

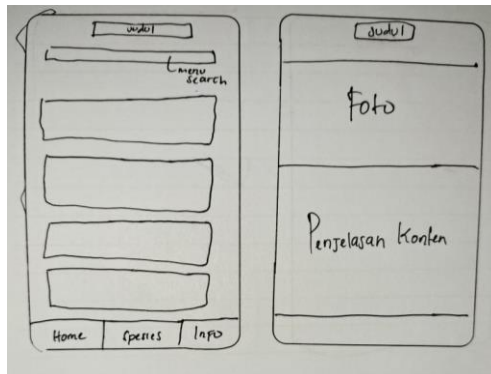
# DOKUMENTASI FITUR SPESIES PERLU DILINDUNGI (ANDROID STUDIO – KOTLIN )

**NAMA : RISMA OKTAVIANI**

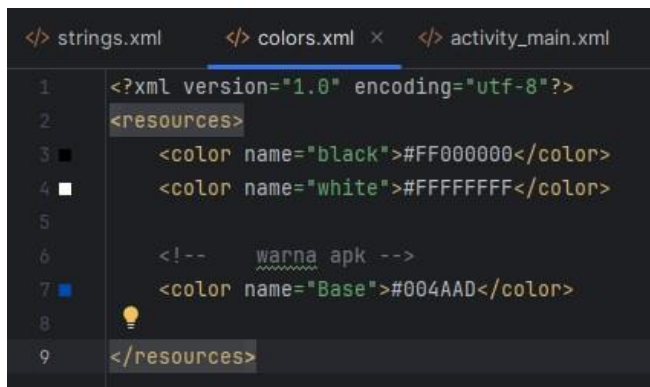
**NPM : 22312071**

**KELAS : IF 22DX**

1. Buat sketsa aplikasi yang akan dibuat



2. Langkah pertama buka aplikasi android studionya
3. Lalu buat project baru pada folder program, setelah itu atur warna terlebih dahulu agar mempermudah dalam kode, disini saya hanya menggunakan 3 warna yaitu, biru, hitam dan putih



4. atur di bagian string untuk bagian penjelasan konten nya nanti

```

1 <resources>
2 <string name="SeaCare">Seacare</string>
3
4 <string name="Dugong">
5 Dugong (Dugong dugon) adalah mamalia laut herbivora yang tersebar di perairan hangat dan dangkal di sepanjang
6 terumbu karang habitat padang lautan yang merupakan sumber makanan utama mereka, serta perburuan ilegal.
7 Upaya konservasi untuk dugong mencakup pendirian kawasan perlindungan laut yang meliputi habitat padang
8
9 </string>
10
11 <string name="Penyu">
12 Penyu adalah reptil laut yang memiliki berbagai spesies seperti penyu hijau, penyu sisik, dan penyu belah ketupat.
13 Upaya konservasi yang efektif untuk penyu meliputi pendirian taman laut dan kawasan konservasi, pengawasan
14
15 </string>
16
17 <string name="Paus">
18 Paus adalah mamalia laut besar yang mencakup paus bergigi (Odontoceti) dan paus balin (Mysticeti). Sejarah
19
20 </string>
21
22 <string name="HiuPaus">
23 Ikan hiu paus (Rhincodon typus) adalah salah satu dari sedikit spesies hiu yang merupakan pemakan plankton.
24 Ancaman terbesar bagi ikan hiu paus berasal dari perburuan komersial untuk siripnya yang memiliki nilai
25
26 </string>
27
28 <string name="PariManta">
29 Ikan pari manta (Manta spp.) adalah pari besar yang dikenal dengan sirip dada yang lebar dan pola spot hitam
30
31 </string>
32 </resources>

```

String digunakan untuk menyimpan teks yang akan digunakan dalam aplikasi dan mengelola menjadi lebih singkat agar mudah dipanggil

