

LAPORAN RESPONSI PENGEMBANGAN APLIKASI BERGERAK

RESPONSI 1



Disusun oleh:

Nama : Afif Imam Rahadi

Nim : L0122006

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

1. Screenshot Source Code

Source code ini merupakan source code yang digunakan untuk responsi ke-1.

MainActivity.kt

```
package com.l0122006.afifimam.responsi1

import ...

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

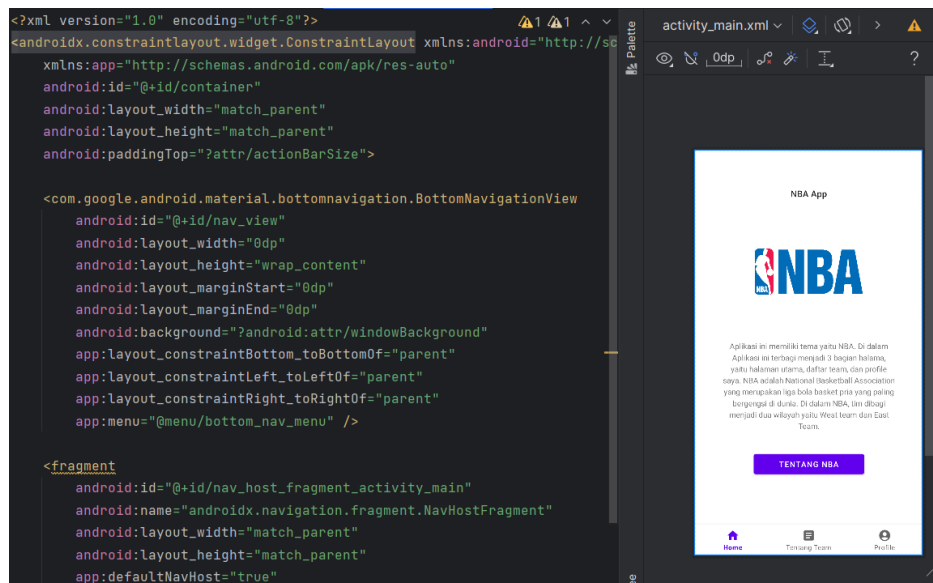
        val navView: BottomNavigationView = binding.navView

        val navController = findNavController(R.id.nav_host_fragment_activity_main)
        val appBarConfiguration = AppBarConfiguration(
            listOf(
                R.id.navigation_home, R.id.navigation_daftar, R.id.navigation_profile
            )
        )
        setupActionBarWithNavController(navController, appBarConfiguration)
        navView.setupWithNavController(navController)
    }
}
```

Ini adalah source code dari halaman MainActivity.kt. Pada bagian awal, seperti biasa terdapat package yang digunakan untuk mengatur file project. Lalu, diikuti dengan import library yang digunakan agar program dapat berjalan dengan baik. Selanjutnya, terdapat class MainActivity yang merupakan turunan dari AppCompatActivity. Berikutnya, mendeklarasikan properti binding dengan tipe ActivityMainBinding. Properti ini digunakan untuk menghubungkan elemen-elemen tampilan yang didefinisikan dalam layout XML activity_main.xml.

Terdapat Metode onCreate yang dipanggil saat aktivitas dibuat. Di sini, ActivityMainBinding.inflate(layoutInflater) digunakan untuk menginisialisasi binding dan setContentView(binding.root) untuk mengatur tampilan aktivitas dengan menggunakan root view dari binding. Selanjutnya, terdapat pengaturan bottom navigation karena pada responsi ini, project menggunakan bottom navigation. Terdapat deklarasi configuration untuk mengatur semua navigasi agar tersinkron dan berjalan dengan baik. Dan jangan lupa untuk sesuaikan dengan id masing-masing navigation.

Activity_main.xml



Code di atas adalah code untuk mengatur layout pada `activity_main.xml`. Tampilan kurang lebih seperti gambar bagian kanan yang terdapat judul halaman, foto, deskripsi singkat tentang aplikasi, dan terakhir terdapat button yang mengarah ke fragment tentang nba. Tampilan di atas menggunakan constraint layout agar terlihat lebih rapih. Pada codingan di atas, tidak terdapat codingan terkait text view, image view dan lain-lain. Ini disebabkan karena adanya binding ke fragment home yang menyebabkan tampilan di fragment home akan masuk ke dalam sini. Dan juga pengaturan pada mobile navigation.

