

LAMPIRAN

AdapterSubMenu

```
import androidx.annotation.NonNull;

import
androidx.recyclerview.widget.Recycl
erView;
```

```
import
com.kelas_b.b220280039.bahasakore
a.R;

import
com.kelas_b.b220280039.bahasakore
a.koneksi.StrukturData;
```

```
import java.util.List;
```

```
public class AdapterSubMenu
extends
RecyclerView.Adapter<AdapterSub
Menu.RecyclerViewAdapter> {
```

```
    private final Context context;

    private final List<StrukturData>
strukturData;

    private final ItemClickListener
itemClickListener;
```

```
    public AdapterSubMenu(Context
context, List<StrukturData>
strukturData, ItemClickListener
itemClickListener) {
```

```
        this.context = context;

        this.strukturData = strukturData;
```

```
        this.itemClickListener =
itemClickListener;
```

```
    }
```

```
    @NonNull
```

```
    @Override
```

```
    public RecyclerViewAdapter
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
```

```
        View view =
LayoutInflater.from(context).inflate(
R.layout.item_sub_menu, parent,
false);
```

```
        return new
RecyclerViewAdapter(view,
itemClickListener);
```

```
    }
```

```
    @Override
```

```
    public void
onBindViewHolder(@NonNull final
RecyclerViewAdapter holder, int
position) {
```

```
        StrukturData datanya =
strukturData.get(position);
```

```
        if (datanya.getKategoriHuruf()
!= null &&
!datanya.getKategoriHuruf().isEmpty
()) {
```

```
            holder.textsubmenu.setText(datanya.
getKategoriHuruf());
```

```
        } else if
(datanya.getKategoriBilangan() !=
null &&
```

```
!datanya.getKategoriBilangan().isEmpty() {
```

```
holder.textsubmenu.setText(datanya.getKategoriBilangan());
```

```
    } else if
    (datanya.getKategoriKosakata() !=
    null &&
    !datanya.getKategoriKosakata().isEmpty()) {
```

```
holder.textsubmenu.setText(datanya.getKategoriKosakata());
```

```
    }
}
```

```
@Override
```

```
public int getItemCount() {
    return strukturData.size();
}
```

```
static class RecyclerViewAdapter
extends RecyclerView.ViewHolder
implements View.OnClickListener{
```

```
    LinearLayout linearsubmenu;
```

```
    TextView textsubmenu;
```

```
    ItemClickListener
itemClickListener;
```

```
    RecyclerViewAdapter(View
itemView, ItemClickListener
itemClickListener) {
```

```
super(itemView);
```

```
        linearsubmenu =
itemView.findViewById(R.id.linearsubmenu);
```

```
        textsubmenu =
itemView.findViewById(R.id.tvsubmenu);
```

```
        this.itemClickListener =
itemClickListener;
```

```
        linearsubmenu.setOnClickListener(this);
```

```
    }
```

```
@Override
```

```
public void onClick(View v) {
```

```
    itemClickListener.OnItemClick(v,
getAdapterPosition());
```

```
    }
```

```
}
```

```
public interface
ItemClickListener{
```

```
    void OnItemClick(View view,
int position);
```

```
    }
```

```
}
```

AdapterTampil

```

public class AdapterTampil extends
RecyclerView.Adapter<AdapterTam
pil.ViewHolder> {

    private final List<StrukturData>
resultDataList;

    private final Context context;

    private final ItemClickListener
itemClickListener;

    public AdapterTampil(Context
context, List<StrukturData>
resultDataList,
AdapterTampil.ItemClickListener
itemClickListener) {

        this.context = context;

        this.resultDataList =
resultDataList;

        this.itemClickListener =
itemClickListener;

    }

    @NonNull

    @Override

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

        View view =
LayoutInflater.from(parent.getConte
xt()).inflate(R.layout.item_tampil,
parent, false);

        return new ViewHolder(view,
itemClickListener);

    }

    @Override

    public void
onBindViewHolder(@NonNull
AdapterTampil.ViewHolder holder,
int position) {

        StrukturData strukturData =
resultDataList.get(position);

        if
(strukturData.getKategoriHuruf() !=
null &&
!strukturData.getKategoriHuruf().isE
mpty()) {

            if
(strukturData.getKategoriHuruf().equ
als("Ganda")) {

                holder.imgbhs.setVisibility(View.VIS
IBLE);

                holder.textbhs.setVisibility(View.VIS
IBLE);

                Glide.with(context).load(BASE_UR
L + "huruf/ganda/" +
strukturData.getGambar()).skipMem
oryCache(true).diskCacheStrategy(D
iskCacheStrategy.NONE).into(holder
.imgbhs);

                holder.textbhs.setText(strukturData.g
etTulisanKorea()+"\n("+strukturData
.getUcapan()+")");

                holder.textarti.setVisibility(View.GO
NE);

```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);

        } else if
(strukturData.getKategoriHuruf().equals("Tunggal")) {
```

```
holder.imgbhs.setVisibility(View.VISIBLE);
```

```
holder.textbhs.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "huruf/tunggal/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.GONE);
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);
```

```
        } else if
(strukturData.getKategoriHuruf().equals("Vokal")) {
```

```
holder.imgbhs.setVisibility(View.VISIBLE);
```

```
holder.textbhs.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "huruf/vokal/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.GONE);
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```

holder.tvartikor.setVisibility(View.GONE);
    } else if
(strukturData.getKategoriHuruf().equals("Kalimat")) {

holder.tvtulkor.setVisibility(View.VISIBLE);

holder.tvlatinkor.setVisibility(View.VISIBLE);

holder.tvartikor.setVisibility(View.VISIBLE);

holder.tvtulkor.setText(strukturData.getTulisanKorea());

holder.tvlatinkor.setText("(" + strukturData.getUcapan() + ")");

holder.tvartikor.setText(strukturData.getArti());

holder.imgbhs.setVisibility(View.GONE);

holder.textbhs.setVisibility(View.GONE);

holder.textarti.setVisibility(View.GONE);
    }
    } else if
(strukturData.getKategoriBilangan()
!= null &&
!strukturData.getKategoriBilangan().isEmpty()) {
    if
(strukturData.getKategoriBilangan().equals("Asli")) {

holder.imgbhs.setVisibility(View.VISIBLE);

holder.textbhs.setVisibility(View.VISIBLE);

Glide.with(context).load(BASE_URL + "bilangan/asli/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder.imgbhs);

holder.textbhs.setText(strukturData.getTulisanKorea() + "\n(" + strukturData.getUcapan() + ")");

holder.textarti.setVisibility(View.GONE);

holder.tvtulkor.setVisibility(View.GONE);

holder.tvlatinkor.setVisibility(View.GONE);

holder.tvartikor.setVisibility(View.GONE);

```

```

        } else if
(strukturData.getKategoriBilangan().
equals("Sino")) {

holder.imgbhs.setVisibility(View.VIS
IBLE);

holder.tvlatinkor.setVisibility(View.V
ISIBLE);

holder.tvartikor.setVisibility(View.VI
SIBLE);