## 5. setelah membuat penjelasan setiap konten, sekarang ke **activity\_main.xml**

### a. **ConstraintLayout**

Layout utama yang mengatur tata letak komponen aplikasi yang akan dibuat

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">

```

### b. **FrameLayout**

Tempat untuk menampung fragmen atau tampilan lainnya, diatur untuk mengisi ruang di atas **BottomNavigationView**

```

9
10 <FrameLayout
11     android:id="@+id/frm_layout"
12     android:layout_width="0dp"
13     android:layout_height="0dp"
14     app:layout_constraintBottom_toBottomOf="@+id/bottom_nav"
15     app:layout_constraintEnd_toEndOf="parent"
16     app:layout_constraintStart_toStartOf="parent"
17     app:layout_constraintTop_toTopOf="parent">
18
19 </FrameLayout>

```

### c. BottomNavigationView

Navigasi bawah untuk aplikasi, menampilkan ikon ikon dengan ukuran dan warna tertentu.

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_nav"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="@color/white"
    app:itemIconSize="35dp"
    app:itemIconTint="@color/black"
    app:itemTextColor="@color/white"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:menu="@menu/bottom_nav">

</com.google.android.material.bottomnavigation.BottomNavigationView>
```

- Sebelum menambahkan ini ke **activity\_main.xml**, harus buat directory menu terlebih dahulu, klik kanan pada **folder res > new > directory > ketik menu**
- Setelah folder menu sudah terbuat, maka buat resource file dalam folder tersebut dengan nama **bottom\_nav**, lalu isi kan dengan nama dan icon, seperti tittle = Home, id = home, icon = house\_solid

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:title="Home"
        android:id="@+id/home"
        android:icon="@drawable/house_solid">
    </item>
    <item
        android:title="Spesies"
        android:id="@+id/spesies"
        android:icon="@drawable/fish_solid">
    </item>
    <item
        android:title="Info"
        android:id="@+id/info"
        android:icon="@drawable/circle_info_solid">
    </item>
</menu>
```

### d. TextView

Textview adalah tools yang digunakan untuk menambahkan teks kedalam activity yang ada, disini saya menggunakan teks view untuk judul pada aplikasi tersebut, dengan margin start dan endnya masing masing 16sp

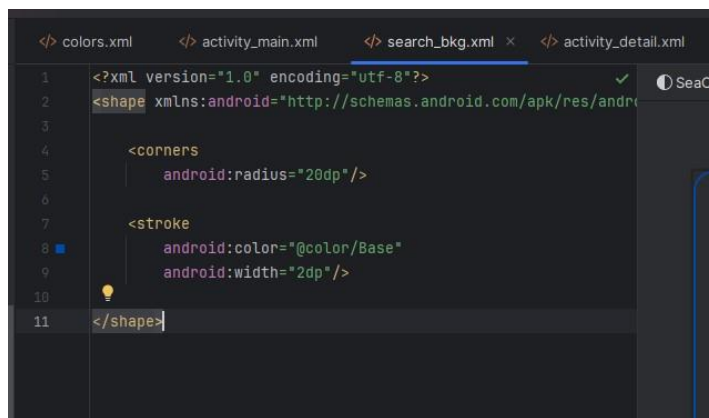
```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/detailTitle"
    android:layout_marginTop="30dp"
    android:text="Spesies yang dilindungi"
    android:textColor="@color/Base"
    android:textStyle="bold"
    android:textSize="25sp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16sp"
    android:textAlignment="center"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

```

#### e. Search View

Sebelum ke searchview, buat terlebih dahulu border untuk searchnya, dengan cara klik kanan pada drawable > resource file > ketik search\_bkg, lalu root elemennya ubah jadi shape



#### < activity\_main.xml >

