONG)
                .show();

                Intent intent = new
Intent(TACheckOut.this, MainActivity.class);
                startActivity(intent);
                hidePDialog();
            }
        }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<String,
String>();
            params.put(NAMA_PEMESAN,
nama_pelanggan.getText().toString());
            params.put(ID_MEJA, String.valueOf(0));
            int i=0;
            for (TACheckOutData hold : modelArrayList) {

```

```

        params.put(ID_MENU + "[" + (i) + "]",
hold.getId());
        params.put(QTY + "[" + (i) + "]",
String.valueOf(hold.getNumber()));
        i++;
    }
    return params;
}
};
RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.add(stringRequest);
}
private ArrayList<TACheckOutData> getModel() {
    ArrayList<TACheckOutData> list = new ArrayList<>();
    ArrayList<String> itemlistid =
getIntent().getStringArrayListExtra("arr_list_id");
    ArrayList<String> itemlistnama =
getIntent().getStringArrayListExtra("arr_list_nama");
    ArrayList<String> itemlistprice =
getIntent().getStringArrayListExtra("arr_list_price");
    for(int i = 0; i<itemlistid.size(); i++){
        TACheckOutData model = new TACheckOutData();
        model.setNumber(1);
        model.setId(itemlistid.get(i));
        model.setNama(itemlistnama.get(i));
        model.setPrice(itemlistprice.get(i));
        model.setPriceFake(itemlistprice.get(i));
        model.setPriceFakeRupiah(itemlistprice.get(i));
        list.add(model);
    }
    return list;
}
public void onDestroy() {
    super.onDestroy();
    hidePDialog();
}
private void hidePDialog() {
    if (pDialog != null) {
        pDialog.dismiss();
        pDialog = null;
    }
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            onBackPressed();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

TACheckOutAdapter.java

```

package tia.restodj;
import android.content.Context;

```

```

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.Locale;
public class TACheckOutAdapter extends BaseAdapter {
    NumberFormat rupiahFormat;
    private Context context;
    public TACheckOutAdapter(Context context) {
        this.context = context;
    }
    @Override
    public int getViewTypeCount() {
        return getCount();
    }
    @Override
    public int getItemViewType(int position) {
        return position;
    }

    @Override
    public int getCount() {
        return TACheckOut.modelArrayList.size();
    }
    @Override
    public Object getItem(int position) {
        return TACheckOut.modelArrayList.get(position);
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        final ViewHolder holder;
        if (convertView == null) {
            holder = new ViewHolder(); LayoutInflater inflater =
(LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView =
inflater.inflate(R.layout.check_out_list, null, true);
            holder.tvId = (TextView)
convertView.findViewById(R.id.id_menu);
            holder.tvNama = (TextView)
convertView.findViewById(R.id.nama_menu);
            holder.tvPrice = (TextView)
convertView.findViewById(R.id.price_menu);
            holder.tvPriceFake = (TextView)
convertView.findViewById(R.id.price_menu_fake);
            holder.tvPriceFakeRupiah = (TextView)
convertView.findViewById(R.id.price_menu_fake_rupiah);

```

```

        holder.tvnumber = (TextView)
convertView.findViewById(R.id.number);
        holder.btn_plus = (Button)
convertView.findViewById(R.id.plus);
        holder.btn_minus = (Button)
convertView.findViewById(R.id.minus);
        convertView.setTag(holder);
    }else {
set as a tag to the view
        holder = (ViewHolder)convertView.getTag();
    }
        rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
        String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(TACheckOut.m
odelArrayList.get(position).getPriceFakeRupiah())));
holder.tvId.setText(TACheckOut.modelArrayList.get(position).getId(
));
holder.tvNama.setText(TACheckOut.modelArrayList.get(position).getN
ama());
holder.tvPrice.setText(String.valueOf(TACheckOut.modelArrayList.ge
t(position).getPrice()));
holder.tvPriceFake.setText(String.valueOf(TACheckOut.modelArrayLis
t.get(position).getPriceFake()));
        holder.tvPriceFakeRupiah.setText(hargonyo);
holder.tvnumber.setText(String.valueOf(TACheckOut.modelArrayList.g
et(position).getNumber()));
        holder.btn_plus.setTag(R.integer.btn_plus_view,
convertView);
        holder.btn_plus.setTag(R.integer.btn_plus_pos, position);
        holder.btn_plus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                View tempview = (View)
holder.btn_plus.getTag(R.integer.btn_plus_view);
                TextView tv = (TextView)
tempview.findViewById(R.id.number);
                TextView tvpori = (TextView)
tempview.findViewById(R.id.price_menu);
                TextView tvpfake = (TextView)
tempview.findViewById(R.id.price_menu_fake);
                TextView tvpfakerupiah = (TextView)
tempview.findViewById(R.id.price_menu_fake_rupiah);
                Integer pos = (Integer)
holder.btn_plus.getTag(R.integer.btn_plus_pos);
                int number =
Integer.parseInt(tv.getText().toString()) + 1;
                int numbercal =
Integer.parseInt(tvpfake.getText().toString()) * number;
                String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(numbercal)))
;
                tvpori.setText(String.valueOf(numbercal));
                tvpfakerupiah.setText(String.valueOf(hargonyo));
                TACheckOut.modelArrayList.get(pos).setNumber(number);
            }
        });
    }
}

```

```

TACheckOut.modelArrayList.get(pos).setPrice(Integer.toString(numbercal));
TACheckOut.modelArrayList.get(pos).setPriceFakeRupiah(hargonyo);
    }
    });
    holder.btn_minus.setTag(R.integer.btn_minus_view,
convertView);
    holder.btn_minus.setTag(R.integer.btn_minus_pos,
position);
    holder.btn_minus.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            View tempview = (View)
holder.btn_minus.getTag(R.integer.btn_minus_view);
            TextView tv = (TextView)
tempview.findViewById(R.id.number);
            TextView tvpori = (TextView)
tempview.findViewById(R.id.price_menu);
            TextView tvpfake = (TextView)
tempview.findViewById(R.id.price_menu_fake);
            TextView tvpfakerupiah = (TextView)
tempview.findViewById(R.id.price_menu_fake_rupiah);
            Integer pos = (Integer)
holder.btn_minus.getTag(R.integer.btn_minus_pos);
            int number =
Integer.parseInt(tv.getText().toString()) - 1;
            int numbercal =
Integer.parseInt(tvpori.getText().toString()) -
Integer.parseInt(tvpfake.getText().toString());
            String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(numbercal)))
;
            tv.setText(String.valueOf(number));
            tvpori.setText(String.valueOf(numbercal));
            tvpfakerupiah.setText(hargonyo);
TACheckOut.modelArrayList.get(pos).setNumber(number);
TACheckOut.modelArrayList.get(pos).setPrice(Integer.toString(numbercal));
TACheckOut.modelArrayList.get(pos).setPriceFakeRupiah(hargonyo);
        }
    });
    return convertView;
}
private class ViewHolder {
    protected Button btn_plus, btn_minus;
    private TextView tvId,
tvNama, tvPrice, tvPriceFake, tvPriceFakeRupiah, tvnumber;
}
}

```

TACheckOutData.java

```

package tia.restodj;
public class TACheckOutData {
    private int number;
    private String price;
}

```



```

private String price_fake;
private String price_fake_rupiah;
private String id;
private String nama;
public String getPrice() {
    return price;
}
public void setPrice(String price) {
    this.price = price;
}
public String getPriceFake() {
    return price_fake;
}
public void setPriceFake(String price_fake) {
    this.price_fake = price_fake;
}
public String getPriceFakeRupiah() {
    return price_fake_rupiah;
}
public void setPriceFakeRupiah(String price_fake_rupiah) {
    this.price_fake_rupiah = price_fake_rupiah;
}
public int getNumber() {
    return number;
}
public void setNumber(int number) {
    this.number = number;
}
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getNama() {
    return nama;
}
public void setNama(String nama) {
    this.nama = nama;
}
}

```

TAMenuData.java

```

package tia.restodj;
public class TAMenuData {
    private String price,thumbnailUrl;
    private String id;
    private String tabel;
    private boolean check;
    public TAMenuData() {}
    public TAMenuData(String price, String thumbnailUrl, String
id,String tabel, boolean check) {
        this.price = price;
        this.thumbnailUrl = thumbnailUrl;
        this.id = id;
        this.tabel = tabel;
    }
}

```

```

        this.check = check;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
    public String getThumbnailUrl() {
        return thumbnailUrl;
    }
    public void setThumbnailUrl(String thumbnailUrl) {
        this.thumbnailUrl = thumbnailUrl;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getTabel() {
        return tabel;
    }
    public void setTabel(String tabel) {
        this.tabel = tabel;
    }
    public boolean isCheckbox() {
        return check;
    }
    public void setCheckbox(boolean check) {
        this.check = check;
    }
}