HomeFragment.kt

```
package com.l0122006.afimam.responsi1.ui.home

import ...

class HomeFragment : Fragment() {
    private var _binding: FragmentHomeBinding? = null
    private val binding get() = _binding!!

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        // Inflate the layout for this fragment
        _binding = FragmentHomeBinding.inflate(inflater, container, attachToParent: false)
        val view = binding.root
        return view
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

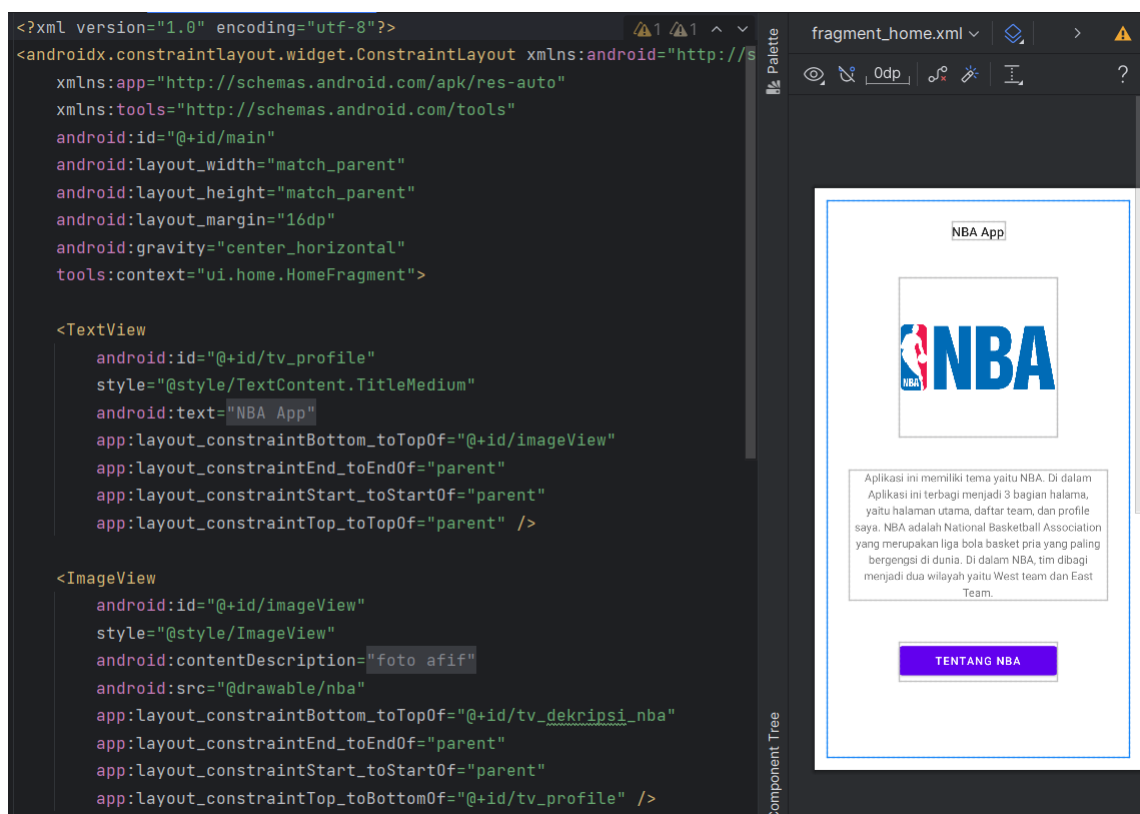
        binding.btnTentangNba.setOnClickListener { it: View?
            Navigation.findNavController(view).navigate(R.id.action_navigation_home_to_tentangNBAFragment)
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        _binding = null
    }
}
```

Code di atas adalah code untuk bagian home. Halaman home ini menggunakan blank fragment. Dalam kelas ini, variabel `_binding` bertipe `FragmentHomeBinding` dideklarasikan sebagai properti yang dapat berubah (nullable). Variabel ini digunakan untuk mengakses elemen-elemen tampilan yang terdapat dalam layout `fragment_home.xml` melalui view binding, yang memudahkan pengelolaan elemen UI secara langsung tanpa perlu menggunakan metode `findViewById`. Metode `onCreateView` bertugas untuk meng-inflate (memuat) layout XML menjadi objek `View`. Di sini, layout `fragment_home.xml` di-inflate menggunakan `FragmentHomeBinding`, dan root view dari binding tersebut dikembalikan sebagai tampilan utama dari fragment ini.

Metode `onViewCreated` dipanggil setelah `onCreateView`, dan digunakan untuk menginisialisasi komponen-komponen UI atau melakukan pengaturan lebih lanjut pada tampilan. Dalam metode ini, ditetapkan sebuah click listener pada tombol dengan ID `btnTentangNba`. Ketika tombol ini diklik, tombol akan dieksekusi untuk menavigasi dari `HomeFragment` ke `TentangNBAFragment` menggunakan `Navigation Component`, yang memudahkan pengelolaan navigasi antar fragmen dalam aplikasi Android.

Fragment_home.xml



Tampilan home fragment ini, sama seperti `activity_main.xml`. Tetapi, pengaturan untuk halaman main dilakukan pada halaman ini. Halaman ini menampilkan nama app, gambar app, deskripsi singkat tentang app, dan terakhir terdapat button untuk mengarah ke halaman lain.

Daftar Fragment.kt

```
package com.l0122006.afifimam.responsi1.ui.daftar

import ...

class DaftarFragment : Fragment(), ListTeamAdapter.OnItemClickListener {
    private lateinit var rvTeams: RecyclerView
    private val list = ArrayList<Team>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val rootView = inflater.inflate(R.layout.fragment_daftar, container, attachToRoot: false)

        rvTeams = rootView.findViewById(R.id.rv_teams)
        rvTeams.setHasFixedSize(true)

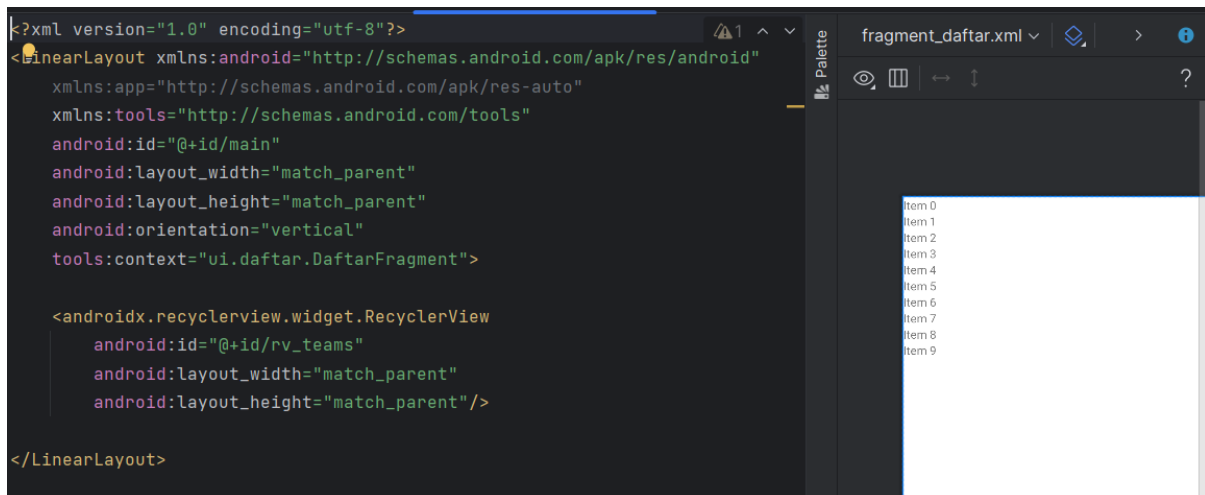
        list.addAll(getListTeams())
        showRecyclerList()

        return rootView
    }

    private fun getListTeams(): ArrayList<Team> {
        val dataName = resources.getStringArray(R.array.data_name)
        val dataDesc = resources.getStringArray(R.array.data_desc)
        val dataImg = resources.obtainTypedArray(R.array.data_img)
        val listTeam = ArrayList<Team>()
        for (i in dataName.indices) {
            val team = Team(dataName[i], dataDesc[i], dataImg.getResourceId(i, defValue: -1))
            listTeam.add(team)
        }
    }
}
```