```

<androidx.appcompat.widget.SearchView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:id="@+id/search"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:iconifiedByDefault="false"
    app:searchHintIcon="@null"
    app:queryHint="Search..."
    android:focusable="false"
    app:closeIcon="@drawable/ic_clear"
    app:searchIcon="@drawable/ic_search"
    android:background="@drawable/search_bkg"/>

```

`androidx.appcompat.widget.SearchView` adalah Komponen tampilan untuk menerima input pencarian dari pengguna.

#### **Layout Width dan Height:**

- `android:layout_height="wrap_content"`: Tinggi menyesuaikan dengan konten di dalam `SearchView`.

#### **Margin:**

- `android:layout_marginTop="80dp"`: Margin atas sebesar 80dp.
- `android:layout_marginStart="20dp"`: Margin awal (kiri) sebesar 20dp.
- `android:layout_marginEnd="20dp"`: Margin akhir (kanan) sebesar 20dp.

#### **ID**

`android:id="@+id/search"`: Memberikan ID unik pada `SearchView` yang dapat digunakan untuk mereferensikan komponen ini dalam kode.

#### **Constraints**

- `app:layout_constraintEnd_toEndOf="parent"`: Menyandarkan sisi akhir `SearchView` ke sisi akhir parent.
- `app:layout_constraintStart_toStartOf="parent"`: Menyandarkan sisi awal `SearchView` ke sisi awal parent.
- `app:layout_constraintTop_toTopOf="parent"`: Menyandarkan sisi atas `SearchView` ke sisi atas parent.

#### **Properti SearchView:**

- `app:iconifiedByDefault="false"` `SearchView` tidak di-ikonifikasi secara default, sehingga selalu ditampilkan dalam bentuk penuh.
- `app:searchHint="@null"` Tidak ada hint teks saat search field kosong.
- `app:queryHint="Search...."` Teks hint yang muncul saat pengguna mulai mengetik.
- `android:focusable="false"` `SearchView` tidak dapat menerima fokus input.
- `app:closeIcon="@drawable/ic_clear"` Ikon yang digunakan untuk tombol clear.
- `app:searchIcon="@drawable/ic_search"` Ikon yang digunakan untuk tombol search.

- *android:background="@drawable/search\_bkg"* Latar belakang yang digunakan untuk SearchView.

#### f. RecyclerView

- *android:layout\_width="match\_parent"* Atribut ini menetapkan lebar RecyclerView harus sama dengan lebar wadah activity.
- *android:layout\_height="0dp"* Atribut ini mengatur tinggi RecyclerView menjadi 0dp.
- *android:id="@+id/recyclerView"* Atribut ini memberikan ID unik (recyclerView) ke RecyclerView. ID ini dapat digunakan untuk mereferensi RecyclerView dari kode aktivitas.
- *android:layout\_marginTop="12dp"* Atribut ini menetapkan margin 12dp di bagian atas RecyclerView.
- *app:layout\_constraintTop\_toBottomOf="@id/search"* Atribut ini menggunakan ConstraintLayout untuk membatasi tepi atas RecyclerView ke bagian bawah tampilan dengan ID search. RecyclerView akan ditempatkan di bawah tampilan dengan ID search.
- *app:layout\_constraintBottom\_toBottomOf="parent"* Atribut ini membatasi tepi bawah RecyclerView ke bagian bawah.
- *app:layout\_constraintStart\_toStartOf="parent"* Atribut ini membatasi tepi kiri RecyclerView ke tepi kiri
- *app:layout\_constraintEnd\_toEndOf="parent"* Atribut ini membatasi tepi kanan RecyclerView ke tepi kanan

### 6. Lanjut ke **Item\_layout.xml** -> file untuk item atau konten yang akan dibuat

#### a. CardView

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      app:cardCornerRadius="20dp"
8      android:layout_marginHorizontal="10dp"
9      android:background="@color/white"
10     android:layout_marginVertical="10dp"
11     app:cardElevation="8dp">
12
13
14     <androidx.constraintlayout.widget.ConstraintLayout
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:background="@color/Base">