Glide.with(context).load(BASE_UR
L + "bilangan/sino/" +
strukturData.getGambar()).skipMem
oryCache(true).diskCacheStrategy(D
iskCacheStrategy.NONE).into(holder
.imgbhs);

holder.textbhs.setText(strukturData.g
etTulisanKorea()+"\n("+strukturData
.getUcapan()+")");

holder.textarti.setVisibility(View.GO
NE);

holder.tvtulkor.setVisibility(View.GO
NE);

holder.tvlatinkor.setVisibility(View.G
ONE);

holder.tvartikor.setVisibility(View.G
ONE);

holder.textarti.setVisibility(View.GO
NE);

} else if
(strukturData.getKategoriBilangan().
equals("Kalimat")) {

holder.tvtulkor.setVisibility(View.VI
SIBLE);

holder.tvlatinkor.setVisibility(View.V
ISIBLE);

holder.tvartikor.setVisibility(View.VI
SIBLE);

holder.tvtulkor.setText(strukturData.
getTulisanKorea());

holder.tvlatinkor.setText("("+struktur
Data.getUcapan()+")");

holder.tvartikor.setText(strukturData.
getAngka());

holder.imgbhs.setVisibility(View.GO
NE);

holder.textbhs.setVisibility(View.GO
NE);

holder.textarti.setVisibility(View.GO
NE);

}

} else if
(strukturData.getKategoriKosakata()
!= null &&
!strukturData.getKategoriKosakata().
isEmpty()) {

switch
(strukturData.getKategoriKosakata())
{

```

case "Buah":

```
holder.imgbhs.setVisibility(View.VISIBLE);
```

```
holder.textbhs.setVisibility(View.VISIBLE);
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "kosakata/buah/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
holder.textarti.setText(strukturData.getKosakata());
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);
```

break;

case "Hewan":

```
holder.imgbhs.setVisibility(View.VISIBLE);
```

```
holder.textbhs.setVisibility(View.VISIBLE);
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "kosakata/hewan/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
holder.textarti.setText(strukturData.getKosakata());
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);
```

break;


```

        case "Keluarga":

            Glide.with(context).load(BASE_URL + "kosakata/keluarga/" +
            strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
            .imgbhs);

            holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
            .getUcapan()+")");

            holder.textarti.setVisibility(View.VISIBLE);

            holder.textarti.setText(strukturData.getKosakata());

            holder.tvtulkor.setVisibility(View.GONE);

            holder.tvlatinkor.setVisibility(View.GONE);

            holder.tvartikor.setVisibility(View.GONE);

            break;

        case "Negara":

            holder.imgbhs.setVisibility(View.VISIBLE);

            holder.textbhs.setVisibility(View.VISIBLE);

            holder.textarti.setVisibility(View.VISIBLE);

            Glide.with(context).load(BASE_URL + "kosakata/negara/" +
            strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
            .imgbhs);

            holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
            .getUcapan()+")");

            holder.textarti.setVisibility(View.VISIBLE);

            holder.textarti.setText(strukturData.getKosakata());

            holder.tvtulkor.setVisibility(View.GONE);

            holder.tvlatinkor.setVisibility(View.GONE);

            holder.tvartikor.setVisibility(View.GONE);

            break;

        case "Profesi":

            holder.imgbhs.setVisibility(View.VISIBLE);

            holder.textbhs.setVisibility(View.VISIBLE);

```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "kosakata/profesi/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
holder.textarti.setText(strukturData.getKosakata());
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);
```

```
break;
case "Warna":
```

```
holder.imgbhs.setVisibility(View.VISIBLE);
```

```
holder.textbhs.setVisibility(View.VISIBLE);
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
Glide.with(context).load(BASE_URL + "kosakata/warna/" +
strukturData.getGambar()).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(holder
.imgbhs);
```

```
holder.textbhs.setText(strukturData.getTulisanKorea()+"\n("+strukturData
.getUcapan()+")");
```

```
holder.textarti.setVisibility(View.VISIBLE);
```

```
holder.textarti.setText(strukturData.getKosakata());
```

```
holder.tvtulkor.setVisibility(View.GONE);
```

```
holder.tvlatinkor.setVisibility(View.GONE);
```

```
holder.tvartikor.setVisibility(View.GONE);
```

```
break;
case "Kalimat":
```

```
holder.tvtulkor.setVisibility(View.VISIBLE);
```

```
holder.tvlatinkor.setVisibility(View.VISIBLE);
```

```
holder.tvartikor.setVisibility(View.VI  
SIBLE);
```

```
holder.tvtulkor.setText(strukturData.  
getTulisanKorea());
```

```
holder.tvlatinkor.setText("(" + struktur  
Data.getUcapan() + ")");
```

```
holder.tvartikor.setText(strukturData.  
getKosakata());
```

```
holder.imgbhs.setVisibility(View.GO  
NE);
```

```
holder.textbhs.setVisibility(View.GO  
NE);
```

```
holder.textarti.setVisibility(View.GO  
NE);
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
@Override
```

```
public int getItemCount() {
```

```
return resultDataList.size();
```

```
}
```

```
public static class ViewHolder  
extends RecyclerView.ViewHolder  
implements View.OnClickListener {
```

```
LinearLayout linearbhs;
```

```
ShapeableImageView imgbhs;
```

```
TextView textbhs, textarti,  
tvulkor, tvlatinkor, tvartikor;
```

```
CardView cardkalimat;
```

```
ItemClickListener  
itemClickListener;
```

```
public ViewHolder(View  
itemView, ItemClickListener  
itemClickListener) {
```

```
super(itemView);
```

```
linearbhs =  
itemView.findViewById(R.id.linearb  
hs);
```

```
imgbhs =  
itemView.findViewById(R.id.imgbhs  
);
```

```
textbhs =  
itemView.findViewById(R.id.textbhs  
);
```

```
textarti =  
itemView.findViewById(R.id.textarti  
);
```

```
cardkalimat =  
itemView.findViewById(R.id.cardkal  
imat);
```

```
tvulkor =  
itemView.findViewById(R.id.tvtulko  
r);
```

```

        tvlatinkor =
itemView.findViewById(R.id.tvlatin
kor);

        tvartikor =
itemView.findViewById(R.id.tvartik
or);

        this.itemClickListener =
itemClickListener;

linearbhs.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {

itemClickListener.OnItemClickBaha
sa(v, getAdapterPosition());
    }
}

public interface ItemClickListener
{
    void OnItemClickBahasa(View
view, int position);
}
}

AwalActivity

public class AwalActivity extends
AppCompatActivity {

        private boolean klikduakali = false;

        @Override
        protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_awa
l);

        ImageView btnMulai =
findViewById(R.id.btnmulai);

        Animation slideAnimation =
AnimationUtils.loadAnimation(this,
R.anim.slide_up_down);

        btnMulai.startAnimation(slideAnima
tion);

        btnMulai.setOnClickListener(v -
> {

            startActivity(new
Intent(AwalActivity.this,
PengantarActivity.class));

        });
    }

    @Override
    public void onBackPressed() {

        if (klikduakali) {

            super.onBackPressed();

            return;