Code di atas adalah code untuk halaman daftar yang menampilkan daftar tim dalam bentuk RecyclerView. DaftarFragment mengimplementasikan ClickListener untuk menangani klik pada item dalam daftar. Pertama, dalam metode onCreateView, layout fragment_daftar di-inflate menjadi rootView, dan RecyclerView dengan ID rv_teams ditemukan dan diinisialisasi. RecyclerView ini kemudian diatur agar memiliki ukuran tetap untuk meningkatkan performa. Data tim diambil dan diisi ke dalam array list menggunakan metode getListTeams. Metode ini membaca nama, deskripsi, dan gambar tim dari sumber daya (resources) aplikasi (data_name, data_desc, data_img) dan membuat objek Team untuk setiap tim yang kemudian ditambahkan ke dalam array list. Metode showRecyclerList digunakan untuk menampilkan daftar tim dalam RecyclerView. Di sini, layout manager diatur sebagai LinearLayoutManager agar item ditampilkan secara vertikal. Sebuah instance dari ListTeamAdapter dibuat dengan data tim yang telah dikumpulkan dan diatur sebagai adapter untuk RecyclerView. Adapter ini juga diatur agar memanggil metode onItemClick ketika sebuah item diklik, dengan meneruskan listener (this).

Fragment_daftar.xml



Pada halaman ini, hanya ada pengaturan untuk recycle view. Root layout dari halaman ini adalah linear layout. Dan di linear layout, terdapat pengaturan untuk recycle view dengan id rv_teams yang nantinya akan diproses.

Data class Team

```
package com.l0122006.afifimam.responsi1.ui.daftar

import ...

@Parcelize
data class Team(
    val name: String,
    val desc: String,
    val img: Int
) : Parcelable
```

Code di atas berisi data class dari Team. Data class Team ini digunakan untuk merepresentasikan data dari sebuah tim, yang terdiri dari tiga properti: name (nama tim) bertipe String, desc (deskripsi tim) bertipe String, dan img (resource ID untuk gambar tim) bertipe Int. Dan pada data class ini terdapat implementasi parcelize untuk interface.

List Team Adapter.kt

```
package com.l0122006.afifimam.responsi1.ui.daftar

import ...

class ListTeamAdapter(private val listTeam: ArrayList<Team>) : RecyclerView.Adapter<ListTeamAdapter.ListViewHolder>() {

    private lateinit var onItemClickCallback: OnItemClickListener

    interface OnItemClickListener {
        fun onItemClick(data: Team)
    }

    fun setOnItemClickCallback(onItemClickCallback: OnItemClickListener) {
        this.onItemClickCallback = onItemClickCallback
    }

    inner class ListViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
        val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
        val tvDesc: TextView = itemView.findViewById(R.id.tv_item_description)

        init {
            itemView.setOnClickListener { it: View! ->
                onItemClickCallback.onItemClick(listTeam[adapterPosition])
            }
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_row_team, parent, false)
        return ListViewHolder(view)
    }
}
```

Code di atas adalah code untuk daftar fragment. Fragment ini bertanggung jawab untuk menampilkan daftar tim dalam bentuk RecyclerView dan mengimplementasikan interaksi pengguna dengan item-item dalam daftar tersebut. Pertama, kelas DaftarFragment mewarisi dari Fragment dan mengimplementasikan antarmuka ListTeamAdapter.OnItemClickListener untuk menangani klik pada item dalam daftar. Dalam rootView, RecyclerView dengan ID rv_teams diinisialisasi dan diatur agar memiliki ukuran tetap (setHasFixedSize(true)) untuk meningkatkan performa. Selanjutnya, data tim ditambahkan ke dalam list dengan memanggil metode getListTeams. Metode ini membaca nama tim, deskripsi tim, dan resource ID untuk gambar tim dari sumber daya aplikasi (data_name, data_desc, data_img). Untuk setiap item, objek Team dibuat dan ditambahkan ke dalam list listTeam. Setelah selesai, TypedArray dataImg direcycle untuk menghindari kebocoran memori.