```

- `android:layout_width="match_parent"`: Atribut ini menetapkan lebar CardView harus sama dengan lebar wadah induknya.
- `android:layout_height="wrap_content"`: Atribut ini menetapkan tinggi CardView harus berdasarkan kontennya. CardView hanya akan menggunakan ruang vertikal yang dibutuhkan untuk menampilkan elemen anaknya.
- `app:cardCornerRadius="20dp"`: Atribut ini mengatur radius sudut CardView menjadi 20dp, sehingga memiliki sudut membulat.
- `android:layout_marginHorizontal="10dp"`: Atribut ini mengatur margin horizontal sebesar 10dp di sekitar CardView.
- `android:background="@color/white"`: Atribut ini mengatur warna latar belakang CardView menjadi putih. Referensi warna di sini kemungkinan menunjuk ke warna yang ditentukan dalam file resource warna terpisah.
- `android:layout_marginVertical="10dp"`: Atribut ini mengatur margin vertikal sebesar 10dp di sekitar CardView.
- `app:cardElevation="8dp"`: Atribut ini mengatur elevasi CardView menjadi 8dp, menciptakan efek bayangan tipis untuk memberikan kedalaman.

#### b. **Imageview**

Digunakan menampilkan gambar dalam aplikasi

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/image"
    android:layout_marginStart="20dp"
    android:padding="8dp"
    android:adjustViewBounds="true"
    android:scaleType="fitXY"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:src="@drawable/dugong" />
```

#### c. **TextView**

Textview adalah tools yang digunakan untuk menambahkan teks kedalam activity yang ada

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/title"
    android:textColor="@color/white"
    android:textSize="15sp"
    android:textStyle="bold"
    android:text="Title"
    android:layout_marginStart="20dp"
    app:layout_constraintStart_toEndOf="@id/image"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" />

```

## 7. MainActivity.kt

```

12 class MainActivity : AppCompatActivity() {
13
14     private lateinit var recyclerView: RecyclerView
15     private lateinit var dataList: ArrayList<DataClass>
16     lateinit var imageUrl: Array<Int>
17     lateinit var titleList: Array<String>
18     lateinit var descList: Array<String>
19     lateinit var detailImageList: Array<Int>
20     private lateinit var myAdapter: AdapterClass
21     private lateinit var searchView: SearchView
22     private lateinit var searchList: ArrayList<DataClass>
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27
28         imageUrl = arrayOf(
29             R.drawable.dugong,
30             R.drawable.lumba,
31             R.drawable.paus,
32             R.drawable.penyu,
33             R.drawable.paus,
34             R.drawable.parimanta,
35             R.drawable.penguin,
36             R.drawable.gurita,
37             R.drawable.ikankarang,
38             R.drawable.hiu

```

### a. Mendefinisikan Data:

Kode ini mendefinisikan beberapa array untuk menyimpan data biodata satwa laut

- imageUrl: Array berisi resource ID gambar untuk setiap satwa.
- titleList: Array berisi judul untuk setiap satwa laut.
- descList: Array berisi deskripsi singkat untuk setiap satwa laut.
- detailImageList: Array berisi resource ID gambar detail untuk setiap satwa laut.



```

titleList = arrayOf(
    "Tentang Dugong",
    "Tentang Lumba-Lumba",
    "Tentang Paus",
    "Tentang Penyu",
    "Tentang Hiu Paus",
    "Tentang Pari manta",
    "Tentang Penguin",
    "Tentang Gurita",
    "Tentang Ikan Karang",
    "Tentang Hiu"
)

descList = arrayOf(
    getString(R.string.Dugong),
    getString(R.string.Lumba),
    getString(R.string.Paus),
    getString(R.string.Penyu),
    getString(R.string.HiuPaus),
    getString(R.string.PariManta),
    getString(R.string.Penguin),
    getString(R.string.Gurita),
    getString(R.string.IkanKarang),
    getString(R.string.Hiu))