```

TAMenuOrderAdapter.java

```

package tia.restodj;
import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.CheckBox;
import android.widget.TextView;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Locale;
public class TAMenuOrderAdapter extends BaseAdapter {
    NumberFormat rupiahFormat;
    private Activity activity;
    private ArrayList<TAMenuData> data;
    private LayoutInflater inflater;
    private View vi;

```

```

        private ViewHolder viewHolder;
        ImageLoader imageLoader =
AppController.getInstance().getImageLoader();

        public TAMenuOrderAdapter(Activity context,
ArrayList<TAMenuData> items) {
            this.activity = context;
            this.data = items;
        }
        @Override
        public int getCount() {
            return data.size();
        }
        @Override
        public Object getItem(int i) {
            return data.get(i);
        }
        @Override
        public long getItemId(int i) {
            return i;
        }
        @Override
        public View getView(final int position, View convertView,
ViewGroup viewGroup) {
            final int pos = position;
            TAMenuData items = data.get(pos);
            /*
            if(view == null) {
                vi = inflater.inflate(R.layout.menu_list, null,true);
                if (imageLoader == null) imageLoader =
AppController.getInstance().getImageLoader();
                viewHolder = new ViewHolder();
                viewHolder.thumbNail = (NetworkImageView)
vi.findViewById(R.id.thumbnail);
                viewHolder.checkBox = (CheckBox)
vi.findViewById(R.id.cb);
                viewHolder.tabel = (TextView)
vi.findViewById(R.id.nama_menu);
                vi.setTag(viewHolder);
            }else {
                viewHolder = (ViewHolder) view.getTag();
                viewHolder.tabel.setText(items.getTabel());
viewHolder.thumbNail.setImageUrl(items.getThumbnailUrl(),
imageLoader);
            }
            */
            if (inflater == null)
                inflater = (LayoutInflater) activity
.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            if (convertView == null)
                convertView = inflater.inflate(R.layout.menu_list,
null);
            if (imageLoader == null) imageLoader =
AppController.getInstance().getImageLoader();
            NetworkImageView thumbNail = (NetworkImageView)
convertView.findViewById(R.id.thumbnail);

```

```

        CheckBox checkBox = (CheckBox)
convertView.findViewById(R.id.cb);
        TextView tabel = (TextView)
convertView.findViewById(R.id.nama_menu);
        TextView price = (TextView)
convertView.findViewById(R.id.harga);
        rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
        String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(items.getPri
ce())));
        thumbNail.setImageUrl(items.getThumbnailUrl(),
imageLoader);
        tabel.setText(items.getTabel());
        price.setText(hargonyo);
        if(items.isCheckbox()){
            checkBox.setChecked(true);
        } else {
            checkBox.setChecked(false);
        }
        return convertView;
    }
    public ArrayList<TAMenuData> getAllData(){
        return data;
    }
    public void setCheckBox(int position){
        TAMenuData items = data.get(position);
        items.setCheckbox(!items.isCheckbox());
        notifyDataSetChanged();
    }
    public class ViewHolder{
        TextView tabel;
        CheckBox checkBox;
        NetworkImageView thumbNail;
    }
}

```

TAMenuOrderFragment.java

```

package tia.restodj;
import android.app.ActionBar;
import android.app.AlertDialog;
import android.app.Fragment;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;

```

```

import com.android.volley.toolbox.JsonArrayRequest;
import tia.restodj.R;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class TAMenuOrderFragment extends Fragment {
    private ActionBar actionBar;
    String IdOrder, Nama;
    Button fab;
    ListView list;
    TextView txt_tabel;
    String dipilih;
    String[] arr;
    private static final String TAG =
MainActivity.class.getSimpleName();
    TAMenuOrderAdapter adapter;
    ProgressDialog pDialog;
    List<TAMenuData> itemList = new ArrayList<TAMenuData>();
    private static String url =
"http://dapurjawarestosa.16mb.com/api/menu";
    public static final String TAG_ID = "id menu";
    public static final String TAG_NAMA = "nama menu";
    public static final String TAG_PRICE = "price";
    public static final String TAG_PICTURE = "foto";
    public interface Callbacks {
        public void onBackPressed();
    }
    public TAMenuOrderFragment() {
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.menu_order,
container, false);
        fab = (Button) rootView.findViewById(R.id.fab);
        list = (ListView) rootView.findViewById(R.id.lv);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String checkbox = "";
                for (TAMenuData hold : adapter.getAllData()) {
                    if (hold.isCheckbox()) {
                        checkbox += "\n" + hold.getTabel();
                    }
                }
                if (!checkbox.isEmpty()) {
                    dipilih = checkbox;
                } else {
                    dipilih = "ANDA BELUM MEMILIH MENU.";
                }
                formSubmit(dipilih);
            }
        });
    }
};

```

```

        adapter = new TAMenuOrderAdapter(getActivity(),
(ArrayList<TAMenuData>) itemList);
        list.setAdapter(adapter);
        callVolley();
        list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView,
View view, int position, long l) {
                adapter.setCheckBox(position);
            }
        });
        return rootView;
    }
    private void formSubmit(final String hasil){
        AlertDialog.Builder dialog = new
AlertDialog.Builder(getActivity());
        LayoutInflater inflater =
getActivity().getLayoutInflater();
        View dialogView =
inflater.inflate(R.layout.tabel_form_submit, null);
        dialog.setView(dialogView);
        dialog.setIcon(R.mipmap.fd);
        dialog.setTitle("MENU YANG DIPILIH");
        dialog.setCancelable(true);
        txt_tabel = (TextView)
dialogView.findViewById(R.id.txt_tabel);
        txt_tabel.setText(hasil);
        final String go = txt_tabel.toString().trim();
        dialog.setCancelable(false).setPositiveButton("OK",new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                String checkbox = "";
                for (TAMenuData hold : adapter.getAllData()) {
                    if (hold.isCheckBox()) {
                        checkbox += "\n" + hold.getTabel();
                    }
                }
                if (!checkbox.isEmpty()) {
                    ArrayList<String> arraid = new
ArrayList<String>();
                    ArrayList<String> arranama = new
ArrayList<String>();
                    ArrayList<String> arraprice = new
ArrayList<String>();
                    for (TAMenuData hold : adapter.getAllData()) {
                        if (hold.isCheckBox()) {
                            String[] arr = {
                                hold.getId(),
                                hold.getTabel(),
                                hold.getPrice()
                            };
                            arraid.add(arr[0]);
                            arranama.add(arr[1]);
                            arraprice.add(arr[2]);
                        }
                    }
                }
            }
        });
    }
}

```

```

        }
        Intent intent = new Intent(getActivity(),
TACheckOut.class);
intent.putStringArrayListExtra("arr_list_id", arraid);
intent.putStringArrayListExtra("arr_list_nama", arranama);
intent.putStringArrayListExtra("arr_list_price", arraprice);
        startActivity(intent);
    }else{
        dialog.cancel();
    }
    }
    }).setNegativeButton("BATAL", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        dialog.cancel();
    }
});
    dialog.show();
}
private void callVolley(){
    itemList.clear();
    pDialog = ProgressDialog.show(getActivity(), null, null);
    pDialog.setContentView(R.layout.progress_bar);
    pDialog.show();
    pDialog.setCancelable(false);
    JsonRequest jArr = new JsonRequest(url, new
Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            Log.d(TAG, response.toString());
            hidePDialog();
            for (int i = 0; i < response.length(); i++) {
                try {
                    JSONObject obj =
response.getJSONObject(i);
                    TAMenuData item = new TAMenuData()
item.setId(obj.getString(TAG_ID));
item.setTabel(obj.getString(TAG_NAMA));
item.setPrice(obj.getString(TAG_PRICE));
item.setThumbnailUrl(obj.getString(TAG_PICTURE));
                    itemList.add(item);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
            adapter.notifyDataSetChanged();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            VolleyLog.d(TAG, "Error: " + error.getMessage());
            hidePDialog();
        }
    });
    ApplicationController.getInstance().addToRequestQueue(jArr);
}

```

```

    public void onDestroy() {
        super.onDestroy();
        hidePDialog();
    }
    private void hidePDialog() {
        if (pDialog != null) {
            pDialog.dismiss();
            pDialog = null;
        }
    }
    public void onBackPressed() {
        Intent intent = new Intent(getActivity(),
MainActivity.class);
        startActivity(intent);
    }
}

```

TOCheckOut.java

```

package tia.restodj;
import android.app.ActionBar;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.PorterDuff;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
public class TOCheckOut extends Activity {
    private ActionBar actionBar;
    private ListView lv;
    public static ArrayList<TOCheckOutData> modelArrayList;
    private tia.restodj.TOCheckOutAdapter TOCheckOutAdapter;
    private Button btnnext;
    String dipilih;
    TextView txt_tabel;
    EditText nama_pelanggan;