Metode showRecyclerViewList digunakan untuk menampilkan daftar tim dalam RecyclerView. Di sini, LinearLayoutManager diatur sebagai layout manager untuk RecyclerView, yang mengatur item ditampilkan secara vertikal. Sebuah instance dari ListTeamAdapter dibuat dengan data tim yang sudah dikumpulkan, dan adapter ini diatur untuk memanggil metode onItemClick ketika sebuah item diklik. Adapter kemudian diatur sebagai adapter untuk RecyclerView.

Detail Activity.kt

```
package com.l0122006.afifimam.responsil.ui.daftar

import ...

class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)

        val imgPhoto: ImageView = findViewById(R.id.img_team)
        val tvName: TextView = findViewById(R.id.tv_nama_team)
        val tvDesc: TextView = findViewById(R.id.tv_team_desc)

        val team = intent.getParcelableExtra<Team>(name: "EXTRA_TEAM") as Team

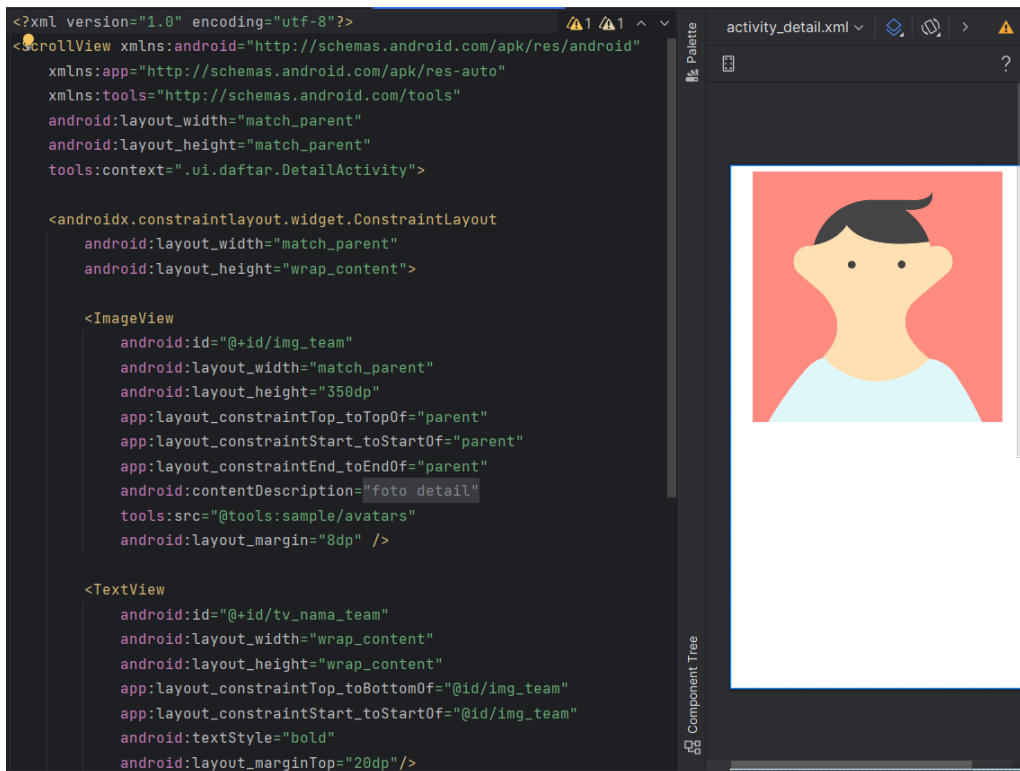
        imgPhoto.setImageResource(team.img)
        tvName.text = team.name
        tvDesc.text = team.desc
    }
}
```

Code di atas adalah code untuk menjalankan detail activit yang digunakan untuk menampilkan detail dari sebuah tim yang dipilih dari daftar di DaftarFragment. Pada class ini, terdapat onCreate, setContentView dipanggil dengan parameter R.layout.activity_detail untuk mengatur layout XML activity_detail sebagai tampilan utama dari aktivitas ini. Selanjutnya, elemen-elemen tampilan seperti ImageView dan TextView diinisialisasi dengan menggunakan findViewById. ImageView dengan ID img_team, TextView dengan ID tv_nama_team, dan TextView dengan ID tv_team_desc ditemukan dan dihubungkan ke variabel imgPhoto, tvName, dan tvDesc.

Kemudian, data Team diambil dari Intent yang memulai aktivitas ini. Data ini dikirim dari aktivitas atau fragment sebelumnya menggunakan kunci EXTRA_TEAM. Metode getParcelableExtra digunakan untuk mengambil objek Team yang diimplementasikan sebagai Parcelable, sehingga memungkinkan objek ini untuk ditransfer antar aktivitas. Objek Team yang diambil kemudian dicasting sebagai Team.

Setelah mendapatkan objek Team, data dari objek ini digunakan untuk memperbarui tampilan. imgPhoto.setImageResource(team.img) mengatur gambar tim di ImageView berdasarkan resource ID yang disimpan dalam objek Team. tvName.text = team.name dan tvDesc.text = team.desc mengatur teks pada TextView untuk menampilkan nama dan deskripsi tim.

Activity_detail.xml



Bagian ini adalah code untuk mengatur tampilan pada halaman detail activity. Tampilan menggunakan scroll view sebagai root karena untuk mengantisipasi teks yang Panjang. Selanjutnya, untuk mengatur posisi setiap text, menggunakan layout constraint. Dalam constraint terdapat ImageView dengan ID `img_team`, text view dengan id `tv_nama_team`, text view kedua dengan id `tv_team_desc`. Pada saat ini tidak ada tampilan apa apa karena data belum terkirim. Data akan terlihat saat pengiriman data sudah berjalan.