```

```

detailImageList = arrayOf(
    R.drawable.dugongframe,
    R.drawable.lumbaframe,
    R.drawable.pausframe,
    R.drawable.penyuframe,
    R.drawable.hiupausframe,
    R.drawable.parimantaframe,
    R.drawable.penguinframe,
    R.drawable.guritaframe,
    R.drawable.karangframe,
    R.drawable.hiuframe
)

recyclerView = findViewById(R.id.recyclerView)
searchView = findViewById(R.id.search)
recyclerView.layoutManager = LinearLayoutManager(context, this)
recyclerView.setHasFixedSize(true)

dataList = arrayListOf<DataClass>()
searchList = arrayListOf<DataClass>()
getData()

searchView.clearFocus()
searchView.setOnQueryTextListener(object : SearchView.OnQueryTextListener {
    override fun onQueryTextSubmit(query: String?): Boolean {
        searchView.clearFocus()
        return true
    }
})

```

## b. Class DataClass:

Kode ini mendefinisikan sebuah class DataClass yang digunakan untuk menyimpan informasi terkait masing-masing satwa laut. Class ini memiliki properti untuk menyimpan:

- image: resource ID gambar
- dataTitle: judul satwa laut

- dataDesc: deskripsi singkat satwa laut
- detailImage: resource ID gambar detail satwa laut

### c. Activity Utama (MainActivity):

- Class MainActivity merupakan class utama yang mengontrol tampilan awal aplikasi.
- Method onCreate dipanggil saat activity dibuat. Method ini melakukan beberapa hal:.

```

94
95  override fun onQueryTextChange(newText: String?): Boolean {
96      searchList.clear()
97      val searchText = newText!!.toLowerCase(Locale.getDefault())
98      if (searchText.isNotEmpty()){
99          dataList.forEach{ it: DataClass
100              if (it.dataTitle.toLowerCase(Locale.getDefault()).contains(searchText)) {
101                  searchList.add(it)
102              }
103          }
104          recyclerView.adapter!!.notifyDataSetChanged()
105      } else {
106          searchList.clear()
107          searchList.addAll(dataList)
108          recyclerView.adapter!!.notifyDataSetChanged()
109      }
110      return false
111  }
112  }
113  myAdapter = AdapterClass(searchList)
114  recyclerView.adapter = myAdapter
115
116  myAdapter.onItemClickListener = { it: DataClass
117      val intent = Intent( packageContext: this, DetailActivity::class.java)
118      intent.putExtra( name: "android", it)
119      startActivity(intent)
120  }
}

private fun getData(){
    for (i in imageList.indices){
        val dataClass = DataClass(imageList[i], titleList[i], descList[i], detailImageList[i])
        dataList.add(dataClass)
    }
    searchList.addAll(dataList)
    recyclerView.adapter = AdapterClass(searchList)
}
}

```

### d. Pencarian:

Kode ini mengimplementasikan fungsionalitas pencarian menggunakan SearchView.

- Ketika pengguna mengetikkan sesuatu di search bar, method *onQueryTextChange* akan dipanggil.
- Method ini memfilter dataList berdasarkan text yang diketikkan pengguna dan memperbarui searchList dengan hasil filter.

- Adapter myAdapter kemudian diberitahu tentang perubahan data menggunakan *notifyDataSetChanged* sehingga tampilan recyclerView diperbarui sesuai dengan hasil pencarian.

#### e. Detail Activity:

Kode digunakan untuk menampilkan detail lengkap dari satwa laut yang dipilih pengguna pada daftar.