```

```

    }

    klikduakali = true;

    Toast.makeText(this, "Tekan
    sekali lagi untuk keluar.",
    Toast.LENGTH_SHORT).show();

    new Handler().postDelayed(() -
    > klikduakali = false, 2000);

    }
}

```

BatchimActivity

```

public class BatchimActivity extends
AppCompatActivity {

```

```

    TextView tvbatchim;

```

```

    @Override

```

```

    protected void onCreate(Bundle
    savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_ba
        tchim);

```

```

        tvbatchim =
        findViewById(R.id.textbatchim);

```

```

        submenubatchim();

```

```

        Toolbar toolbar =
        findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

```

```

        ActionBar mAction =
        getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new
        TextView(getApplicationContext());

        titleTextView.setText("Materi
        Batchim".toUpperCase());

        titleTextView.setTextSize(20);

```

```

        titleTextView.setTextColor(Color.W
        HITE);

```

```

        titleTextView.setGravity(Gravity.CE
        NTER);

```

```

        Typeface customFont =
        ResourcesCompat.getFont(this,
        R.font.chewy);

```

```

        titleTextView.setTypeface(customFo
        nt);

```

```

        mAction.setDisplayOptions(ActionB
        ar.DISPLAY_SHOW_CUSTOM);

```

```

        mAction.setCustomView(titleTextVi
        ew, new
        ActionBar.LayoutParams(ActionBar.
        LayoutParams.WRAP_CONTENT,
        ActionBar.LayoutParams.WRAP_C
        ONTENT, Gravity.CENTER));

```

```

        mAction.setDisplayHomeAsUpEnabled(true);

```

```

        toolbar.setNavigationOnClickListener(v -> onBackPressed());

```

```

    }

    @Override
    public boolean
    onOptionsItemSelected(MenuItem
    item) {

        if (item.getItemId() ==
        android.R.id.home) {

            onBackPressed();

        }

        return
        super.onOptionsItemSelected(item);
    }

    private void submenubatchim() {

        ApiInterface apiInterface =
        ApiClient.getClient().create(ApiI
        nterface.class);

        Call<List<StrukturData>> call
        = apiInterface.getBatchim();

        call.enqueue(new
        Callback<List<StrukturData>>() {

            @Override

            public void
            onResponse(@NonNull
            Call<List<StrukturData>> call,
            @NonNull
            Response<List<StrukturData>>
            response) {

                if(response.isSuccessful()
                && response.body() != null){

```

```

                List<StrukturData>
                dataModels = response.body();

                StrukturData result =
                dataModels.get(0);

                String htmlContent =
                result.getPenjelasan();

                tvbatchim.setText(Html.fromHtml(ht
                mlContent,
                Html.FROM_HTML_MODE_LEG
                ACY, null, new HrTagHandler()));

                }else{

                Toast.makeText(BatchimActivity.this
                , "Response unsuccessful or empty",
                Toast.LENGTH_SHORT).show();

                }

                }

                @Override

                public void
                onFailure(@NonNull
                Call<List<StrukturData>> call,
                @NonNull Throwable t) {

                    Toast.makeText(BatchimActivity.this
                    , "Error occurred: " + t.getMessage(),
                    Toast.LENGTH_SHORT).show();

                    }

                    });

                    }

                    }

```

HasilActivity

```

public class HasilActivity extends
AppCompatActivity {

    TextView tvnilai;

    LinearLayout bgulangi, bgkeluar;

    String submenu,kate,nilai;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_ha
sil);

        Intent get = getIntent();

        submenu =
get.getStringExtra("SubMenu");

        kate =
get.getStringExtra("Kategori");

        nilai =
get.getStringExtra("Hasil");

        if (nilai == null) {
            nilai = "0";
        }

        tvnilai =
findViewById(R.id.tvnilai);

        tvnilai.setText(nilai);

```

```

        bgulangi =
findViewById(R.id.bgulangi);

        bgkeluar =
findViewById(R.id.bgkeluar);

        bgulangi.setOnClickListener(v -
> {

            Intent it = new
Intent(HasilActivity.this,
LatihanActivity.class);

            it.putExtra("SubMenu",
submenu);

            it.putExtra("Kategori", kate);

            startActivity(it);

            finish();

        });

        bgkeluar.setOnClickListener(v -
> {

            Intent it;

            if (submenu.isEmpty() &&
kate.isEmpty()){

                it = new
Intent(HasilActivity.this,
MenuActivity.class);

            } else {

                it = new
Intent(HasilActivity.this,
TampilActivity.class);

            }

            it.putExtra("SubMenu",
submenu);

            it.putExtra("KategoriSubMenu",
kate);

```

```

        startActivity(it);
        finish();
    });

    Toolbar toolbar =
        findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    ActionBar mAction =
        getSupportActionBar();
    assert mAction != null;

    TextView titleTextView = new
        TextView(getApplicationContext());
    if (submenu.isEmpty() &&
        kate.isEmpty()){
        titleTextView.setText("Hasil
        Latihan".toUpperCase());
    } else {
        titleTextView.setText("Hasil
        Latihan".toUpperCase() + " " +
        submenu.toUpperCase());
    }
    titleTextView.setTextSize(20);

    titleTextView.setTextColor(Color.W
    HITE);

    titleTextView.setGravity(Gravity.CE
    NTER);

    Typeface customFont =
        ResourcesCompat.getFont(this,
        R.font.chewy);

    titleTextView.setTypeface(customFo
    nt);

    TextView subtitleTextView =
        new
        TextView(getApplicationContext());
    if (submenu.isEmpty() &&
        kate.isEmpty()){
        subtitleTextView.setText("Semua
        Kategori".toUpperCase());
    } else {
        subtitleTextView.setText(kate.toUpp
        erCase());
    }

    subtitleTextView.setTextSize(15);

    subtitleTextView.setTextColor(Color.
    WHITE);

    subtitleTextView.setGravity(Gravity.
    CENTER);

    LinearLayout titleLayout = new
        LinearLayout(getApplicationContext
        ());

    titleLayout.setOrientation(LinearLay
    out.VERTICAL);

    LinearLayout.LayoutParams
    params = new
        LinearLayout.LayoutParams(LinearL
        ayout.LayoutParams.WRAP_CONT
        ENT,

```



```
LinearLayout.LayoutParams.WRAP_
_CONTENT);
```

```
    params.gravity =
Gravity.CENTER;
```

```
titleLayout.addView(titleTextView,
params);
```

```
titleLayout.addView(subtitleTextVie
w, params);
```

```
mAction.setDisplayOptions(ActionB
ar.DISPLAY_SHOW_CUSTOM);
```

```
mAction.setCustomView(titleLayout,
new
ActionBar.LayoutParams(ActionBar.
LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_C
ONTENT, Gravity.CENTER));
    }
}
```