Profile Fragment.kt

```
class ProfileFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

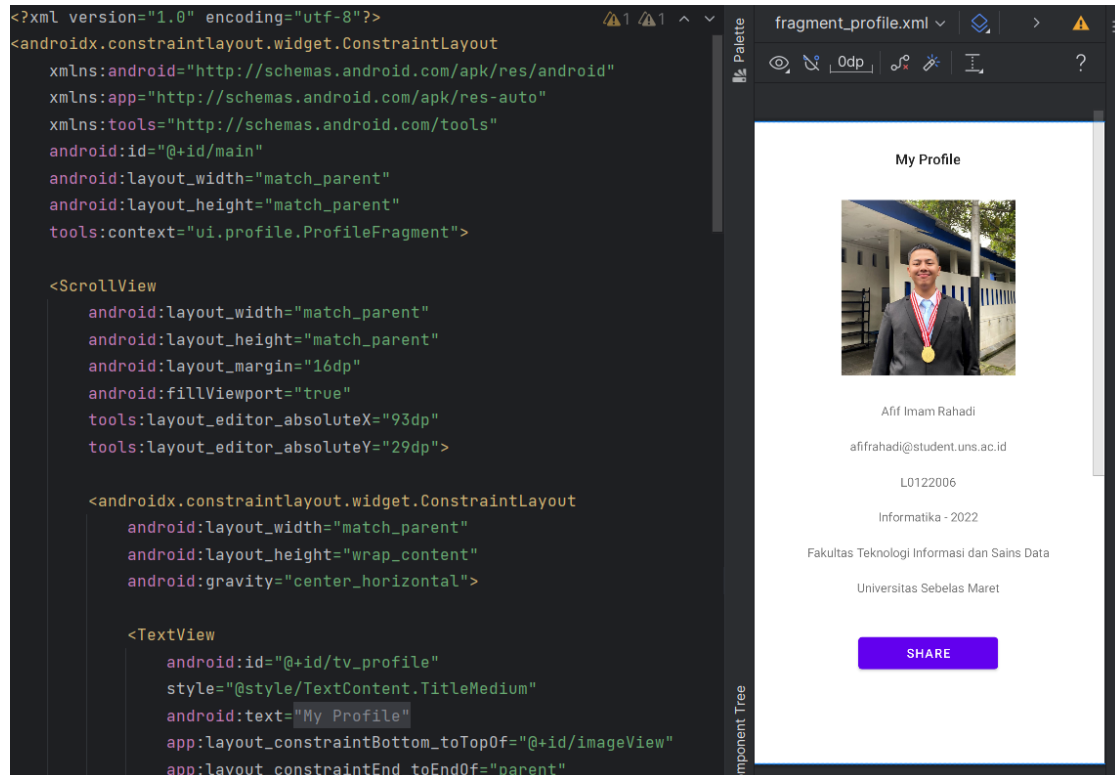
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let { it: Bundle
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val view = inflater.inflate(R.layout.fragment_profile, container, attachToRoot: false)

        // Set up the onClickListener for btn_share
        val btnShare: Button = view.findViewById(R.id.btn_share)
        btnShare.setOnClickListener { it: View?
            val intent = Intent(Intent.ACTION_SEND).apply { this: Intent
                type = "text/plain"
                putExtra(Intent.EXTRA_SUBJECT, value: "Name : Afif Imam Rahadi \n Nim : L0122006")
                putExtra(Intent.EXTRA_TEXT, value: "Name : Afif Imam Rahadi \n Nim : L0122006")
            }
            // Display a chooser for the user to select an app to share the content
            startActivity(Intent.createChooser(intent, title: "Share via"))
        }
    }
}
```

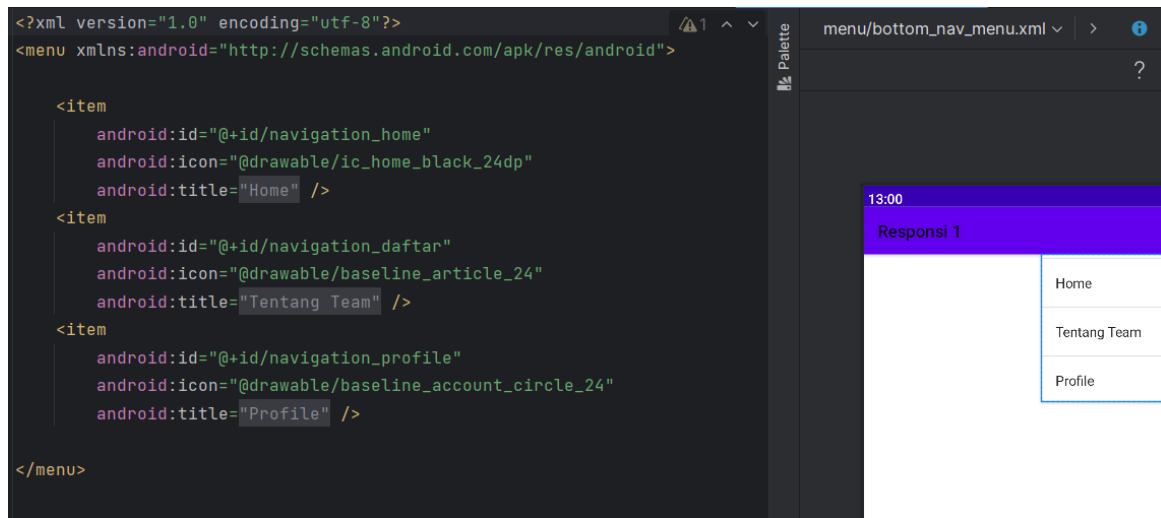
Code di atas adalah code untuk mengatur profile fragment. Pada bagian ini karena membuat fragment blank baru, maka masih ada sisaan code dari android itu. Untuk lebih detailnya, perhatikan method onCreateView karena pada method ini mengatur arah button dan juga pengiriman data dengan intent. Metode onCreateView meng-inflate layout XML fragment_profile dan mengembalikan tampilan yang dihasilkan sebagai View. Di dalam metode ini, sebuah tombol (btn_share) ditemukan dan diatur untuk menangani klik pengguna dengan menggunakan setOnClickListener. Ketika tombol btn_share diklik, sebuah Intent dengan aksi Intent.ACTION_SEND dibuat. Intent ini diatur untuk mengirim teks dengan tipe MIME text/plain. Dua ekstra ditambahkan ke dalam intent: Intent.EXTRA_SUBJECT dan Intent.EXTRA_TEXT, keduanya berisi informasi profil pengguna ("Name: Afif Imam Rahadi \n Nim: L0122006"). startActivity digunakan untuk memulai aktivitas baru dengan intent yang dibuat, dan Intent.createChooser ditambahkan untuk menampilkan pilihan aplikasi yang dapat digunakan untuk berbagi konten.

