



MOBILE APPS DEVELOPER

BANK MANDIRI

Muhammad Afrizal Bagaskara

CONTENT

Introduction

About Task 5

Demo

Application

Summary



INTRODUCTION

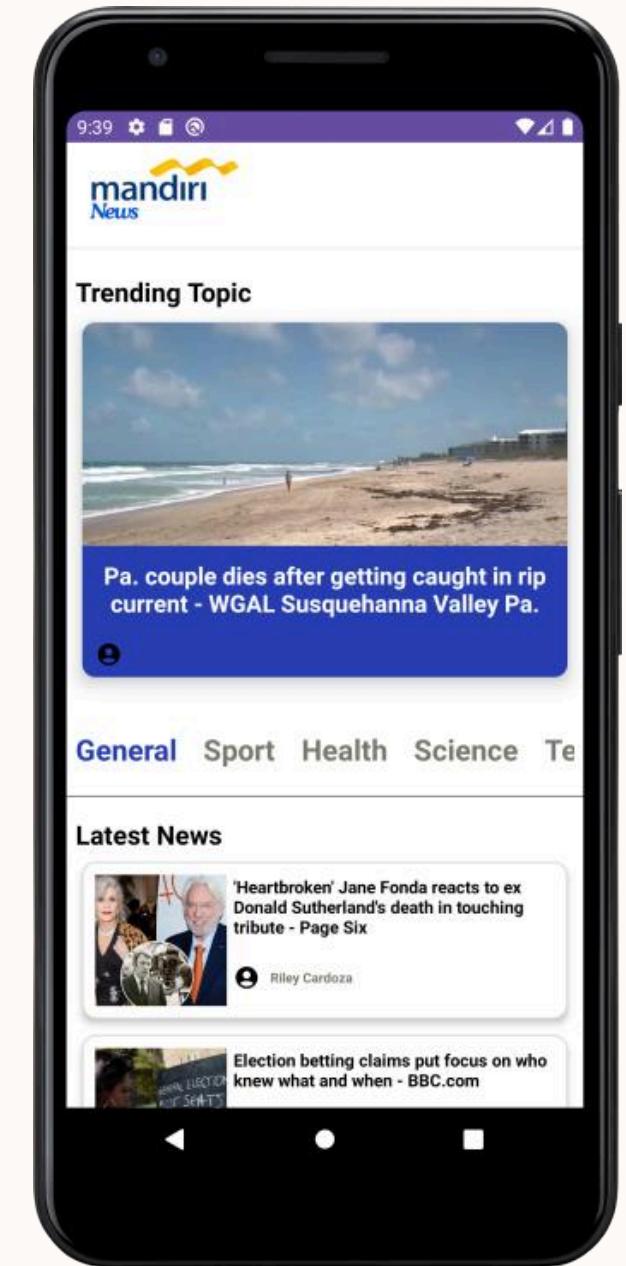
Halo, nama saya Muhammad Afrizal Bagaskara. Saya seorang freshgraduate yang lulus dari universitas Sriwijaya. Saya mengikuti program Bank Mandiri Mobile Apps Developer ini untuk memperdalam ilmu tentang Mobile Apps Developer karena memang saya tertarik dengan Mobile Apps Developer. Dengan mengikuti program ini saya memperoleh pengalaman serta portofolio yang berharga.

TENTANG TASK 5

Dengan membuat list berita dari task 5 ini saya jadi paham bagaimana cara menggunakan API dan mengimplementasikannya ke dalam android studio.

DEMO APLIKASI

Gambar di samping adalah tampilan home dari app list film yang saya buat. Selanjutnya saya akan menjelaskan codenya.



Menambahkan Dependencies di gradle

```
implementation("androidx.core:core-splashscreen:1.0.0")
```

Untuk pustaka dari *AndroidX* untuk menangani layar splash di aplikasi *Android*

```
implementation("com.squareup.retrofit2:retrofit:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.9.0")
implementation("com.squareup.okhttp3/logging-interceptor:4.8.1")
```

1. *Retrofit (2.9.0)* untuk mengirim dan menerima data dari server melalui *HTTP*, sangat berguna untuk interaksi dengan *API RESTful*.
2. *Converter-Gson (2.9.0)* untuk *Retrofit* mengkonversi *JSON* ke objek *Kotlin*
3. *Logging Interceptor (4.8.1)* dari *OkHttp* mencatat rincian permintaan dan respons *HTTP*, membantu dalam debugging dengan menyediakan log lengkap komunikasi jaringan.