LatihanActivity

```
public class LatihanActivity extends
AppCompatActivity {
```

```
    ImageView imgsoal;
```

```
    TextView tvbanyaksoal, tvsoal,
tvsoal1, tvsoal2, tvsoal3, tvsoal4;
```

```
    List<StrukturData> dataModels;
```

```
    int currentQuestionIndex = 0;
```

```
    int score = 0;
```

```
    String submenu, kate;
```

```
    @Override
```

```
    protected void onCreate(Bundle
savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_lat
ihan);
```

```
        Intent get = getIntent();
```

```
        submenu =
get.getStringExtra("SubMenu");
```

```
        kate =
get.getStringExtra("Kategori");
```

```
        tvbanyaksoal =
findViewById(R.id.tvbanyaksoal);
```

```
        imgsoal =
findViewById(R.id.imgsoal);
```

```
        tvsoal =
findViewById(R.id.tvsoal);
```

```
        tvsoal1 =
findViewById(R.id.tvsoal1);
```

```
        tvsoal2 =
findViewById(R.id.tvsoal2);
```

```
        tvsoal3 =
findViewById(R.id.tvsoal3);
```

```
        tvsoal4 =
findViewById(R.id.tvsoal4);
```

```
        submenulatihan(kate);
```

```
tvsoal1.setOnClickListener(v ->
checkAnswer(tvsoal1.getText().toStri
ng()));
```

```
tvsoal2.setOnClickListener(v ->
checkAnswer(tvsoal2.getText().toStri
ng()));
```

```
tvsoal3.setOnClickListener(v ->
checkAnswer(tvsoal3.getText().toStri
ng()));
```

```
tvsoal4.setOnClickListener(v ->
checkAnswer(tvsoal4.getText().toStri
ng()));
```

```
Toolbar toolbar =
findViewById(R.id.toolbar);
```

```
setSupportActionBar(toolbar);
```

```
ActionBar mAction =
getSupportActionBar();
```

```
assert mAction != null;
```

```
TextView titleTextView = new
TextView(getApplicationContext());
```

```
if (submenu.isEmpty() &&
kate.isEmpty()){
```

```
titleTextView.setText("Latihan".toUp
perCase());
```

```
    } else {
```

```
titleTextView.setText("Latihan".toUp
perCase() + " " +
submenu.toUpperCase());
```

```
    }
```

```
titleTextView.setTextSize(20);
```

```
titleTextView.setTextColor(Color.W
HITE);
```

```
titleTextView.setGravity(Gravity.CE
NTER);
```

```
    Typeface customFont =
ResourcesCompat.getFont(this,
R.font.chewy);
```

```
titleTextView.setTypeface(customFo
nt);
```

```
    TextView subtitleTextView =
new
TextView(getApplicationContext());
```

```
    if (submenu.isEmpty() &&
kate.isEmpty()){
```

```
subtitleTextView.setText("Semua
Kategori".toUpperCase());
```

```
    } else {
```

```
subtitleTextView.setText(kate.toUpp
erCase());
```

```
    }
```

```
subtitleTextView.setTextSize(15);
```

```
subtitleTextView.setTextColor(Color.
WHITE);
```

```
subtitleTextView.setGravity(Gravity.
CENTER);
```

```
    LinearLayout titleLayout = new
LinearLayout(getApplicationContext
());
```

```
titleLayout.setOrientation(LinearLay
out.VERTICAL);
```

```
        LinearLayout.LayoutParams
params = new
LinearLayout.LayoutParams(LinearL
ayout.LayoutParams.WRAP_CONT
ENT,
LinearLayout.LayoutParams.WRAP
_CONTENT);

        params.gravity =
Gravity.CENTER;
```

```
titleLayout.addView(titleTextView,
params);
```

```
titleLayout.addView(subtitleTextVie
w, params);
```

```
mAction.setDisplayOptions(ActionB
ar.DISPLAY_SHOW_CUSTOM);
```

```
mAction.setCustomView(titleLayout,
new
ActionBar.LayoutParams(ActionBar.
LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_C
ONTENT, Gravity.CENTER));
```

```
mAction.setDisplayHomeAsUpEnabled(true);
```

```
toolbar.setNavigationOnClickListene
r(v -> onBackPressed());

}
```

```
@Override
```

```
        public boolean
onOptionsItemSelected(MenuItem
item) {

            if (item.getItemId() ==
android.R.id.home) {

                onBackPressed();

            }

            return
super.onOptionsItemSelected(item);

        }
```

```
        private void
submenulatihan(String kate) {

            ApiInterface apiInterface =
ApiClient.getClient().create(ApiI
nterface.class);
```

```
            Call<List<StrukturData>> call
= apiInterface.getLatihan(kate);

            call.enqueue(new
Callback<List<StrukturData>>() {
```

```
                @Override

                public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {

                    if (response.isSuccessful()
&& response.body() != null) {

                        dataModels =
response.body();
```

```
loadQuestion(currentQuestionIndex);
```

```

        } else {

Toast.makeText(LatihanActivity.this,
"Response unsuccessful or empty",
Toast.LENGTH_SHORT).show();

        }
    }

    @Override
    public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {

Toast.makeText(LatihanActivity.this,
"Error occurred: " + t.getMessage(),
Toast.LENGTH_SHORT).show();

        }
    });
}

private void loadQuestion(int
index) {
    if (index < dataModels.size()) {

        StrukturData question =
dataModels.get(index);

        tvbanyaksoal.setText("Soal "
+ (index + 1) + " dari " +
dataModels.size() + " Pertanyaan");

Glide.with(getApplicationContext())
        .load(BASE_URL +
"latihan/" + question.getGambar())

        .skipMemoryCache(true)

        .diskCacheStrategy(DiskCacheStrate
gy.NONE)

        .into(imgsoal);

tvsoal.setText(question.getSoal());

tvsoal1.setText(question.getA());

tvsoal2.setText(question.getB());

tvsoal3.setText(question.getC());

tvsoal4.setText(question.getD());
    }
}

private void checkAnswer(String
selectedAnswer) {

    StrukturData currentQuestion =
dataModels.get(currentQuestionInde
x);

    String correctAnswer =
currentQuestion.getJawaban();

    if
(selectedAnswer.equals(correctAnsw
er)) {

        if (submenu.isEmpty() &&
kate.isEmpty()){

            score += 5;

```

```

        } else {
            if (kate.equals("Ganda")) {
                score += 20;
            } else {
                score += 10;
            }
        }
    }

    // Toast.makeText(this, "Nilai: "
    + score,
    Toast.LENGTH_SHORT).show();

    if (currentQuestionIndex <
    dataModels.size() - 1) {
        currentQuestionIndex++;

        loadQuestion(currentQuestionIndex);
    } else {
        //
        Toast.makeText(LatihanActivity.this,
        "You have completed all questions.
        Final Score: " + score,
        Toast.LENGTH_LONG).show();

        Intent it = new
        Intent(LatihanActivity.this,
        HasilActivity.class);

        it.putExtra("SubMenu",
        submenu);

        it.putExtra("Kategori", kate);

        it.putExtra("Hasil",
        String.valueOf(score));

        startActivity(it);
    }
}