Fragment_profile.xml



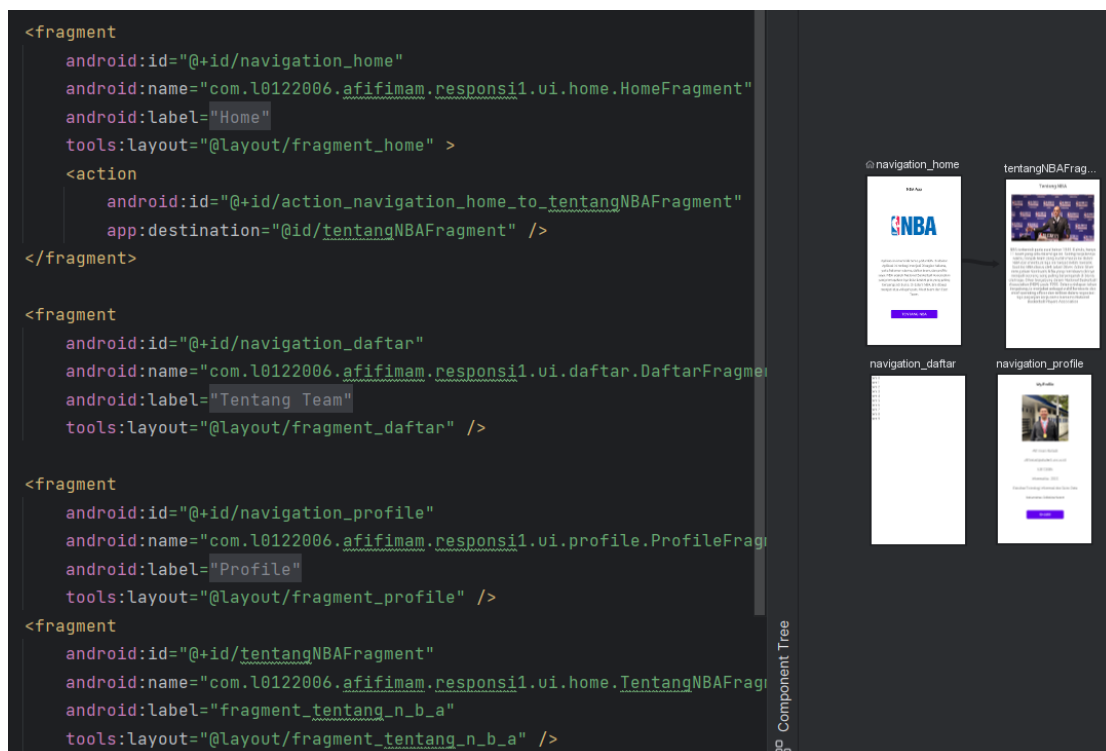
Ini adalah code untuk mengatur tampilan pada halaman profile. Halaman profile menggunakan constraint layout sebagai root layout. Dalam ConstraintLayout, terdapat ScrollView yang memungkinkan konten untuk digulir jika melebihi ukuran layar. ScrollView ini memiliki lebar dan tinggi yang sesuai dengan layar (match_parent) dan margin sebesar 16dp di sekelilingnya. Atribut android:fillViewport="true" memastikan bahwa ScrollView akan mengisi seluruh tampilan saat konten lebih sedikit dari ukuran layar. Elemen pertama dalam ConstraintLayout ini adalah TextView dengan ID tv_profile, yang menampilkan teks "My Profile". Berikutnya adalah ImageView dengan ID imageView, yang menampilkan gambar profil dengan sumber @drawable/foto_profile. Di bawah ImageView, terdapat beberapa TextView yang masing-masing menampilkan informasi profil seperti nama lengkap (tv_nama_lengkap), email (tv_email), NIM (tv_nim), program studi dan angkatan (tv_prodi_angkatan), fakultas (tv_fakultas), dan universitas (tv_univeritas). Semua TextView ini menggunakan gaya TextContent.Paragraf dan diatur secara vertikal di bawah elemen sebelumnya, serta di tengah secara horizontal. Terakhir, terdapat sebuah Button dengan ID btn_share yang memungkinkan pengguna untuk berbagi informasi profil. Tombol ini menggunakan gaya ButtonGeneral.Share dan ditempatkan di bawah tv_univeritas serta di tengah secara horizontal. Margin atas sebesar 20dp dan margin bawah sebesar 70dp ditambahkan di sekitar tombol.

Bottom nav menu.xml



Ini adalah bagian dari pengaturan bottom nav. Pada bottom nav terdapat 3 item yaitu home, daftar, dan profile. Pada setiap bottom nav terdapat gambar untuk setiap navigasinya dan setiap gambar sudah disimpan pada folder drawable.