### 8. Detail\_acitivity.xml

Tampilan ini menggunakan `androidx.constraintlayout.widget.ConstraintLayout` sebagai layout utama. `ConstraintLayout` adalah layout yang membantu untuk memposisikan elemen secara relatif terhadap satu sama lain atau terhadap tepi layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/main"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:background="@color/white"
9   tools:context=".DetailActivity">
```

#### a. TextView

Menampilkan judul satwa laut dengan ukuran teks yang besar dan tebal.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/detailTitle"
    android:layout_marginTop="30dp"
    android:text="Dugong"
    android:textStyle="bold"
    android:textColor="@color/Black"
    android:textSize="28sp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16sp"
    android:textAlignment="center"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/detailDesc"
    android:layout_marginTop="20dp"
    android:text="Dugong (Dugong dugon) adalah mamalia laut herbivora yan..."
    android:textColor="@color/black"
    android:textAlignment="viewStart"
    android:textSize="12sp"
    android:padding="12dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/detailImage"/>
```

#### b. ImageView

Menampilkan gambar detail satwa laut dengan ukuran tinggi 200dp.

```
27 <ImageView
28     android:layout_width="wrap_content"
29     android:layout_height="200dp"
30     android:id="@+id/detailImage"
31     android:layout_marginTop="20dp"
32     android:scaleType="fitXY"
33     android:src="@drawable/dugongframe"
34     android:layout_marginStart="8dp"
35     android:layout_marginEnd="8dp"
36     app:layout_constraintStart_toStartOf="parent"
37     app:layout_constraintEnd_toEndOf="parent"
38     app:layout_constraintTop_toBottomOf="@id/detailTitle"/>
```

## 9. DetilAcitivity Class

```
10
11 class DetailActivity : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         enableEdgeToEdge()
15         setContentView(R.layout.activity_detail)
16
17         val getData = intent.getParcelableExtra<DataClass>("android")
18         if (getData != null) {
19             val detailTitle: TextView = findViewById(R.id.detailTitle)
20             val detailDesc: TextView = findViewById(R.id.detailDesc)
21             val detailImage: ImageView = findViewById(R.id.detailImage)
22             detailTitle.text = getData.dataTitle
23             detailDesc.text = getData.dataDesc
24             detailImage.setImageResource(getData.dataDetailImage)
25         }
26     }
27 }
```

### a. Class DetailActivity:

Class DetailActivity adalah class yang mengontrol tampilan detail dari satwa laut yang dipilih pengguna.

### b. Method onCreate:

Method onCreate dipanggil saat activity dibuat. Method ini melakukan beberapa hal:

- **enableEdgeToEdge():** Memungkinkan konten activity ditampilkan secara penuh, menghilangkan status bar (secara opsional).
- **setContentView(R.layout.activity\_detail):** Menyetel layout untuk activity ini menggunakan activity\_detail.xml.
- **Mengirimkan Data:**
  - Kode ini mengambil data yang dikirimkan dari MainActivity menggunakan `intent.getParcelableExtra<DataClass>("android")`.

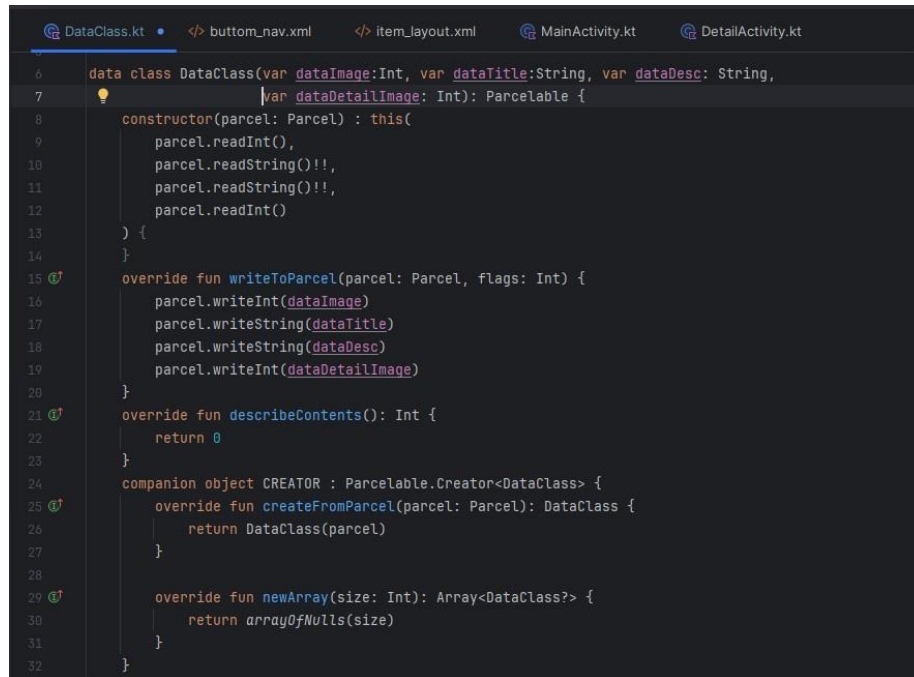
- Variabel `getData` akan berisi objek `DataClass` yang menyimpan informasi satwa laut yang dipilih.