```

```

        finish();
    }
}

```

MenuActivity

```

public class MenuActivity extends
    AppCompatActivity {

```

```

    LinearLayout bgmenuhangeul,
    bgmenubatchim, bgmenubilangan,
    bgmenukosakata, bgmenulatihan;

```

```

    @Override
    protected void onCreate(Bundle
    savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_m
        enu);

        bgmenuhangeul =
        findViewById(R.id.bgmenuhangeul);

        bgmenubatchim =
        findViewById(R.id.bgmenubatchim);

        bgmenubilangan =
        findViewById(R.id.bgmenubilangan)
        ;

        bgmenukosakata =
        findViewById(R.id.bgmenukosakata)
        ;
    }
}

```

```

        bgmenulatihan =
        findViewById(R.id.bgmenulatihan);

```

```

bgmenuhangeul.setOnClickListener(
v -> {
    Intent it = new
    Intent(MenuActivity.this,
    SubMenuActivity.class);

    it.putExtra("SubMenu",
    "Hangeul");

    startActivity(it);

});

```

```

bgmenubatchim.setOnClickListener(
v -> startActivity(new
    Intent(MenuActivity.this,
    BatchimActivity.class)));

```

```

bgmenubilangan.setOnClickListener(
v -> {
    Intent it = new
    Intent(MenuActivity.this,
    SubMenuActivity.class);

    it.putExtra("SubMenu",
    "Bilangan");

    startActivity(it);

});

```

```

bgmenukosakata.setOnClickListener
(v -> {
    Intent it = new
    Intent(MenuActivity.this,
    SubMenuActivity.class);

```

```

        it.putExtra("SubMenu",
        "Kosakata");

        startActivity(it);

});

```

```

bgmenulatihan.setOnClickListener(v
-> {
    Intent it = new
    Intent(MenuActivity.this,
    LatihanActivity.class);

    it.putExtra("SubMenu", "");

    it.putExtra("Kategori", "");

    startActivity(it);

});

```

```

        Toolbar toolbar =
        findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction =
        getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new
        TextView(getApplicationContext());

        titleTextView.setText("Menu
        Materi".toUpperCase());

        titleTextView.setTextSize(20);

```

```

        titleTextView.setTextColor(Color.W
        HITE);

```

```

        titleTextView.setGravity(Gravity.CE
        NTER);

```

```

        Typeface customFont =
ResourcesCompat.getFont(this,
R.font.chewy);

titleTextView.setTypeface(customFont);

mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

mAction.setCustomView(titleTextView, new
ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT, Gravity.CENTER));

mAction.setDisplayHomeAsUpEnabled(true);

toolbar.setNavigationOnClickListener(v -> onBackPressed());
    }

    @Override
    public boolean
onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() ==
android.R.id.home) {
            onBackPressed();
        }
        return
super.onOptionsItemSelected(item);
    }

```

```

    }

```

PengantarActivity

```

public class PengantarActivity
extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_pengantar);

        findViewById(R.id.bgselanjutnya).setOnClickListener(v -> {
            startActivity(new
Intent(PengantarActivity.this,
MenuActivity.class));
        });

        Toolbar toolbar =
findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction =
getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new
TextView(getApplicationContext());

        titleTextView.setText("Tahukah
Kamu?".toUpperCase());
    }

```

```

        titleTextView.setTextSize(20);

        titleTextView.setTextColor(Color.W
HITE);

        titleTextView.setGravity(Gravity.CE
NTER);

        Typeface customFont =
ResourcesCompat.getFont(this,
R.font.chewy);

        titleTextView.setTypeface(customFo
nt);

        mAction.setDisplayOptions(ActionB
ar.DISPLAY_SHOW_CUSTOM);

        mAction.setCustomView(titleTextVi
ew, new
ActionBar.LayoutParams(ActionBar.
LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_C
ONTENT, Gravity.CENTER));

        mAction.setDisplayHomeAsUpEnabled(
true);

        toolbar.setNavigationOnClickListene
r(v -> onBackPressed());
    }

```

```

    @Override

    public boolean
onOptionsItemSelected(MenuItem
item) {

```

```

        if (item.getItemId() ==
android.R.id.home) {

            onBackPressed();

        }

        return
super.onOptionsItemSelected(item);

    }
}

```

SubMenuActivity

```

public class SubMenuActivity
extends AppCompatActivity {

    private RecyclerView
recyclerView;

    private SwipeRefreshLayout
mSwipeRefreshLayout;

    private RelativeLayout
errorlayout;

    private ImageView imgerror;

    private TextView errortitle,
errormessage;

    private Button btnretry;

    AdapterSubMenu
adapterSubMenu;

    AdapterSubMenu.ItemClickListener
itemClickListener;

    List<StrukturData> strukturData;

    String nmsubmenu;

    @Override

```



```

        protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_su
b_menu);

        Intent get = getIntent();

        nmsubmenu =
get.getStringExtra("SubMenu");

        errorlayout =
findViewById(R.id.rel_error);

        imgerror =
findViewById(R.id.imgerror);

        errortitle =
findViewById(R.id.tverror);

        errormessage =
findViewById(R.id.tvsuberror);

        btnretry =
findViewById(R.id.btnretry);

        mSwipeRefreshLayout =
findViewById(R.id.swaplistsubmenu
);

        recyclerView =
findViewById(R.id.listviewsubmenu
);

        recyclerView.setLayoutManager(new
LinearLayoutManager(this));

        recyclerView.setHasFixedSize(true);

        if (Objects.equals(nmsubmenu,
"Hangeul")){

            submenuhangeul();

        } else if
(Objects.equals(nmsubmenu,
"Bilangan")) {

            submenubilangan();

        } else if
(Objects.equals(nmsubmenu,
"Kosakata")) {

            submenukosakata();

        }

        mSwipeRefreshLayout.setColorSche
meResources(

            R.color.bhskorealight);

        mSwipeRefreshLayout.setOnRefresh
Listener(() -> {

            if
(Objects.equals(nmsubmenu,
"Hangeul")){

                submenuhangeul();

            } else if
(Objects.equals(nmsubmenu,
"Bilangan")) {

                submenubilangan();

            } else if
(Objects.equals(nmsubmenu,
"Kosakata")) {

                submenukosakata();