```



```

        String IdOrder, Nama;
        ProgressDialog pDialog;
        NumberFormat rupiahFormat;
        private static final String TAG =
MainActivity.class.getSimpleName();
        private static final String REGISTER_URL =
"http://dapurjawarestosa.16mb.com/api/order";
        public static final String NAMA_PEMESAN = "nama_pemesan";
        public static final String ID_MEJA = "id_meja";
        public static final String ID_MENU = "id_menu";
        public static final String QTY = "qty";
        public interface Callbacks {
            public void onBackPressed();
        }
        @Override
        public void onBackPressed() {
            super.onBackPressed();
            Intent intent = new Intent(TOCheckout.this,
MainActivity.class);
            startActivity(intent);
            finish();
        }
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.check_out);
            actionBar = getSupportActionBar();
            actionBar.setDisplayHomeAsUpEnabled(true);
            actionBar.setTitle("KEMBALI");
            lv = (ListView) findViewById(R.id.lv);
            btnnext = (Button) findViewById(R.id.next);
            nama_pelanggan = (EditText)
findViewById(R.id.nama_pelanggan);
            IdOrder = getIntent().getExtras().getString("IdTabel");
            nama_pelanggan.getBackground().setColorFilter(getResources().getColor(R.color.colorBacround), PorterDuff.Mode.ADD.SRC_IN);

            modelArrayList = getModel();
            TOCheckoutAdapter = new TOCheckoutAdapter(this);
            lv.setAdapter(TOCheckoutAdapter);
            rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
            btnnext.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    if(nama_pelanggan.getText().toString().length()==0)
                    {
                        nama_pelanggan.setError("Tolong Isi Nama
Anda");
                    }
                    return;
                }
            }
            else
            {
                String nama = "";
                int sum=0;
                for (TOCheckoutData hold : modelArrayList) {
                    /*

```

```

        int arr[]={19,20,20,20,20};
        int sum=0;
        for(int i:arr) {
            sum += i;
        }
        /*
        sum
+=Integer.parseInt(hold.getPrice().toString());
        String hargatambahqty =
rupiahFormat.format(Double.parseDouble(hold.getPrice()));
        nama += "\n" + hold.getNama() +
            " : " + hold.getNumber() +
            " : " + hargatambahqty;
    }
        String pelanggan = "A.N : " +
nama_pelanggan.getText().toString() + "\n" ;
        String totalhargo =
rupiahFormat.format(Double.parseDouble(String.valueOf(sum)));
        String fakesum = "\n" + "\n" + "Total : " +
totalhargo ;

        if (!nama.isEmpty()) {
            dipilih = pelanggan + nama + fakesum;
        } else {
            dipilih = "ANDA BELUM MEMILIH MENU.";
        }
        formSubmit(dipilih);
    }
    });
}
private void formSubmit(String hasil){
    AlertDialog.Builder dialog = new
AlertDialog.Builder(this);
    LayoutInflater inflater = this.getLayoutInflater();
    View dialogView =
inflater.inflate(R.layout.tabel_form_submit, null);
    dialog.setView(dialogView);
    dialog.setIcon(R.mipmap.fd);
    dialog.setTitle("MENU YANG DIPILIH");
    dialog.setCancelable(true);
    txt_tabel = (TextView)
dialogView.findViewById(R.id.txt_tabel);
    txt_tabel.setText(hasil);
    final String go = txt_tabel.toString().trim();
    dialog.setCancelable(false).setPositiveButton("PESAN",new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            if(go == "ANDA BELUM MEMILIH MENU."){
                dialog.cancel();
            }else{
                /*
                String[] arr = new String[]{};
                for (TOCheckoutData hold : modelArrayList) {
                    arr = new String[]{
nama_pelanggan.getText().toString(),
                        IdOrder,

```

```

                                hold.getId(),
String.valueOf(hold.getNumber())
                                };
                                }
                                */
                                SendGoYo();
                                }
                                }
                                }).setNegativeButton("BATAL", new
DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int id) {
                                dialog.cancel();
                                }
                                });
                                dialog.show();
                                }
                                public void SendGoYo() {
                                pDialog = ProgressDialog.show(this, null, null);
                                pDialog.setContentView(R.layout.progress_bar);
                                pDialog.show();
                                pDialog.setCancelable(false);
                                //final String id_meja =
txt_tabel.getText().toString().trim();
                                //final String name =
txt_tabel.getText().toString().trim();
                                /*
                                final String nama_pemesan = arr[0].toString().trim();
                                final String id_meja = arr[1].toString().trim();
                                final String id_menu = arr[2].toString().trim();
                                final String qty = arr[3].toString().trim();
                                */
                                StringRequest stringRequest = new
StringRequest(Request.Method.POST, REGISTER_URL,
                                new Response.Listener<String>() {
                                @Override
                                public void onResponse(String response) {
                                Intent intent = new
Intent(TOCheckout.this, MainActivity.class);
                                startActivity(intent);
                                hidePDialog();
                                }
                                },
                                new Response.ErrorListener() {
                                @Override
                                public void onErrorResponse(VolleyError error)
                                {
                                Toast.makeText(TOCheckout.this, error.toString(), Toast.LENGTH_LONG)
                                .show();
                                Intent intent = new
Intent(TOCheckout.this, MainActivity.class);
                                startActivity(intent);
                                hidePDialog();
                                }
                                }) {
                                @Override
                                protected Map<String, String> getParams() {

```

```

String>());
        Map<String,String> params = new HashMap<String,
        String>();
        /*
        params.put (NAMA_PEMESAN,nama_pemesan);
        params.put (ID_MEJA,id_meja);
        params.put (ID_MENU,id_menu);
        params.put (QTY,qty);
        */
        /*
        ArrayList<String> itemlistid =
        getIntent().getStringArrayListExtra("arr_list_id");
        for(int i=0; i<itemlistid.size(); i++) {
            //params.put (TAG_IDS + "[" + i + "]",
        all_ids.get(i).toString());
            params.put (NAMA_PEMESAN + "[" + i +
        "]" ,nama_pelanggan.getText().toString());
            params.put (ID_MEJA + "[" + i + "]",IdOrder);
            params.put (ID_MENU + "[" + i +
        "]" ,itemlistid.get(i).toString());
            params.put (QTY + "[" + i +
        "]" ,itemlistid.get(i).toString());
        }
        */
        params.put (NAMA_PEMESAN,
        nama_pelanggan.getText().toString());
        params.put (ID_MEJA, IdOrder);
        int i=0;
        for (TOCheckOutData hold : modelArrayList) {

            params.put (ID_MENU + "[" +(i)+ "]",
        hold.getId());
            params.put (QTY + "[" +(i)+ "]",
        String.valueOf(hold.getNumber()));
            i++;
        }
        return params;
    }
};
RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.add(stringRequest);
}
private ArrayList<TOCheckOutData> getModel() {
    ArrayList<TOCheckOutData> list = new ArrayList<>();
    ArrayList<String> itemlistid =
        getIntent().getStringArrayListExtra("arr_list_id");
    ArrayList<String> itemlistnama =
        getIntent().getStringArrayListExtra("arr_list_nama");
    ArrayList<String> itemlistprice =
        getIntent().getStringArrayListExtra("arr_list_price");
    for(int i = 0; i<itemlistid.size(); i++){
        TOCheckOutData model = new TOCheckOutData();
        model.setNumber(1);
        model.setId(itemlistid.get(i));
        model.setNama(itemlistnama.get(i));
        model.setPrice(itemlistprice.get(i));
        model.setPriceFake(itemlistprice.get(i));
    }
}

```

```

        model.setPriceFakeRupiah(itemlistprice.get(i));
        list.add(model);
    }
    return list;
}
public void onDestroy() {
    super.onDestroy();
    hidePDialog();
}
private void hidePDialog() {
    if (pDialog != null) {
        pDialog.dismiss();
        pDialog = null;
    }
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            onBackPressed();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

TOCheckOutAdapter.java

```

package tia.restodj;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.Locale;
public class TOCheckOutAdapter extends BaseAdapter {
    NumberFormat rupiahFormat;
    private Context context;
    public TOCheckOutAdapter(Context context) {
        this.context = context;
    }
    @Override
    public int getViewTypeCount() {
        return getCount();
    }
    @Override
    public int getItemViewType(int position) {
        return position;
    }
    @Override
    public int getCount() {
        return TOCheckOut.modelArrayList.size();
    }
}

```