### c. Menampilkan Detail:

Jika `getData` tidak null (artinya data berhasil diterima), kode ini mengisi informasi detail ke dalam view terkait:

- `findViewById(R.id.detailTitle):` Digunakan untuk mendapatkan referensi ke `TextView` dengan ID `detailTitle` pada layout.
- Text pada `detailTitle` kemudian diubah sesuai dengan `dataTitle` yang terdapat pada objek `getData`.
- Operasi serupa dilakukan untuk `detailDesc` (deskripsi) dan `detailImage` (gambar detail) menggunakan informasi dari objek `DataClass`.

## 10. DataClass



```

1  data class DataClass(var dataImage: Int, var dataTitle: String, var dataDesc: String,
2  |                     var dataDetailImage: Int): Parcelable {
3
4  |     constructor(parcel: Parcel) : this(
5  |         parcel.readInt(),
6  |         parcel.readString()!!,
7  |         parcel.readString()!!,
8  |         parcel.readInt()
9  |     ) {
10 |     }
11
12 |     override fun writeToParcel(parcel: Parcel, flags: Int) {
13 |         parcel.writeInt(dataImage)
14 |         parcel.writeString(dataTitle)
15 |         parcel.writeString(dataDesc)
16 |         parcel.writeInt(dataDetailImage)
17 |     }
18
19 |     override fun describeContents(): Int {
20 |         return 0
21 |     }
22
23 |     companion object CREATOR : Parcelable.Creator<DataClass> {
24 |         override fun createFromParcel(parcel: Parcel): DataClass {
25 |             return DataClass(parcel)
26 |         }
27 |     }
28
29 |     override fun newArray(size: Int): Array<DataClass?> {
30 |         return arrayOfNulls(size)
31 |     }
32 | }

```

`DataClass` adalah sebuah class yang digunakan untuk menyimpan informasi terkait masing-masing satwa laut.

- Class ini memanfaatkan keyword `data`, yang secara otomatis membuat beberapa hal:
  - Konstruktor primary dengan semua properti sebagai parameter.

- Getter dan setter untuk semua properti (walaupun dalam contoh ini hanya ada `var` yang berarti properti dapat langsung diubah).
- Fungsi `toString` yang mengembalikan representasi string dari objek.
- Fungsi `equals` dan `hashCode` untuk perbandingan objek.

a. **Properti:**

- `dataImage`: resource ID gambar untuk satwa laut.
- `dataTitle`: Judul satwa laut (berupa String).
- `dataDesc`: Deskripsi singkat satwa laut (berupa String).
- `dataDetailImage`: resource ID gambar detail satwa laut.

b. **Implementasi Parcelable:**

Class ini mengimplementasikan interface `Parcelable`. Interface ini digunakan untuk memungkinkan antarmuka antar Activity dalam aplikasi Android untuk saling bertukar data.