```

```

        }
    });

    itemClickListener = (view,
position) -> {

        if
(Objects.equals(nmsubmenu,
"Hangeul")){

            String kategori =
strukturData.get(position).getKategor
iHuruf();

            Intent it = new
Intent(SubMenuActivity.this,
TampilActivity.class);

            it.putExtra("SubMenu",
nmsubmenu);

            it.putExtra("KategoriSubMenu",
kategori);

            startActivity(it);

        } else if
(Objects.equals(nmsubmenu,
"Bilangan")) {

            String kategori =
strukturData.get(position).getKategor
iBilangan();

            Intent it = new
Intent(SubMenuActivity.this,
TampilActivity.class);

            it.putExtra("SubMenu",
nmsubmenu);

            it.putExtra("KategoriSubMenu",
kategori);

            startActivity(it);

        } else if
(Objects.equals(nmsubmenu,
"Kosakata")) {

            String kategori =
strukturData.get(position).getKategor
iKosakata();

            Intent it = new
Intent(SubMenuActivity.this,
TampilActivity.class);

            it.putExtra("SubMenu",
nmsubmenu);

            it.putExtra("KategoriSubMenu",
kategori);

            startActivity(it);

        }

        Toolbar toolbar =
findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction =
getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new
TextView(getApplicationContext());

        titleTextView.setText("Materi
".toUpperCase() +
nmsubmenu.toUpperCase());

        titleTextView.setTextSize(20);

        titleTextView.setTextColor(Color.W
HITE);

```

```
titleTextView.setGravity(Gravity.CE  
NTER);
```

```
    Typeface customFont =  
ResourcesCompat.getFont(this,  
R.font.chewy);
```

```
titleTextView.setTypeface(customFo  
nt);
```

```
mAction.setDisplayOptions(ActionB  
ar.DISPLAY_SHOW_CUSTOM);
```

```
mAction.setCustomView(titleTextVi  
ew, new  
ActionBar.LayoutParams(ActionBar.  
LayoutParams.WRAP_CONTENT,  
ActionBar.LayoutParams.WRAP_C  
ONTENT, Gravity.CENTER));
```

```
mAction.setDisplayHomeAsUpEnabled  
led(true);
```

```
toolbar.setNavigationOnClickListene  
r(v -> onBackPressed());  
}
```

```
@Override  
  
public boolean  
onOptionsItemSelected(MenuItem  
item) {  
  
    if (item.getItemId() ==  
android.R.id.home) {  
  
        onBackPressed();  
  
    }  
}
```

```
        return  
super.onOptionsItemSelected(item);  
    }  
}
```

```
private void submenuhangeul() {
```

```
errorlayout.setVisibility(View.GONE  
);
```

```
    showLoading();
```

```
    ApiInterface apiInterface =  
ApiClient.getApiClient().create(ApiI  
nterface.class);
```

```
    Call<List<StrukturData>> call  
= apiInterface.getHuruf();
```

```
    call.enqueue(new  
Callback<List<StrukturData>>() {
```

```
        @Override
```

```
        public void  
onResponse(@NonNull  
Call<List<StrukturData>> call,  
@NonNull  
Response<List<StrukturData>>  
response) {
```

```
            hideLoading();
```

```
            if(response.isSuccessful()  
&& response.body() != null){
```

```
onGetResult(response.body());
```

```
        } else {
```

```
            String errorCode;
```

```
            switch  
(response.code()){
```

```

        case 404:
            errorCode = "404
not found";
            break;
        case 500:
            errorCode = "500
server broken";
            break;
        default:
            errorCode =
"Unknown error";
            break;
    }

```

```

showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);
    }
}

```

```

@Override
    public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {
        hideLoading();

onErrorLoading(t.toString());
    }
});
}

```

```

private void submenubilangan() {

errorlayout.setVisibility(View.GONE
);

    showLoading();

    ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiI
nterface.class);

    Call<List<StrukturData>> call
= apiInterface.getBilangan();

    call.enqueue(new
Callback<List<StrukturData>>() {

        @Override

        public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {

            hideLoading();

            if(response.isSuccessful()
&& response.body() != null){

onGetResult(response.body());

            }else{

                String errorCode;

                switch
(response.code()){

                    case 404:

                        errorCode = "404
not found";

                        break;

```

```

        case 500:
            errorCode = "500
server broken";
            break;
        default:
            errorCode =
"Unknown error";
            break;
    }

    showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);
    }
}

@Override
    public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {
        hideLoading();

        onErrorLoading(t.toString());
    }
});
}

private void submenukosakata() {
    errorlayout.setVisibility(View.GONE
);

```

```

        showLoading();

        ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiI
nterface.class);

        Call<List<StrukturData>> call
= apiInterface.getKosakata();

        call.enqueue(new
Callback<List<StrukturData>>() {

            @Override
            public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {
                hideLoading();

                if(response.isSuccessful()
&& response.body() != null){

                    onGetResult(response.body());
                }else{
                    String errorCode;

                    switch
(response.code()){
                        case 404:
                            errorCode = "404
not found";

                            break;

                        case 500:
                            errorCode = "500
server broken";

                            break;

```

```

        default:
            errorCode =
"Unknown error";
            break;
        }

showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);
    }
}

@Override

    public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {
        hideLoading();

onErrorLoading(t.toString());
    }
});
}

    private void showLoading() {

mSwipeRefreshLayout.setRefreshing
(true);
    }

    private void hideLoading() {
        mSwipeRefreshLayout.setRefreshing
(false);
    }

    @SuppressWarnings("NotifyDataSetChan
ged")
    private void
onGetResult(List<StrukturData>
results) {
        adapterSubMenu = new
AdapterSubMenu(this, results,
itemClickListener);

        adapterSubMenu.notifyDataSetChan
ged();

        recyclerView.setAdapter(adapterSub
Menu);

        strukturData = results;
    }

    private void
onErrorLoading(String message) {

showErrorMessage(R.drawable.no_r
esult, "Oops...", "Network failure,
Please Try Again!\n"+ message);
    }

    private void
showErrorMessage(int imageView,
String title, String message){

```

```

        if(errorlayout.getVisibility() ==
View.GONE){

errorlayout.setVisibility(View.VISIB
LE);

        }

imgerror.setImageResource(imageVi
ew);

        errortitle.setText(title);

        errormessage.setText(message);

        btnretry.setOnClickListener(v -
> {

            if
(Objects.equals(nmsubmenu,
"Hangeul")){

                submenuhangeul();

            } else if
(Objects.equals(nmsubmenu,
"Bilangan")) {

                submenubilangan();

            } else if
(Objects.equals(nmsubmenu,
"Kosakata")) {

                submenukosakata();

            }

        });

    }
}

```

```

}

TampilActivity

public class TampilActivity extends
AppCompatActivity {

    private RecyclerView
recyclerView;

    private SwipeRefreshLayout
mSwipeRefreshLayout;

    private RelativeLayout
errorlayout;

    private ImageView imgerror;

    private TextView errortitle,
errormessage;

    private Button btnretry;

    AdapterTampil adapterTampil;

    AdapterTampil.ItemClickListener
itemClickListener;

    List<StrukturData> strukturData;

    String nmsubmenu,
kategorisubmenu;

    TextToSpeech tts;

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_ta
mpil);