```

@Override
public Object getItem(int position) {
    return TOCheckout.modelArrayList.get(position);
}
@Override
public long getItemId(int position) {
    return 0;
}
@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    final ViewHolder holder;
    if (convertView == null) {
        holder = new ViewHolder(); LayoutInflater inflater =
(LayoutInflater) context
.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView =
inflater.inflate(R.layout.check_out_list, null, true);
        holder.tvId = (TextView)
convertView.findViewById(R.id.id_menu);
        holder.tvNama = (TextView)
convertView.findViewById(R.id.nama_menu);
        holder.tvPrice = (TextView)
convertView.findViewById(R.id.price_menu);
        holder.tvPriceFake = (TextView)
convertView.findViewById(R.id.price_menu_fake);
        holder.tvPriceFakeRupiah = (TextView)
convertView.findViewById(R.id.price_menu_fake_rupiah);
        holder.tvnumber = (TextView)
convertView.findViewById(R.id.number);
        holder.btn_plus = (Button)
convertView.findViewById(R.id.plus);
        holder.btn_minus = (Button)
convertView.findViewById(R.id.minus);
        convertView.setTag(holder);
    }else {
        holder = (ViewHolder) convertView.getTag();
    }
    rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
    String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(TOCheckout.m
odelArrayList.get(position).getPriceFakeRupiah())));
    holder.tvId.setText(TOCheckout.modelArrayList.get(position).getId(
));
    holder.tvNama.setText(TOCheckout.modelArrayList.get(position).getN
ama());
    holder.tvPrice.setText(String.valueOf(TOCheckout.modelArrayList.ge
t(position).getPrice()));
    holder.tvPriceFake.setText(String.valueOf(TOCheckout.modelArraylis
t.get(position).getPriceFake()));
        holder.tvPriceFakeRupiah.setText(hargonyo);
    holder.tvnumber.setText(String.valueOf(TOCheckout.modelArrayList.g
et(position).getNumber()));
        holder.btn_plus.setTag(R.integer.btn_plus_view,
convertView);
        holder.btn_plus.setTag(R.integer.btn_plus_pos, position);

```

```

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

                View tempview = (View)
holder.btn_plus.getTag(R.integer.btn_plus_view);
                TextView tv = (TextView)
tempview.findViewById(R.id.number);
                TextView tvpori = (TextView)
tempview.findViewById(R.id.price_menu);
                TextView tvpfake = (TextView)
tempview.findViewById(R.id.price_menu_fake);
                TextView tvpfakerupiah = (TextView)
tempview.findViewById(R.id.price_menu_fake_rupiah);
                Integer pos = (Integer)
holder.btn_plus.getTag(R.integer.btn_plus_pos);
                int number =
Integer.parseInt(tv.getText().toString()) + 1;
                int numbercal =
Integer.parseInt(tvpfake.getText().toString()) * number;
                String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(numbercal)))
;
                tv.setText(String.valueOf(number));
                tvpori.setText(String.valueOf(numbercal));
                tvpfakerupiah.setText(String.valueOf(hargonyo));
                TOCheckout.modelArrayList.get(pos).setNumber(number);
                TOCheckout.modelArrayList.get(pos).setPrice(Integer.toString(numbe
rcal));
                TOCheckout.modelArrayList.get(pos).setPriceFakeRupiah(hargonyo);
            }
        });
        holder.btn_minus.setTag(R.integer.btn_minus_view,
convertView);
        holder.btn_minus.setTag(R.integer.btn_minus_pos,
position);
        holder.btn_minus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                View tempview = (View)
holder.btn_minus.getTag(R.integer.btn_minus_view);
                TextView tv = (TextView)
tempview.findViewById(R.id.number);
                TextView tvpori = (TextView)
tempview.findViewById(R.id.price_menu);
                TextView tvpfake = (TextView)
tempview.findViewById(R.id.price_menu_fake);
                TextView tvpfakerupiah = (TextView)
tempview.findViewById(R.id.price_menu_fake_rupiah);
                Integer pos = (Integer)
holder.btn_minus.getTag(R.integer.btn_minus_pos);
                int number =
Integer.parseInt(tv.getText().toString()) - 1;

```

```

        int numbercal =
Integer.parseInt(tvpori.getText().toString()) -
Integer.parseInt(tvpfake.getText().toString());
        String hargonyo =
rupiahFormat.format(Double.parseDouble(String.valueOf(numbercal)))
;
        tv.setText(String.valueOf(number));
tvpori.setText(String.valueOf(numbercal));
        tvpfakerupiah.setText(hargonyo);
TOCheckout.modelArrayList.get(pos).setNumber(number);
TOCheckout.modelArrayList.get(pos).setPrice(Integer.toString(numbercal));
TOCheckout.modelArrayList.get(pos).setPriceFakeRupiah(hargonyo);
    }
    });
    return convertView;
}
private class ViewHolder {
    protected Button btn_plus, btn_minus;
    private TextView tvId,
tvNama, tvPrice, tvPriceFake, tvPriceFakeRupiah, tvnumber;
}
}

```

TOCheckoutData.java

```

package tia.restodj;
public class TOCheckoutData {
    private int number;
    private String price;
    private String price_fake;
    private String price_fake_rupiah;
    private String id;
    private String nama;
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
    public String getPriceFake() {
        return price_fake;
    }
    public void setPriceFake(String price_fake) {
        this.price_fake = price_fake;
    }
    public String getPriceFakeRupiah() {
        return price_fake_rupiah;
    }
    public void setPriceFakeRupiah(String price_fake_rupiah) {
        this.price_fake_rupiah = price_fake_rupiah;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
}

```



```

    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
}

```

TOMenuAdapter.java

```

package tia.restodj;
import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.CheckBox;
import android.widget.TextView;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;
import tia.restodj.R;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Locale;
public class TOMenuAdapter extends BaseAdapter {
    NumberFormat rupiahFormat;
    private Activity activity;
    private ArrayList<TOMenuData> data;
    private LayoutInflater inflater;
    private View vi;
    private ViewHolder viewHolder;
    ImageLoader imageLoader =
AppController.getInstance().getImageLoader();
    public TOMenuAdapter(Activity context, ArrayList<TOMenuData>
items) {
        this.activity = context;
        this.data = items;
    }
    @Override
    public int getCount() {
        return data.size();
    }
    @Override
    public Object getItem(int i) {
        return data.get(i);
    }
    @Override
    public long getItemId(int i) {

```

```

        return i;
    }
    @Override
    public View getView(final int position, View convertView,
    ViewGroup viewGroup) {
        final int pos = position;
        TOMenuData items = data.get(pos);
        /*
        if(view == null) {
            vi = inflater.inflate(R.layout.menu_list, null,true);
            if (imageLoader == null) imageLoader =
    ApplicationController.getInstance().getImageLoader();
            viewHolder = new ViewHolder();
            viewHolder.thumbNail = (NetworkImageView)
    vi.findViewById(R.id.thumbnail);
            viewHolder.checkBox = (CheckBox)
    vi.findViewById(R.id.cb);
            viewHolder.tabel = (TextView)
    vi.findViewById(R.id.nama_menu);
            vi.setTag(viewHolder);
        }else {
            viewHolder = (ViewHolder) view.getTag();
            viewHolder.tabel.setText(items.getTabel());
    viewHolder.thumbNail.setImageUrl(items.getThumbnailUrl(),
    imageLoader);
        }
        */
        if (inflater == null)
            inflater = (LayoutInflater) activity
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        if (convertView == null)
            convertView = inflater.inflate(R.layout.menu_list,
    null);
        if (imageLoader == null) imageLoader =
    ApplicationController.getInstance().getImageLoader();
        NetworkImageView thumbNail = (NetworkImageView)
    convertView.findViewById(R.id.thumbnail);
        CheckBox checkBox = (CheckBox)
    convertView.findViewById(R.id.cb);
        TextView tabel = (TextView)
    convertView.findViewById(R.id.nama_menu);
        TextView price = (TextView)
    convertView.findViewById(R.id.harga);
        rupiahFormat = NumberFormat.getInstance(Locale.GERMANY);
        String hargonyo =
    rupiahFormat.format(Double.parseDouble(String.valueOf(items.getPri
    ce())));
        thumbNail.setImageUrl(items.getThumbnailUrl(),
    imageLoader);
        tabel.setText(items.getTabel());
        price.setText(hargonyo);
        if(items.isCheckbox()){
            checkBox.setChecked(true);
        } else {
            checkBox.setChecked(false);
        }
    }

```

```

        return convertView;
    }
    public ArrayList<TOMenuData> getAllData(){
        return data;
    }
    public void setCheckBox(int position){
        TOMenuData items = data.get(position);
        items.setCheckbox(!items.isCheckbox());
        notifyDataSetChanged();
    }
    public class ViewHolder{
        TextView tabel;
        CheckBox checkBox;
        NetworkImageView thumbNail;
    }
}

```

TOMenuData.java

```

package tia.restodj;
public class TOMenuData {
    private String price,thumbnailUrl;
    private String id;
    private String tabel;
    private boolean check;
    public TOMenuData() {}
    public TOMenuData(String price, String thumbnailUrl, String
id,String tabel, boolean check) {
        this.price = price;
        this.thumbnailUrl = thumbnailUrl;
        this.id = id;
        this.tabel = tabel;
        this.check = check;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
    public String getThumbnailUrl() {
        return thumbnailUrl;
    }
    public void setThumbnailUrl(String thumbnailUrl) {
        this.thumbnailUrl = thumbnailUrl;
    }
    public String getId() {
        return id;
    }
}

public void setId(String id) {
    this.id = id;
}
public String getTabel() {
    return tabel;
}
public void setTabel(String tabel) {

```