Mobile Navigation



Ini adalah bagian pengaturan untuk arah dari setiap fragment. Khusus pada bagian fragment home, terdapat action ke dalam fragment tentang NBA. Ini karena terdapat button yang mengarah ke halaman tersebut. Untuk yang lainnya, hanya deklarasi setiap fragment untuk mengatur halaman pada setiap bottom navigasinya.

2. Screenshot Emulator

Berikut adalah hasil pada emulator apabila source code dijalankan.

Halaman Utama



Halaman utama merupakan halaman home. Pada halaman ini terdapat judul aplikasi di atas, foto, dan deskripsi singkat. Pada bagian bawah, terdapat button yang mengarah ke fragment lainnya. Dan terakhir, halaman ini menggunakan bottom navigation sehingga dapat berubah halaman dengan mengklik icon di bottom navigation.

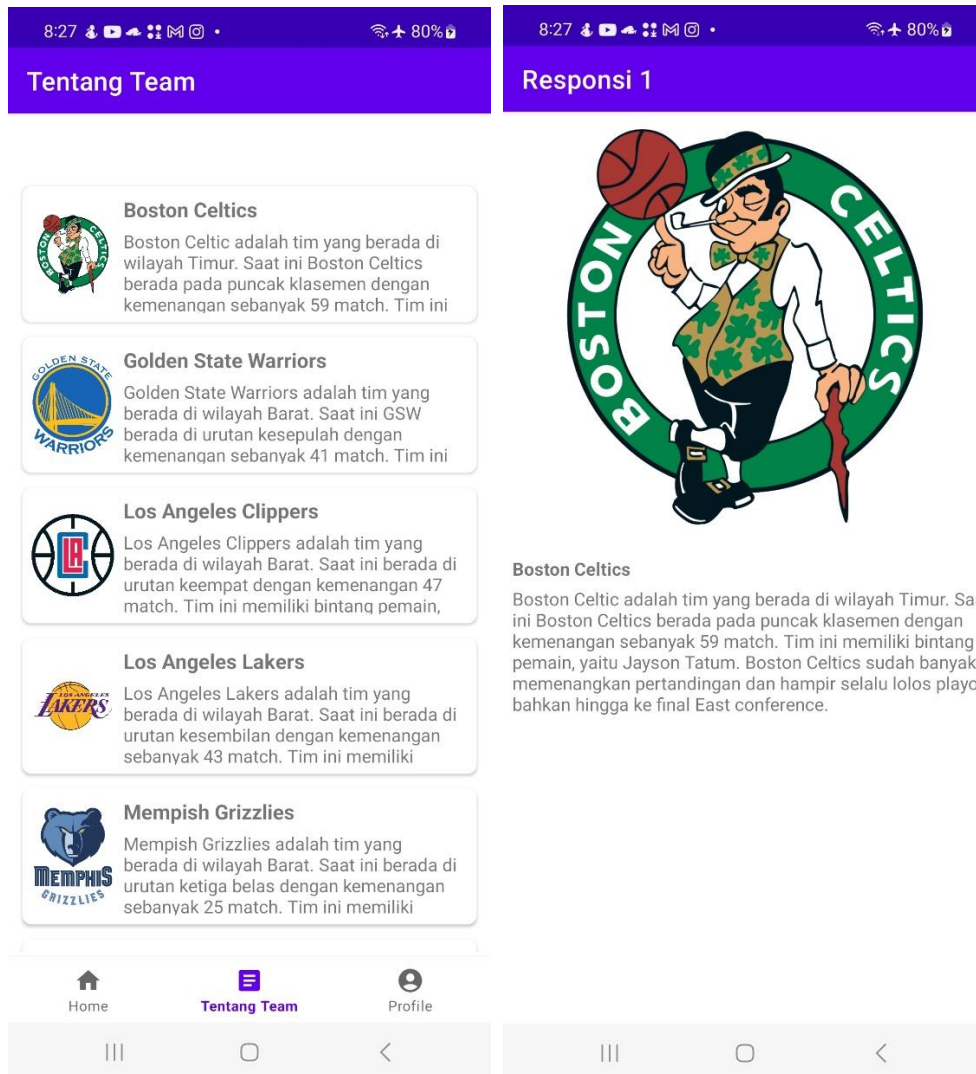
Halaman Tentang NBA



NBA terbentuk pada awal tahun 1949. Dahulu, hanya 11 team yang ada dalam liga ini. Seiring berjalannya waktu, banyak team yang sudah masuk ke dalam NBA dan membuat liga ini menjadi lebih menarik. Saat ini, NBA diurus oleh Adam Silver. Adam Silver merupakan Komisaris NBA yang membawa dirinya menjadi seorang yang paling berpengaruh di bisnis olahraga. Silver bergabung dalam National Basketball Association (NBA) pada 1992. Selama delapan tahun bergabung, ia menjabat sebagai wakil komisaris dan chief operating officer dan terlibat dalam negosiasi tiga perjanjian kerja sama bersama National Basketball Players Association

Halaman ini merupakan halaman lanjutan dari halaman home. Halaman ini menggunakan fragment tambahan. Pada halaman ini terdapat foto dari pengurus NBA bernama Adam Silver dan sedikit deksirpsi lebih detail terkait NBA. Dan pada bagian bawah, sama seperti bagian home, terdapat bottom navigation.