```

```

    }

    Intent get = getIntent();

    nmsubmenu =
    get.getStringExtra("SubMenu");

    kategorisubmenu =
    get.getStringExtra("KategoriSubMenu");

    errorlayout =
    findViewById(R.id.rel_error);

    imgerror =
    findViewById(R.id.imgerror);

    errortitle =
    findViewById(R.id.tverror);

    errormessage =
    findViewById(R.id.tvsuberror);

    btnretry =
    findViewById(R.id.btnretry);

    mSwipeRefreshLayout =
    findViewById(R.id.swaplisttampil);

    recyclerView =
    findViewById(R.id.listviewtampil);

    recyclerView.setLayoutManager(new
    LinearLayoutManager(this));

    recyclerView.setHasFixedSize(true);

    if(!kategorisubmenu.equals("Kalimat
    ")) {

    recyclerView.setLayoutManager(new
    GridLayoutManager(this, 4));

    }

    if (Objects.equals(nmsubmenu,
    "Hangeul")){

    tampilhangeul(kategorisubmenu);

    } else if
    (Objects.equals(nmsubmenu,
    "Bilangan")) {

    tampilbilangan(kategorisubmenu);

    } else if
    (Objects.equals(nmsubmenu,
    "Kosakata")) {

    tampilkosakata(kategorisubmenu);

    }

    mSwipeRefreshLayout.setColorSchemeResources(
    R.color.bhskorealight);

    mSwipeRefreshLayout.setOnRefresh
    Listener(() -> {

    if
    (Objects.equals(nmsubmenu,
    "Hangeul")){

    tampilhangeul(kategorisubmenu);

    } else if
    (Objects.equals(nmsubmenu,
    "Bilangan")) {

```



```

        }
    tampilbilangan(kategorisubmenu);
    } else if
    (Objects.equals(nmsubmenu,
    "Kosakata")) {

    tampilkosakata(kategorisubmenu);
    }
    });

    tts = new
    TextToSpeech(getApplicationContext()
    t(), status -> {

        if (status !=
        TextToSpeech.ERROR) {

            String engine =
            "com.google.android.tts";

            List<TextToSpeech.EngineInfo>
            engines = tts.getEngines();

            for
            (TextToSpeech.EngineInfo
            engineInfo : engines) {

                if
                (engineInfo.name.equals(engine)) {

                    tts.setEngineByPackageName(engine
                    );

                    break;

                }

            }

            tts.setLanguage(new
            Locale("ko", "KR"));

            tts.setSpeechRate(0.6f);
        }
    });

    itemClickListener = (view,
    position) -> {

        String ucapkorea =
        strukturData.get(position).getTulisan
        Korea();

        tts.speak(ucapkorea.replaceAll("\\<.*
        ?\\>", ""),
        TextToSpeech.QUEUE_FLUSH,
        null);

    };

    ExtendedFloatingActionButton
    btnlatih =
    findViewById(R.id.btnlatihan);

    btnlatih.setOnClickListener(new
    View.OnClickListener() {

        @Override

        public void onClick(View v)

        {

            Intent it = new
            Intent(TampilActivity.this,
            LatihanActivity.class);

            it.putExtra("SubMenu",
            nmsubmenu);

            it.putExtra("Kategori",
            kategorisubmenu);

            startActivity(it);

        }

    });

```

```

if(kategorisubmenu.equals("Kalimat")){

    btnlatih.setVisibility(View.GONE);

    } else {

    btnlatih.setVisibility(View.VISIBLE)
    ;

    }

    Toolbar toolbar =
    findViewById(R.id.toolbar);

    setSupportActionBar(toolbar);

    ActionBar mAction =
    getSupportActionBar();

    assert mAction != null;

    TextView titleTextView = new
    TextView(getApplicationContext());

    titleTextView.setText(nmsubmenu.to
    UpperCase() + " " +
    kategorisubmenu.toUpperCase());

    titleTextView.setTextSize(20);

    titleTextView.setTextColor(Color.W
    HITE);

    titleTextView.setGravity(Gravity.CE
    NTER);

    Typeface customFont =
    ResourcesCompat.getFont(this,
    R.font.chewy);

    titleTextView.setTypeface(customFo
    nt);

    mAction.setDisplayOptions(ActionB
    ar.DISPLAY_SHOW_CUSTOM);

    mAction.setCustomView(titleTextVi
    ew, new
    ActionBar.LayoutParams(ActionBar.
    LayoutParams.WRAP_CONTENT,
    ActionBar.LayoutParams.WRAP_C
    ONTENT, Gravity.CENTER));

    mAction.setDisplayHomeAsUpEnabled(true);

    toolbar.setNavigationOnClickListener(v -> onBackPressed());

    }

    @Override

    public boolean
    onOptionsItemSelected(MenuItem
    item) {

        if (item.getItemId() ==
        android.R.id.home) {

            onBackPressed();

        }

        return
        super.onOptionsItemSelected(item);

    }

    private void tampilhangeul(String
    katenya) {

```

```

errorlayout.setVisibility(View.GONE
);

showLoading();

ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiI
nterface.class);

Call<List<StrukturData>> call
=
apiInterface.getHurufTampil(katenya
);

call.enqueue(new
Callback<List<StrukturData>>() {

@Override

public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {

hideLoading();

if(response.isSuccessful()
&& response.body() != null){

onGetResult(response.body());

} else{

String errorCode;

switch
(response.code()){

case 404:

errorCode = "404
not found";

break;

```

```

case 500:

errorCode = "500
server broken";

break;

default:

errorCode =
"Unknown error";

break;

}

showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);

}

}

@Override

public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {

hideLoading();

onErrorLoading(t.toString());

}

});

}

private void tampilbilangan(String
katenya) {

```

```
errorlayout.setVisibility(View.GONE
);
```

```
showLoading();
```

```
ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiI
nterface.class);
```

```
Call<List<StrukturData>> call
=
apiInterface.getBilanganTampil(kate
nya);
```

```
call.enqueue(new
Callback<List<StrukturData>>() {
```

```
@Override
```

```
public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {
```

```
hideLoading();
```

```
if(response.isSuccessful()
&& response.body() != null){
```

```
onGetResult(response.body());
```

```
}else{
```

```
String errorCode;
```

```
switch
(response.code()){
```

```
case 404:
```

```
errorCode = "404
not found";
```

```
break;
```

```
case 500:
```

```
errorCode = "500
server broken";
```

```
break;
```

```
default:
```

```
errorCode =
"Unknown error";
```

```
break;
```

```
}
```

```
showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);
```

```
}
```

```
}
```

```
@Override
```

```
public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {
```

```
hideLoading();
```

```
onErrorLoading(t.toString());
```

```
}
```

```
});
```

```
}
```

```
private void tampilkosakata(String
katenya) {
```

```

errorlayout.setVisibility(View.GONE
);

showLoading();

ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiI
nterface.class);

Call<List<StrukturData>> call
=
apiInterface.getKosakataTampil(kate
nya);

call.enqueue(new
Callback<List<StrukturData>>() {

@Override

public void
onResponse(@NonNull
Call<List<StrukturData>> call,
@NonNull
Response<List<StrukturData>>
response) {

hideLoading();

if(response.isSuccessful()
&& response.body() != null){

onGetResult(response.body());

} else {

String errorCode;

switch
(response.code()){

case 404:

errorCode = "404
not found";

break;