```

        this.tabel = tabel;
    }
    public boolean isChecked() {
        return check;
    }
    public void setCheckbox(boolean check) {
        this.check = check;
    }
}

```

TOMenuOrder.java

```

package tia.restodj;
import android.app.ActionBar;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonArrayRequest;
import tia.restodj.R;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class TOMenuOrder extends Activity {
    private ActionBar actionBar;
    String IdOrder, Nama;
    Button fab;
    ListView list;
    TextView txt_tabel;
    String dipilih;
    String[] arr;
    private static final String TAG =
MainActivity.class.getSimpleName();
    TOMenuAdapter adapter;
    ProgressDialog pDialog;
    List<TOMenuData> itemList = new ArrayList<TOMenuData>();
    private static String url =
"http://dapurjawarestosa.16mb.com/api/menu";
    public static final String TAG_ID = "id_menu";
    public static final String TAG_NAMA = "nama_menu";
    public static final String TAG_PRICE= "price";
    public static final String TAG_PICTURE= "foto";

```

```

public interface Callbacks {
    public void onBackPressed();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.menu_order);
    actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setTitle("KEMBALI");
    IdOrder = getIntent().getExtras().getString("IdTabel");
    fab      = (Button) findViewById(R.id.fab);
    list     = (ListView) findViewById(R.id.lv);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String checkbox = "";
            for (TOMenuData hold : adapter.getAllData()) {
                if (hold.isCheckbox()) {
                    checkbox += "\n" + hold.getTabel();
                }
            }
            if (!checkbox.isEmpty()) {
                dipilih = checkbox;
            } else {
                dipilih = "ANDA BELUM MEMILIH MENU.";
            }
            formSubmit(dipilih);
        }
    });
    adapter = new TOMenuAdapter(this, (ArrayList<TOMenuData>)
itemList);
    list.setAdapter(adapter);
    callVolley();
    list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView,
View view, int position, long l) {
            adapter.setCheckBox(position);
        }
    });
}
private void formSubmit(final String hasil){
    AlertDialog.Builder dialog = new
AlertDialog.Builder(this);
    LayoutInflater inflater = this.getLayoutInflater();
    View dialogView =
inflater.inflate(R.layout.tabel_form_submit, null);
    dialog.setView(dialogView);
    dialog.setIcon(R.mipmap.fd);
    dialog.setTitle("MENU YANG DIPILIH");
    dialog.setCancelable(true);
    txt_tabel = (TextView)
dialogView.findViewById(R.id.txt_tabel);
    txt_tabel.setText(hasil);
}

```

```

        final String go = txt_tabel.toString().trim();
        dialog.setCancelable(false).setPositiveButton("OK", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                String checkbox = "";
                for (TOMenuData hold : adapter.getAllData()) {
                    if (hold.isCheckbox()) {
                        checkbox += "\n" + hold.getTabel();
                    }
                }
                if (!checkbox.isEmpty()) {
                    ArrayList<String> arraid = new
ArrayList<String>();
                    ArrayList<String> arranama = new
ArrayList<String>();
                    ArrayList<String> arraprice = new
ArrayList<String>();
                    for (TOMenuData hold : adapter.getAllData()) {
                        if (hold.isCheckbox()) {
                            String[] arr = {
                                hold.getId(),
                                hold.getTabel(),
                                hold.getPrice()
                            };
                            arraid.add(arr[0]);
                            arranama.add(arr[1]);
                            arraprice.add(arr[2]);
                        }
                    }
                    Intent intent = new Intent(TOMenuOrder.this,
TOCheckOut.class);
                    intent.putStringArrayListExtra("arr_list_id", arraid);
                    intent.putStringArrayListExtra("arr_list_nama", arranama);
                    intent.putStringArrayListExtra("arr_list_price", arraprice);
                    intent.putExtra("IdTabel", IdOrder);
                    startActivity(intent);
                }else{
                    dialog.cancel();
                }
            }
        }).setNegativeButton("BATAL", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        dialog.show();
    }
    private void callVolley(){
        itemList.clear();
        pDialog = ProgressDialog.show(TOMenuOrder.this, null, null);
        pDialog.setContentView(R.layout.progress_bar);
        pDialog.show();
        pDialog.setCancelable(false);
    }

```

```

        JSONArrayRequest jArr = new JSONArrayRequest(url, new
Response.Listener<JSONArray>() {
    @Override
    public void onResponse(JSONArray response) {
        Log.d(TAG, response.toString());
        hideProgressDialog();
        for (int i = 0; i < response.length(); i++) {
            try {
                JSONObject obj =
response.getJSONObject(i);
                TOMenuData item = new TOMenuData();
                item.setId(obj.getString(TAG_ID));
                item.setLabel(obj.getString(TAG_NAME));
                item.setPrice(obj.getString(TAG_PRICE));
                item.setThumbnailUrl(obj.getString(TAG_PICTURE));
                itemList.add(item);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        adapter.notifyDataSetChanged();
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        VolleyLog.d(TAG, "Error: " + error.getMessage());
        hideProgressDialog();
    }
}); AppController.getInstance().addToRequestQueue(jArr);
}
public void onDestroy() {
    super.onDestroy();
    hideProgressDialog();
}
private void hideProgressDialog() {
    if (pDialog != null) {
        pDialog.dismiss();
        pDialog = null;
    }
}
@Override
public void onBackPressed() {
    super.onBackPressed();
    Intent intent = new Intent(TOMenuOrder.this,
MainActivity.class);
    startActivity(intent);
    finish();
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            onBackPressed();
            return true;
    }
    return super.onOptionsItemSelected(item);
}

```

```

    }
}

```

TOOrderAdapter.java

```

package tia.restodj;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
import tia.restodj.R;
import java.util.List;

public class TOOrderAdapter extends BaseAdapter {
    private Activity activity;
    private LayoutInflater inflater;
    private List<TOOrderData> alldataItems;
    public TOOrderAdapter(Activity activity, List<TOOrderData>
alldataItems) {
        this.activity = activity;
        this.alldataItems = alldataItems;
    }
    @Override
    public int getCount() {
        return alldataItems.size();
    }
    @Override
    public Object getItem(int location) {
        return alldataItems.get(location);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        if (inflater == null)
            inflater = (LayoutInflater)
activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        if (convertView == null)
            convertView = inflater.inflate(R.layout.order_list,
null);
        TextView nama = (TextView)
convertView.findViewById(R.id>Nama);
        TOOrderData m = alldataItems.get(position);
        nama.setText(m.getNama());
        return convertView;
    }
}

```

TOOrderData.java

```

package tia.restodj;
public class TOOrderData {

```



```

private String id;
private String nama;
public TOOrderData() {}
public TOOrderData(String id,String nama) {
    this.id = id;
    this.nama = nama;
}
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getNama() {
    return nama;
}
public void setNama(String nama) {
    this.nama = nama;
}
}

```

TOOrderFragment.java

```

package tia.restodj;
import android.app.Fragment;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonArrayRequest;
import tia.restodj.R;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class TOOrderFragment extends Fragment {

    private static final String TAG =
TOOrderFragment.class.getSimpleName();
    private ProgressDialog pDialog;
    private List<tia.restodj.TOOrderData> TOOrderData = new
ArrayList<tia.restodj.TOOrderData>();
    private ListView listView;
    private TOOrderAdapter adapter;
    public TOOrderFragment() {}
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {

```

```

        View rootView = inflater.inflate(R.layout.fragment_order,
container, false);
        GetOrder();
        listView = (ListView)
rootView.findViewById(R.id.list_order);
        adapter = new TOrderAdapter(getActivity(), TOrderData);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                Intent intent = new Intent(getActivity(),
TMenuOrder.class);
                intent.putExtra("IdTabel",
TOrderData.get(position).getId());
                startActivity(intent);
            }
        });
        return rootView;
    }
    public void GetOrder(){
        progressDialog = ProgressDialog.show(getActivity(), null, null);
        progressDialog.setContentView(R.layout.progress_bar);
        progressDialog.show();
        progressDialog.setCancelable(false);
        String url = "http://dapurjawarestosa.16mb.com/api/meja";
        JSONArrayRequest AllDataOne = new JSONArrayRequest(url,
            new Response.Listener<JSONArray>() {
                @Override
                public void onResponse(JSONArray response) {
                    Log.d(TAG, response.toString());
                    hideProgressDialog();
                    for (int i = 0; i < response.length();
i++) {
                        try {
                            JSONObject obj =
response.getJSONObject(i);
                            tia.restodj.TOrderData listdata =
new TOrderData();
                            listdata.setId(obj.getString("id_meja"));
                            listdata.setNama(obj.getString("nama_meja"));
                            TOrderData.add(listdata);
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                    adapter.notifyDataSetChanged();
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    VolleyLog.d(TAG, "Error: " + error.getMessage());
                    hideProgressDialog();
                }
            }
        });
    }
}

```