## 11. AdapterClass

Class ini berfungsi sebagai adapter untuk RecyclerView.

RecyclerView adalah komponen yang menampilkan daftar item yang dapat digulir. Adapter bertanggung jawab atas pembuatan, pengikatan data, dan pengelolaan tampilan individual item pada RecyclerView.

```
class AdapterClass(private val dataList: ArrayList<DataClass>):
    RecyclerView.Adapter<AdapterClass.ViewHolderClass>() {

    var onItemClick: ((DataClass)-> Unit)? = null

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolderClass {
        val itemView = LayoutInflater.from(parent.context).
            inflate(R.layout.item_layout, parent, attachToRoot: false)
        return ViewHolderClass(itemView)
    }

    override fun onBindViewHolder(holder: ViewHolderClass, position: Int) {
        val currentItem = dataList[position]
        holder.rvImage.setImageResource(currentItem.dataImage)
        holder.rvtitle.text = currentItem.dataTitle

        holder.itemView.setOnClickListener { it: View?
            onItemClick?.invoke(currentItem)
        }
    }

    override fun getItemCount(): Int {
        return dataList.size
    }

    class ViewHolderClass(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val rvImage: ImageView = itemView.findViewById(R.id.image)
        val rvtitle: TextView = itemView.findViewById(R.id.title)
    }
}
```

**a. Konstruktor:**

- Konstruktor primary class ini menerima parameter `dataList` berupa `ArrayList<DataClass>`.
- Parameter ini berisi daftar objek `DataClass` yang menyimpan informasi satwa laut.

**b. Properti:**

- `onItemClick`: Variabel ini bertipe lambda function yang akan dipanggil ketika pengguna mengklik item pada `RecyclerView`.
- Lambda function ini menerima objek `DataClass` yang terkait dengan item yang diklik.

**c. Fungsi `onCreateViewHolder`:**

- Fungsi ini dipanggil ketika `RecyclerView` perlu membuat tampilan baru untuk sebuah item.
- Fungsi ini melakukan hal berikut:
  - Menggunakan `LayoutInflater` untuk meng inflate layout item individual (biasanya `item_layout.xml`) ke dalam view.
  - Membangun objek `ViewHolderClass` yang menyimpan referensi ke view komponen penyusun item (seperti `ImageView` dan `TextView`).
  - Mengembalikan objek `ViewHolderClass` yang baru dibuat.

**d. Fungsi `onBindViewHolder`:**

- Fungsi ini dipanggil untuk mengikat data ke tampilan item individual pada `RecyclerView`, Fungsi ini menerima:
  - `holder`: Objek `ViewHolderClass` yang berisi referensi ke view komponen penyusun item.
  - `position`: Posisi item saat ini pada `RecyclerView` (index pada `dataList`).
    - Fungsi ini:
      - Mengambil objek `DataClass` dari `dataList` berdasarkan posisi item.
      - Mengatur image view pada item (`rvImage`) dengan resource ID gambar dari objek `DataClass`.



- Mengatur text view pada item (`rvtitle`) dengan judul satwa laut dari objek `DataClass`.
- Menyetel click listener pada view item (`itemView`).
- Ketika item diklik, lambda function yang ditetapkan ke `onItemClick` akan dipanggil.
- Fungsi tersebut akan menerima objek `DataClass` yang terkait dengan item yang diklik.

**e. Fungsi `getItemCount`:**

- Fungsi ini mengembalikan jumlah item pada `RecyclerView`.
- Ini memberitahu `RecyclerView` berapa banyak item yang perlu dikelola oleh adapter.

**f. Class `ViewHolderClass`:**

- Class ini merupakan view holder untuk item individual pada `RecyclerView`.
- Class ini menyimpan referensi ke view komponen penyusun item, seperti `ImageView` untuk gambar dan `TextView` untuk judul.

## 12. HASIL / OUTPUT

### Menu Utamanya



### Menu Search



### Menu Penjelasam