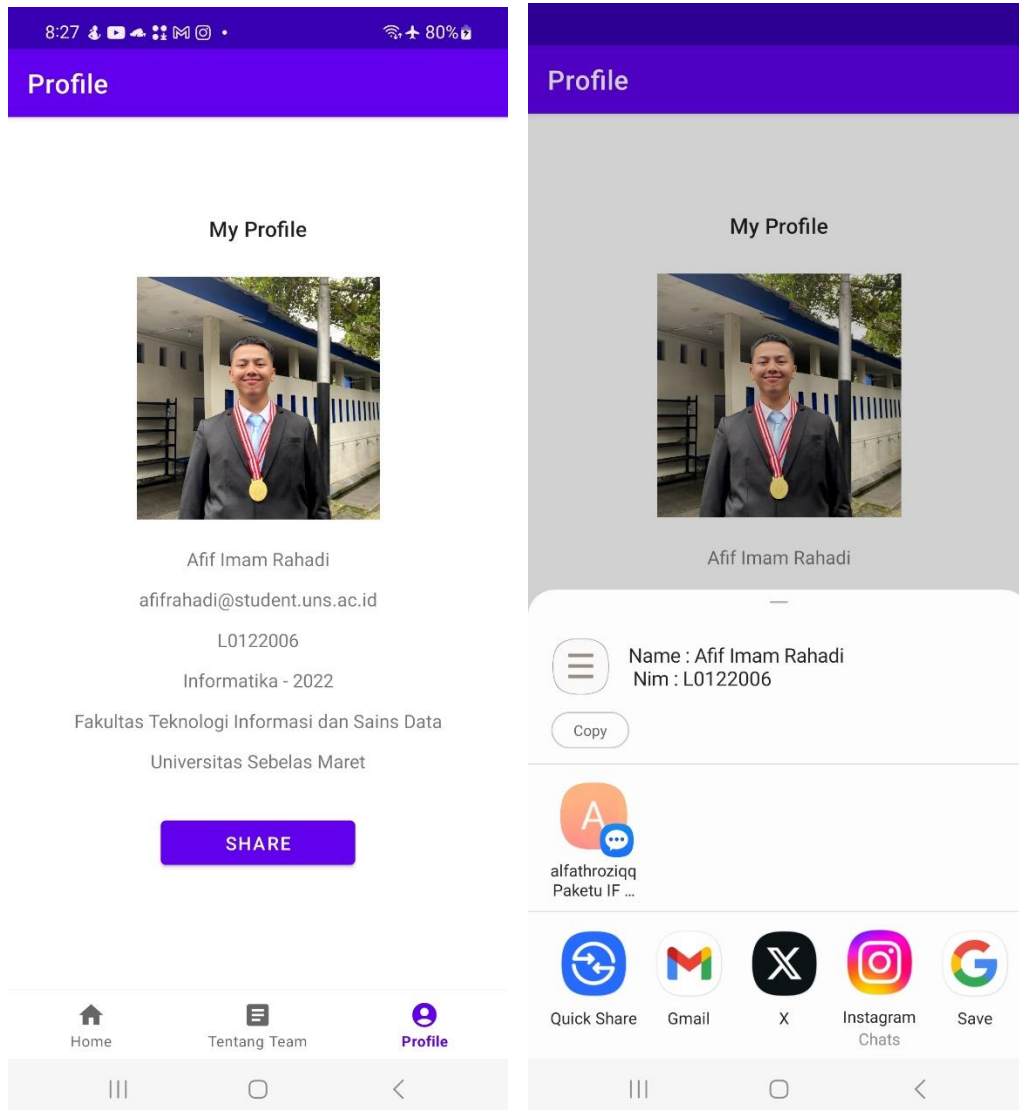
List Team Dan Detail Team



Pada halaman ini, terdapat list dari 10 team yang berada pada NBA. List ini terdiri dari foto team, nama team, dan deskripsi singkat. Tampilan pada bagian ini menggunakan recycle view dan cardview sehingga setiap data dibungkus oleh cardview.

Pada foto selanjutnya, adalah foto dari detail pada list team. Pada halaman list team, terdapat onclick yang mengarah ke activity detail sehingga muncul data seperti gambar bagian kanan. Data yang ditampilkan diambil dari list team sehingga mengirim data menggunakan inten. Data berupa gambar team, nama team, dan deskripsi secara lengkap. Pada bagian detail ini, tampilan berupa scroll view sehingga jika deskripsi lebih banyak, user dapat melakukan scroll pada halaman detail.

Halaman Profile



Halaman ini adalah halaman dari profile saya. Di dalam halaman ini, terdapat data data pribadi berupa nama lengkap, email nim, prodi, angkatan, fakultas dan universitas. Dan pada bagian bawah, terdapat button untuk melakukan share data diri. Saya membuat share ini mempunyai value sehingga subject dan isinya akan otomatis terisi dengan nama lengkap dan juga nim saya.

3. Kesimpulan

Kesimpulan pada responsi kali ini adalah setiap materi yang diajarkan pada praktikum 3-6 semuanya terpakai pada responsi 1. Mulai dari pembelajaran mengenai pengenalan android seperti memasukkan tulisan menggunakan textview, gambar dengan imageview, dan lain-lain. Pada awal pengenalan juga terdapat materi pengiriman data dengan intent yang digunakan pada fitur button share dan juga untuk masuk ke halaman detail activity.

Tidak hanya itu, responsi ini juga menggunakan project bottom navigation yang sebelumnya sudah diajarkan. Dan pengampilkasian halaman menggunakan fragment. Terdapat juga action dari fragment home ke dalam activity detail. Selain itu, materi list menggunakan recycle view juga diaplikasikan pada halaman daftar team sehingga dapat menampilkan 10 data sesuai dengan tema aplikasi yang dibuat. Dalam recycle view juga menerapkan pengaplikasian layout dengan cardview sehingga template list seragam dan terlihat lebih rapih dan terstruktur.