```
        });  
AppController.getInstance().addToRequestQueue (AllDataOne;  
    }  
    @Override  
    public void onDestroy() {  
        super.onDestroy();  
        hidePDialog();  
    }  
    private void hidePDialog() {  
        if (pDialog != null) {  
            pDialog.dismiss();  
            pDialog = null;  
        }  
    }  
}
```

LISTING PROGRAM APLIKASI WEB

dashboard.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-12">
      <h3 class="page-header">Home</h3>
    </div>
    <!-- /.col-lg-12 -->
  </div>
  <div class="row">
    <div class="col-lg-3 col-md-6">
      <div class="panel panel-info">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-comment fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div class="huge"><?=$pesanan;?></div>
              <div>Pesanan Hari ini</div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col-lg-3 col-md-6">
      <div class="panel panel-green">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-shopping-cart fa-
5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div
class="huge"><?=$penjualan;?></div>
              <div>Penjualan Hari ini</div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col-lg-3 col-md-6">
      <div class="panel panel-yellow">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-bookmark-o fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div
class="huge"><?=$meja_tersedia;?></div>
              <div>Meja Tersedia</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-red">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="fa fa-bookmark fa-5x"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div
class="huge"><?=$meja_terpakai;?></div>
                    <div>Meja Terpakai</div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-md-6">
        <div class="panel panel-info">
            <div class="panel-heading">
                <i class="fa fa-comment fa-fw"></i> Pesanan
            </div>
            <!-- /.panel-heading -->
            <div class="panel-body">
                <div class="list-group">
                    <?php
                    $total_pesanan = array();
                    if($list_pesanan)
                    {
                        foreach($list_pesanan as
$sp)
                        {
                            $total_pesanan[] =
$sp['total_harga'];
                        }
                    }
                    <a
href="<?=site_url('transaksi');?>?kode_transaksi=<?=$sp['kode_trans
aksi'];?>" class="list-group-item">
                        <i class="fa fa-comment fa-fw"></i>
                        <?=$sp['nama_meja'];?> : <?=$sp['nama'];?>
                        <span class="pull-right
small"><b><?=rupiah($sp['total_harga']);?></b>
                        </span>
                    </a>
                    <?php
                    }
                </div>
                <a href="javascript:;" class="list-group-
item list-group-item-info">
                    <b>Total = <?=count($total_pesanan);?>

```

```

                <span                                class="pull-right
small"><?=rupiah(array_sum($total_pesanan));?>
                </span>
                </b>
            </a>
        </div>
        <!-- /.list-group -->
    </div>
    <!-- /.panel-body -->
    <div class="panel-footer">
        <a href="<?=site_url('transaksi');?>" class="btn
btn-default btn-block">Lihat Semua</a>
    </div>
</div>
</div>
<div class="col-md-6">
    <div class="panel panel-success">
        <div class="panel-heading">
            <i class="fa fa-shopping-cart fa-fw"></i>
Penjualan Hari ini
        </div>
        <!-- /.panel-heading -->
        <div class="panel-body">
            <div class="list-group">
                <?php
                $total_penjualan = array();
                if($list_penjualan)
                {
                    foreach($list_penjualan
as $lp)
                    {
                        $total_penjualan[]
= $lp['total_harga'];
                <?>
                <div class="list-group-item">
                    <i class="fa fa-shopping-cart fa-
fw"></i> <?=$lp['nama'];?>
                    <span                                class="pull-right
small"><b><?=rupiah($lp['total_harga']);?></b>
                    </span>
                </div>
                <?php
                    }
                }
                <?>
                <div class="list-group-item list-group-
item-success">
                    <b>Total                                =
<?=count($total_penjualan);?>
                    <span                                class="pull-right
small"><?=rupiah(array_sum($total_penjualan));?>
                    </span>
                    </b>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
</div>

```

user edit form.php

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-12">
            <h3 class="page-header">Edit Users Profile</h3>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12">
            <?php echo $errors; ?>
            <?php echo $this->session->flashdata('msg');?>
            <?php echo
form_open(current_url(),array('class'=>'form-horizontal client-
validation'));?>
                <div class="form-group">
                    <label class="control-label col-sm-
2">Username</label>
                    <div class="col-sm-10">
                        <?php echo
form_input(array('name'=>'username', 'type'=>'text', 'id'=>'username
', 'class'=>'form-control
disabled', 'value'=>${d['username']}, 'placeholder'=>'Username', 'reado
nly'=>TRUE));?>
                    </div>
                </div>
                <div class="form-group">
                    <label class="control-label col-sm-
2">Nama</label>
                    <div class="col-sm-10">
                        <?php echo
form_input(array('name'=>'name', 'type'=>'text', 'id'=>'name', 'class
'=>'form-
control', 'value'=>${d['name']}, 'placeholder'=>'Nama', 'required'=>TRU
E));?>
                    </div>
                </div>
                <div class="form-group">
                    <label class="control-label col-sm-
2">Password</label>
                    <div class="col-sm-10">
                        <?php echo
form_input(array('name'=>'password', 'type'=>'password', 'id'=>'pass
word', 'class'=>'form-
control', 'value'=>NULL, 'placeholder'=>'Password'));?>
                    </div>
                </div>
                <div class="form-group">
                    <label class="control-label col-sm-2">Konfirmasi
Password</label>
                    <div class="col-sm-10">

```

```

        <?php echo
form_input(array('name'=>'repassword','type'=>'password','id'=>'re
password','class'=>'form-
control','value'=>NULL,'placeholder'=>'Konfirmasi Password','data-
rule-equalto'=>'#password','data-fv-identical-
field'=>'password','data-fv-identical'=>'true'));?>
    </div>
</div>
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <button type="submit" class="btn
btn-primary">Simpan</button>
        </div>
    </div>
    <?php echo form_close();?>
</div>
</div>
</div>

```

daftar menu list.php

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-sm-12">
            <h3 class="page-header">Data Daftar Menu</h3>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12">
            <?php echo $this->session->flashdata('msg');?>
            <div class="row" style="margin-bottom: 10px;">
                <div class="col-sm-12">
                    <?php echo anchor('daftar_menu/add', '<i
class="fa fa-plus"></i> Tambah',array('class'=>'btn btn-
primary'))?>
                </div>
            </div>
        </div>
    </div>
    <div class="col-sm-12">
        <div class="table-responsive">
            <table class="table table-bordered table-hover
datatable table-striped">
                <colgroup>
                    <col class="col-xs-1">
                        <col>
                        <col>
                        <col>
                        <col>
                        <col>
                    </colgroup>
                <thead>
                    <tr>
                        <th>No.</th>
                        <th>Nama Menu</th>
                        <th>Kategori</th>
                        <th>Harga</th>

```



```

                <th>Tgl Dibuat</th>
                <th>User</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $i = $offset + 1;
                foreach ($data as $d)
                {
                    ?>
                    <tr>
                        <td class="text-center"><?php
echo $i;?></td>
                        <td><?php echo
$d['nama_menu'];?></td>
                        <td><?php echo
$d['nama_kategori'];?></td>
                        <td><?php echo
rupiah($d['harga']);?></td>
                        <td><?php echo
indo_date($d['tgl_dibuat'],TRUE,TRUE);?></td>
                        <td><?php echo
$d['name'];?></td>
                        <td class="text-
center"><?php echo
anchor("daftar_menu/edit/".$d['id_daftar_menu'], '<i class="fa fa-
pencil"></i>',array('class'=>'btn btn-sm btn-info','data-
toggle'=>'tooltips','title'=>'Edit')) ;?> <?php echo
anchor("daftar_menu/delete/".$d['id_daftar_menu'], '<i class="fa
fa-times"></i>',array('class'=>'btn btn-sm btn-danger','data-
toggle'=>'tooltips','title'=>'Delete','onclick'=>'return
confirm('\Yakin akan Menghapus?');')) ;?></td>
                    </tr>
                    <?php $i++; }?>
                </tbody>
            </table>
        </div>
    </div>
</div>
</div>
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal"
aria-label="Close"><span aria-
hidden="true">&times;</span></button>
                <h4 class="modal-title" id="myModalLabel">Preview
Daftar_menu</h4>
            </div>
            <div class="modal-body">
            </div>
        </div>
    </div>

