

Berita.Ku

```
1 package com.project.beritaku.di
2
3 > import ...
4
21 val datastoreModule = module { this: Module
22     single { SettingsPreferences(androidContext().dataStore) }
23 }
24
25
26 val networkModule = module { this: Module
27     single { ApiConfig.getApiService() }
28     single { RemoteDataSource(get()) }
29 }
30
31 val repositoryModule = module { this: Module
32     single<NewsRepository> { NewsRepositoryImpl(get()) }
33     single<SettingsRepository> { SettingsRepositoryImpl(get()) }
34 }
35
36 val useCaseModule = module { this: Module
37     single<NewsUseCase> { NewsUseCaseImpl(get()) }
38     single<SettingsUseCase> { SettingsUseCaseImpl(get()) }
39 }
40
41 val viewModelModule = module { this: Module
42     viewModel { SplashViewModel(get()) }
43     viewModel { HomeViewModel(get()) }
44     viewModel { SettingsViewModel(get()) }
45 }
```

```
package com.project.beritaku.data
> import ...
@Suppress("UNCHECKED_CAST")
abstract class NetworkBoundResource<ResultType, RequestType> {
    private var result: Flow<Resource<ResultType>> = flow { this: FlowCollector<Resource<ResultType>>
        emit(Resource.Loading())
        when (val apiResponse = createCall().first()) {
            is ApiResponse.Success -> {
                emit(
                    Resource.Success(apiResponse.data as ResultType)
                )
            }
            is ApiResponse.Empty -> {
                emit(
                    Resource.Success(apiResponse as ResultType)
                )
            }
            is ApiResponse.Error -> {
                emit(
                    Resource.Error(apiResponse.errorMessage)
                )
            }
        }
    }
    protected abstract suspend fun createCall(): Flow<ApiResponse<RequestType>>
    fun asFlow(): Flow<Resource<ResultType>> = result
}
```

```
package com.project.beritaku.ui.adapter
> import ...
class NewsAdapter :
    ListAdapter<ArticlesItem, NewsAdapter.MyViewHolder>(DIFF_CALLBACK) {
    var onItemClickListener: ((ArticlesItem) -> Unit)? = null
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) = MyViewHolder(
        ItemNewsContainerBinding.inflate(
            LayoutInflater.from(parent.context),
            parent,
            attachToParent = false
        )
    )
    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        getItem(position)?.let { it: ArticlesItem
            holder.bind(it)
        }
    }
    inner class MyViewHolder(private val binding: ItemNewsContainerBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(news: ArticlesItem) {
            binding.apply { this: ItemNewsContainerBinding
                Glide.with(context).load(news.urlToImage).requestBuilder<Drawable>()
                    .placeholder(R.drawable.dummy_image)
                    .transition(DrawableTransitionOptions.withCrossFade())
                    .into(ivNews)
                tvNewsTitle.text = news.title
                tvTime.text = dateFormatter(news.publishedAt)
                root.setOnClickListener { it: View
                    onItemClickListener?.invoke(news)
                }
            }
        }
    }
    companion object {
        val DIFF_CALLBACK = object : DiffUtil.ItemCallback<ArticlesItem>() {
            override fun areItemsTheSame(
                oldItem: ArticlesItem,
                newItem: ArticlesItem
            ): Boolean {
                return oldItem == newItem
            }
        }
    }
}
```

Deskripsi :

Berita.Ku adalah aplikasi berita mobile yang dirancang untuk memberikan pengalaman membaca berita yang personal dan terkini bagi penggunanya. Dengan antarmuka yang intuitif dan user-friendly, Berita.Ku memungkinkan pengguna untuk mengakses berbagai kategori berita, mulai dari politik, ekonomi, teknologi, hiburan, hingga olahraga.

Progress & Kendala :

Untuk kendala yang dialami sejauh ini terdapat pada pembuatan Bottom Menu Navigation yang dimana harus setup menu dengan menghubungkan resource navigation ke fragment-fragment yang dituju.