```

```

case 500:

errorCode = "500
server broken";

break;

default:

errorCode =
"Unknown error";

break;

}

showErrorMessage(R.drawable.no_r
esult, "No Result", "Please Try
Again!\n"+ errorCode);

}

}

@Override

public void
onFailure(@NonNull
Call<List<StrukturData>> call,
@NonNull Throwable t) {

hideLoading();

onErrorLoading(t.toString());

}

});

}

private void showLoading() {

mSwipeRefreshLayout.setRefreshing
(true);

```

```

    }

    private void hideLoading() {

mSwipeRefreshLayout.setRefreshing
(false);

    }

    @SuppressWarnings("NotifyDataSetChan
ged")

    private void
onGetResult(List<StrukturData>
results) {

        adapterTampil = new
AdapterTampil(this, results,
itemClickListener);

adapterTampil.notifyDataSetChanged();

recyclerView.setAdapter(adapterTam
pil);

        strukturData = results;

    }

    private void
onErrorLoading(String message) {

showErrorMessage(R.drawable.no_r
esult, "Oops...", "Network failure,
Please Try Again!\n"+ message);

    }

        private void
showErrorMessage(int imageView,
String title, String message){

            if(errorlayout.getVisibility() ==
View.GONE){

errorlayout.setVisibility(View.VISIB
LE);

            }

imgerror.setImageResource(imageVi
ew);

errortitle.setText(title);

errormessage.setText(message);

        btnretry.setOnClickListener(v -
> {

            if
(Objects.equals(nmsubmenu,
"Hangeul")){

tampilhangeul(kategorisubmenu);

            } else if
(Objects.equals(nmsubmenu,
"Bilangan")) {

tampilbilangan(kategorisubmenu);

            } else if
(Objects.equals(nmsubmenu,
"Kosakata")) {

tampilkosakata(kategorisubmenu);

```

```
    }  
  });  
  
}  
}
```

Lampiran_monitoring



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Nursari Maulidia Sulaiman	Pembimbing I : Ade Hastuty, ST., S.Kom., MT
NIM : 220280039	Pembimbing II : Mugaffir Yunus, ST., MT
Judul Skripsi : Pengembangan Aplikasi Pembelajaran Bahasa Korea	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Tambahkan class diagram	[Signature]	Konsultasi 1 - Revisi Aplikasi	[Signature]
Konsultasi 2 - Perbaiki white box & Flowgraph.	[Signature]	Konsultasi 2 - Revisi Aplikasi Tambahkan latihan pada tiap kategori	[Signature]
Konsultasi 3 - Lengkapi materi dasar Pemrograman	[Signature]	Konsultasi 3 Use Case Diagram Beri penjelasan tiap use case	[Signature]
Konsultasi 4 - Perbaiki Abstrak & kesimpulan	[Signature]	Konsultasi 4 Acc Seminar Hari!	[Signature]
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Nursari Maulidia Sulaiman	Pembimbing I : Ade Hastuty, ST., S.Kom., MT
NIM : 220280039	Pembimbing II : Mugaffir Yunus, ST., MT
Judul Skripsi : Pengembangan Aplikasi Pembelajaran Bahasa Korea	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Tambahkan class diagram		Konsultasi 1 - Revisi Aplikasi	
Konsultasi 2 - Perbaiki white box & Flowgraph.		Konsultasi 2 - Revisi Aplikasi Tambahkan latihan pada tiap kategori	
Konsultasi 3 - Lengkapi materi dasar pemrograman		Konsultasi 3 Use Case Diagram Beri penjelasan tiap use case	
Konsultasi 4 - Perbaiki Abstrak & kesimpulan		Konsultasi 4 Acc Seminar Hasil	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Nursari Maulidia Sulaiman	Pembimbing I : Ade Hastuty, ST., S.Kom., MT
NIM : 220260039	Pembimbing II : Mugaffir Yunus, ST., MT
Judul Skripsi : Pengembangan Aplikasi Pembelajaran Bahasa Korea	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Tambahkan teori white box dan black box		Konsultasi 1	
Konsultasi 2		Konsultasi 2 Acc	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

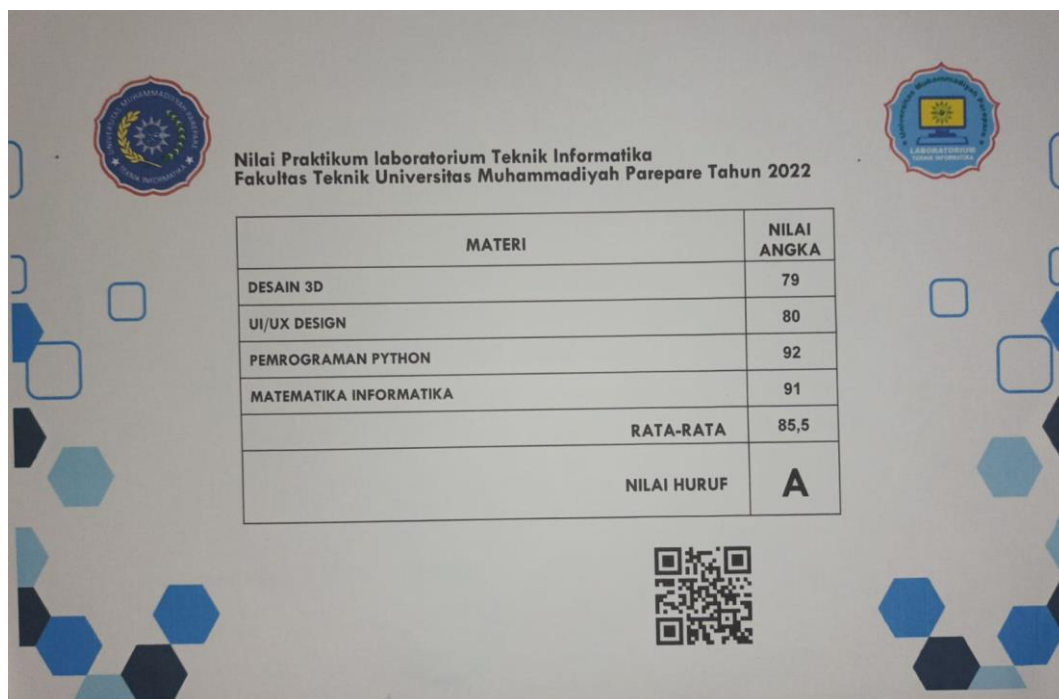
Lampiran_sertifikat



**Nilai Praktikum laboratorium Teknik Informatika
Fakultas Teknik Universitas Muhammadiyah Parepare Tahun 2023**

MATERI	NILAI ANGKA
PEMROGRAMAN WEB 2	82,5
PEMROGRAMAN JAVA	80
PEMROGRAMAN DELPHI 10	86,5
JARINGAN KOMPUTER 2	85
RATA-RATA	83,50
NILAI HURUF	A









Lampiran_buku