```

```

    </div>
  </div>
</div>

```

form.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-12">
      <h3 class="page-header">Laporan Penjualan</h3>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12">
      <?php echo $errors; ?>
      <?php echo
form_open('laporan_penjualan/cetak',array('class'=>'form-
horizontal client-validation','target'=>'_blank'));?>
      <div class="form-group">
        <label class="control-label col-sm-
2">Periode</label>
        <div class="col-md-2">
          <?php echo
form_input(array('name'=>'dari','id'=>'dari','class'=>'form-
control
datetimepicker','placeholder'=>'Dari','required'=>TRUE));?>
        </div>
        <div class="col-md-1 text-center">
          <i class="fa fa-minus"></i>
        </div>
        <div class="col-md-2">
          <?php echo
form_input(array('name'=>'sampai','id'=>'sampai','class'=>'form-
control
datetimepicker','placeholder'=>'Sampai','required'=>TRUE));?>
        </div>
      </div>
      <div class="form-group">
        <label class="control-label col-sm-2">Jenis
Laporan</label>
        <div class="col-sm-10">
          <?php echo
form_dropdown('jenis_laporan',array('rekap'=>'Rekap
Penjualan','detail'=>'Detail Penjualan','menu'=>'Rekap Penjualan
Per Menu'),NULL,'class="form-control"');?>
        </div>
      </div>
      <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
          <button type="submit" class="btn
btn-primary"><i class="fa fa-print"></i> Cetak</button>
        </div>
      </div>
      <?php echo form_close();?>
    </div>
  </div>
</div>

```

meja list.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-sm-12">
      <h3 class="page-header">Data Meja</h3>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12">
      <?php echo $this->session->flashdata('msg');?>
      <div class="row" style="margin-bottom: 10px;">
        <div class="col-sm-12">
          <?php echo anchor('meja/add', '<i
class="fa fa-plus"></i> Tambah',array('class'=>'btn btn-
primary'))?>
        </div>
      </div>
    </div>
    <div class="col-sm-12">
      <div class="table-responsive">
        <table class="table table-bordered table-hover
datatable table-striped">
          <colgroup>
            <col class="col-xs-1">
              <col>
              <col>
              <col>
              <col class="col-xs-1">
            </colgroup>
            <thead>
              <tr>
                <th>No.</th>
                <th>Nama Meja</th>
                <th>Tgl Dibuat</th>
                <th>User</th>
                <th>Action</th>
              </tr>
            </thead>
            <tbody>
              <?php
                $i = $offset + 1;
                foreach ($data as $d)
                {
                  ?>
                  <tr>
                    <td class="text-center"><?php
echo $i;?></td>
                    <td><?php echo
$d['nama_meja'];?></td>
                    <td><?php echo
indo_date($d['tgl_dibuat'],TRUE,TRUE);?></td>
                    <td><?php echo
$d['name'];?></td>
                    <td class="text-
center"><?php echo anchor("meja/edit/".$d['id_meja'], '<i
class="fa fa-pencil"></i>',array('class'=>'btn btn-sm btn-

```

```

info', 'data-toggle'=>'tooltips', 'title'=>'Edit')) ;?> <?php echo
anchor("meja/delete/" . $d['id_meja'], '<i class="fa fa-
times"></i>', array('class'=>'btn btn-sm btn-danger', 'data-
toggle'=>'tooltips', 'title'=>'Delete', 'onclick'=>'return
confirm(\ 'Yakin akan Menghapus? \ ');')) ;?></td>
</tr>
<?php $i++; }?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>

```

```

<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
aria-label="Close"><span aria-
hidden="true">&times;</span></button>
        <h4 class="modal-title" id="myModalLabel">Preview
Meja</h4>
      </div>
      <div class="modal-body">
      </div>
    </div>
  </div>
</div>
</div>

```

menu kategori list.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-sm-12">
      <h3 class="page-header">Data Kategori Menu</h3>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12">
      <?php echo $this->session->flashdata('msg');?>
      <div class="row" style="margin-bottom: 10px;">
        <div class="col-sm-12">
          <?php echo anchor('menu_kategori/add', '<i
class="fa fa-plus"></i> Tambah', array('class'=>'btn btn-
primary'))?>
        </div>
      </div>
    </div>
  </div>
  <div class="col-sm-12">
    <div class="table-responsive">
      <table class="table table-bordered table-hover
datatable table-striped">
        <colgroup>
          <col class="col-xs-1">

```

```

        <col>
        <col>
        <col>
        <col class="col-xs-1">
    </colgroup>
    <thead>
        <tr>
            <th>No.</th>
            <th>Nama Kategori</th>
            <th>Tgl Dibuat</th>
            <th>User</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        <?php
        $i = $offset + 1;
        foreach ($data as $d)
        {
            ?>
            <tr>
                <td class="text-center"><?php
echo $i;?></td>
                <td><?php echo
$d['nama_kategori'];?></td>
                <td><?php echo
indo_date($d['tgl_dibuat'],TRUE,TRUE);?></td>
                <td><?php echo
$d['name'];?></td>
                <td class="text-
center"><?php echo
anchor("menu_kategori/edit/".$d['id_menu_kategori'], '<i class="fa
fa-pencil"></i>',array('class'=>'btn btn-sm btn-info','data-
toggle'=>'tooltips','title'=>'Edit')) ;?> <?php echo
anchor("menu_kategori/delete/".$d['id_menu_kategori'], '<i
class="fa fa-times"></i>',array('class'=>'btn btn-sm btn-
danger','data-
toggle'=>'tooltips','title'=>'Delete','onclick'=>'return
confirm('\Yakin akan Menghapus?');')) ;?></td>
            </tr>
            <?php $i++; }?>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">

```

```

        <button type="button" class="close" data-dismiss="modal"
        aria-label="Close"><span aria-
        hidden="true">&times;</span></button>
        <h4 class="modal-title" id="myModalLabel">Preview
        Menu_kategori</h4>
    </div>
    <div class="modal-body">
    </div>
</div>
</div>
</div>

```

roles list.php

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-sm-12">
            <h3 class="page-header">Data Level User</h3>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12">
            <?php echo $this->session->flashdata('msg');?>
            <div class="row" style="margin-bottom: 10px;">
                <div class="col-sm-12">
                    <?php echo anchor('roles/add', '<i
                    class="fa fa-plus"></i> Tambah', array('class'=>'btn btn-
                    primary'))?>
                </div>
            </div>
        </div>
    </div>
    <div class="col-sm-12">
        <div class="table-responsive">
            <table class="table table-bordered table-hover
            datatable table-striped">
                <colgroup>
                    <col class="col-xs-1">
                        <col>
                            <col class="col-xs-1">
                                </colgroup>
                                <thead>
                                    <tr>
                                        <th>No.</th>
                                        <th>Level</th>
                                        <th>Action</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    <?php
                                    $i = $offset + 1;
                                    foreach ($data as $d)
                                    {
                                        ?>
                                        <tr>
                                            <td class="text-center"><?php
echo $i;?></td>

```

```
 <?php echo $d['role'];?></td>  view.php <div id="page-wrapper"> <div class="row"> <div class="col-sm-12"> <h3 class="page-header">Kasir</h3> </div> </div> <div class="row"> <div class="col-md-12"> <?php echo $this->session->flashdata('msg');?> </div> <div class="col-md-6"> <div class="table-responsive"> <h5>Daftar Pesanan Hari ini</h5> <table class="table table-bordered table-hover datatable table-striped"> <colgroup> <col> <col> <col> | |
```

```

        <col>
        <col>
        <col class="col-xs-1">
</colgroup>
    <thead>
        <tr>
            <th>No.</th>
            <th>Meja</th>
            <th>Nama</th>
            <th>Total</th>
            <th>Status</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        <?php
        if($data)
        {
            $i=1;
            foreach ($data as $d)
            {
                $status_color = NULL;
                if($d['status'] ==
'lunas')
                {
                    $status_color =
'success';
                }
                elseif($d['status'] ==
'selesai')
                {
                    $status_color =
'info';
                }
                elseif($d['status'] ==
'pesan')
                {
                    $status_color =
'warning';
                }
                else
                {
                    $status_color =
'danger';
                }
            }
        }
        ?>
        <tr>
            <td class="text-
center"><?=$i;?></td>
            <td><?=$d['nama_meja'];?></td>
            <td><?=$d['nama'];?></span></td>
            <td class="text-
right"><b><?=$rupiah($d['total_harga']);?></td>

```



```

                                <td class="text-
center"><span class="label label-
<?=$status_color;?>"><?=ucfirst($d['status']);?></span></span></td
>
                                <td class="text-center">
                                <?php
                                if($d['status'] ==
'lunas')
                                {
                                    //echo
                                anchor("transaksi/cetak_ulang?kode_transaksi=". $d['kode_transaksi']
                                ], '<i class="fa fa-print"></i>', array('class'=>'btn btn-sm btn-
                                info tooltips', 'title'=>'Cetak Ulang'));
                                }
                                else
                                {
                                    echo
                                anchor("transaksi?kode_transaksi=". $d['kode_transaksi'], '<i
                                class="fa fa-money"></i>', array('class'=>'btn btn-sm btn-warning
                                tooltips', 'title'=>'Bayar'));
                                }
                                ?>
                                </td>
                                </tr>
                                <?php
                                $i++;}
                                } ?>
                                </tbody>
                                </table>
                                </div>
                                </div>
                                <div class="col-md-6">
                                <?php echo
                                form_open('transaksi/billing', array('class'=>', 'id'=>'form_bayar'
                                , 'name'=>'form', 'autocomplete'=>'off', 'data-msg'=>'Apakah Anda
                                Yakin?'));?>
                                <div class="table-responsive">
                                <table class="table table-bordered">
                                <tbody>
                                <tr>
                                    <th>Nama</th>
                                    <td>
                                        <input
                                type="text" name="nama" id="nama" class="form-control"
                                placeholder="Nama Pemesan" value="<?=$dp['nama'];?>" required />
                                    </td>
                                </tr>
                                <tr>
                                    <th>Pilih
                                Meja</th>
                                    <td>
                                        <div
                                class="checkbox">
                                        <?php
                                if($meja_dipesan)

```

```

{
?>

<label>

    <input type="radio" name="id_meja" class="meja"
value="<?=$meja_dipesan['id_meja'];?>" checked />
<?=$meja_dipesan['nama_meja'];?>

    <input type="hidden"
name="nama_meja[<?=$meja_dipesan['id_meja'];?>]"
value="<?=$meja_dipesan['nama_meja'];?>" />

</label>

<?php
}
?>
<?php
//echo
form_hidden('nama_meja',$meja_dipesan['nama_meja']);
//echo
form_hidden('id_meja',$meja_dipesan['id_meja']);

if($meja)
{

foreach($meja as $i=>$m)

{
?>

<label>

    <input type="radio" name="id_meja" class="meja"
value="<?=$i;?>" /> <?=$m;?>

    <input type="hidden" name="nama_meja[<?=$i;?>]"
value="<?=$m;?>" />

</label>

<?php

}
?>
</div>
</td>
</tr>
<tr>
<td colspan="2">
<?php echo
form_dropdown('id_daftar_menu_option',$daftar_menu,NULL,'id="menu"
class="form-control select2" data-tags="true" data-
placeholder="Cari Menu");?>

```

```



```

```

                                <td class="text-
left"><input type="hidden" name="harga[]" class="harga"
value="<?=$ddpd['harga'];?>" /><span class="pull-
right"><?=$ddpd['harga'];?></span></td>
                                <td><div
class="form-
group"><?=form_input(array('type'=>'number', 'min'=>'1', 'readonly'=
>true, 'value'=>$ddpd['jumlah_item']*$ddpd['harga'], 'name'=>'sub_to
tal[]', 'required'=>TRUE, 'class'=>'form-control input-sm
sub_total'));?></div></td>
                                <td><a
href="javascript:;" class="btn btn-danger btn-sm delete-order"><i
class="fa fa-times"></i></a></td>
                                </tr>
                                <?php
                                }
                                ?>
                                </tbody>
                                <tfoot>
                                <tr>
                                <td
colspan="2"><h3>Total</h3></td>
                                <td colspan="3"
class="text-left">
                                <input
type="hidden" name="total" id="total"
value="<?=$dp['total_harga'];?>" />
                                <h3><strong>Rp <span id="total_display" class="pull-
right"><?=rupiah($dp['total_harga'], FALSE);?></span></strong></h3>
                                </td>
                                </tr>
                                <tr>
                                <td
colspan="2"><h4>Bayar</h4></td>
                                <td colspan="3">
                                <input
type="number" name="cash" id="cash" class="form-control input-lg
text-right" />
                                </td>
                                </tr>
                                <tr>
                                <td
colspan="2"><h3>Kembali</h3></td>
                                <td colspan="3"
class="text-left">
                                <h3><strong>Rp <span id="pengembalian" class="pull-
right">0</span></strong></h3>
                                </td>
                                </tr>
                                <tr>
                                <td colspan="5"
class="text-right">

```

```



```

preview struk.php

```

<div class="modal-body" id="struk">
  <h4 class="text-center">RM Dapur Jawa</h4>
  <p class="text-center">Jl. MP Mangkunegara No. 16<br />085268781693</p>
  <table class="table table-show-struk">
    <colgroup>
      <col class="col-xs-6">
      <col class="col-xs-3">
      <col class="col-xs-3">
    </colgroup>
    <tbody>
      <tr>
        <td colspan="3">
          Nama : <?=$nama; ?>
          <?php
            if($meja)
            {
              ?>
              <br />Meja : <?=$meja; ?>
              <?php } ?>
        </td>
      </tr>
      <tr>
        <td colspan="3">Pesanan</td>
      </tr>
      <?php
        foreach($data as $k=>$d)

```

```

        {
            $border_top = NULL;
            if ($k==0)
            {
                $border_top = 'class="border-top"';
            }
        }
    ?>
    <tr <?=$border_top;?>
        <td><?=$d['nama_menu'];?></td>
        <td><?=$d['jumlah_item'];?> X
    <?=$rupiah($d['harga']);?></td>
        <td class="text-
right"><?=$rupiah($d['sub_total']);?></td>
    </tr>
    <?php
    }
    ?>
    <tr class="border-top">
        <td colspan="2" class="text-right">Total
    &nbsp;&nbsp;&nbsp;&nbsp;</td>
        <td class="text-
right"><h4><?=$rupiah($total);?></h4></td>
    </tr>
    <tr>
        <td colspan="2" class="text-right">Bayar
    &nbsp;&nbsp;&nbsp;&nbsp;</td>
        <td class="text-
right"><?=$rupiah($bayar);?></td>
    </tr>
    <tr>
        <td colspan="2" class="text-right">Kembali
    &nbsp;&nbsp;&nbsp;&nbsp;</td>
        <td class="text-right border-top"><?php
    $pengembalian = $bayar-$total;
    echo rupiah($pengembalian);
    ?></td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <td colspan="3" class="text-center">Terima
    Kasih<br/>Selamat Menikmati</td>
    </tr>
    <tr class="border-top">
        <td colspan="3" class="text-
center"><?=$userdata->name;?>, <?=$indo_date(date('Y-m-d
H:i:s'), TRUE, TRUE);?></td>
    </tr>
</tfoot>
</table>
</div>
<div class="modal-footer">
<a href="javascript:;" class="btn btn-primary" id="print-struk"
data-url="<?=$site_url('print_struks/kantin');?>" id-
transaksi="<?=$id_transaksi;?>" data-

```

```

redirect="<?=@site_url('transaksi');?>"<i class="fa fa-print"></i>
Cetak Struk</a>
<a href="<?=@site_url('transaksi');?>" type="button" class="btn
btn-default cancel-print-struk">Batal Cetak Struk</a>
</div>

```

users list.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-sm-12">
      <h3 class="page-header">Data Pengguna</h3>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12">
      <?php echo $this->session->flashdata('msg');?>
      <div class="row" style="margin-bottom: 10px;">
        <div class="col-sm-12">
          <?php echo anchor('user/add', '<i
class="fa fa-plus"></i> Tambah',array('class'=>'btn btn-
primary'))?>
        </div>
      </div>
    </div>
  </div>
  <div class="col-sm-12">
    <div class="table-responsive">
      <table class="table table-bordered table-hover
datatable table-striped">
        <colgroup>
          <col class="col-xs-1">
            <col>
            <col>
            <col>
            <col>
            <col>
            <col class="col-xs-1">
          </colgroup>
          <thead>
            <tr>
              <th>No.</th>
              <th>Username</th>
              <th>Nama</th>
              <th>Level</th>
              <th>Tgl Dibuat</th>
              <th>Terkahir Login</th>
              <th>Action</th>
            </tr>
          </thead>
          <tbody>
            <?php
            $i = $offset + 1;
            foreach ($data as $d)
            {
              ?>
            <tr>

```

```

                                <td class="text-center"><?php
echo $i;?></td>
                                <td><?php echo
                                <td><?php echo
                                <td><?php echo
                                <td><?php echo
                                <td><?php echo
                                <td><?php echo
                                <td class="text-
                                center"><?php echo anchor("user/edit/".$d['user_id'], '
                                class="fa fa-pencil"></i>', array('class'=>'btn btn-sm btn-
                                info', 'data-toggle'=>'tooltips', 'title'=>'Edit')) ;?> <?php echo
                                anchor("user/delete/".$d['user_id'], '
                                class="fa fa-
                                times"></i>', array('class'=>'btn btn-sm btn-danger', 'data-
                                toggle'=>'tooltips', 'title'=>'Delete', 'onclick'=>'return
                                confirm('\Yakin akan Menghapus?');')) ;?></td>
                                </tr>
                                <?php $i++; }?>
                                </tbody>
                                </table>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
aria-label="Close"><span aria-
hidden="true">&times;</span></button>
        <h4 class="modal-title" id="myModalLabel">Preview
User</h4>
      </div>
      <div class="modal-body">
      </div>
    </div>
  </div>
</div>
</div>

```