Menambahkan Dependencies di gradle

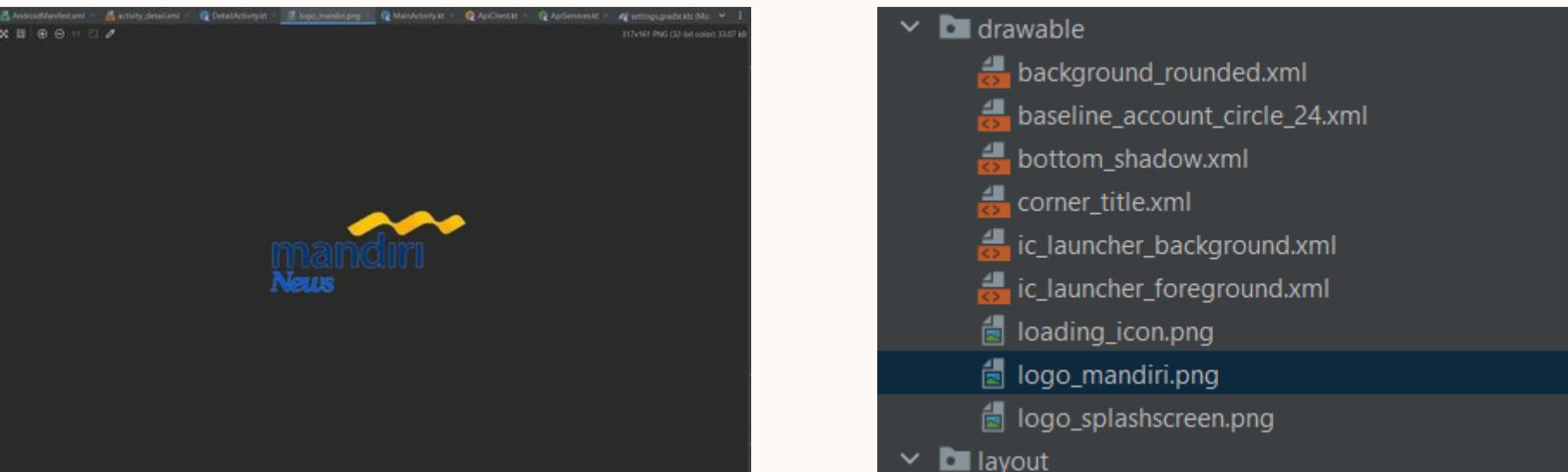
```
implementation("com.github.bumptech.glide:glide:4.10.0")  
  
implementation("com.google.android.material:material:1.2.1")
```

Menambahkan color

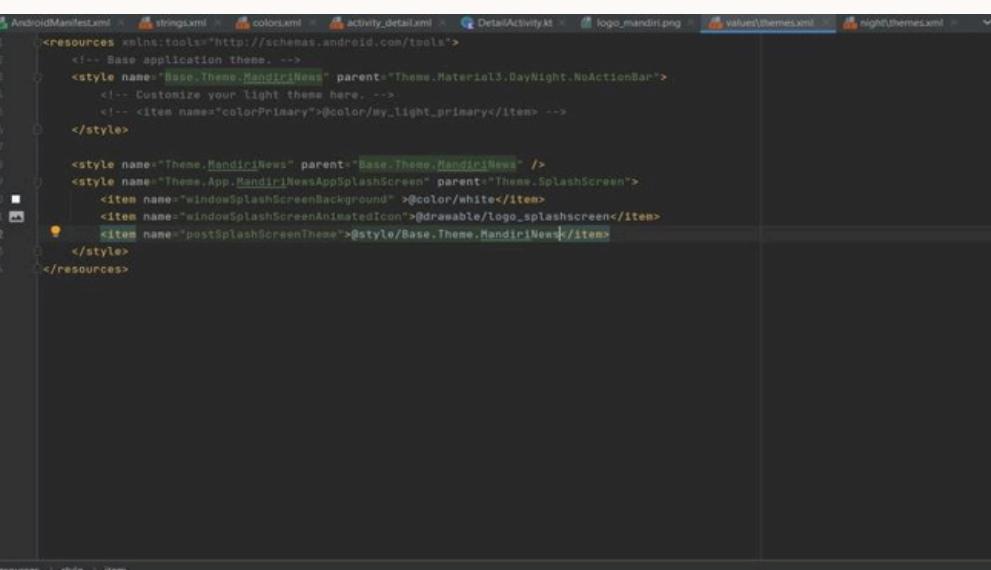
```
1  <?xml version="1.0" encoding="utf-8"?>  
2  <resources>  
3      <color name="black">#FF000000</color>  
4      <color name="white">#FFFFFF</color>  
5  
6      <color name="blue">#283EB3</color>  
7      <color name="gray">#7A7A6E</color>  
8  </resources>
```



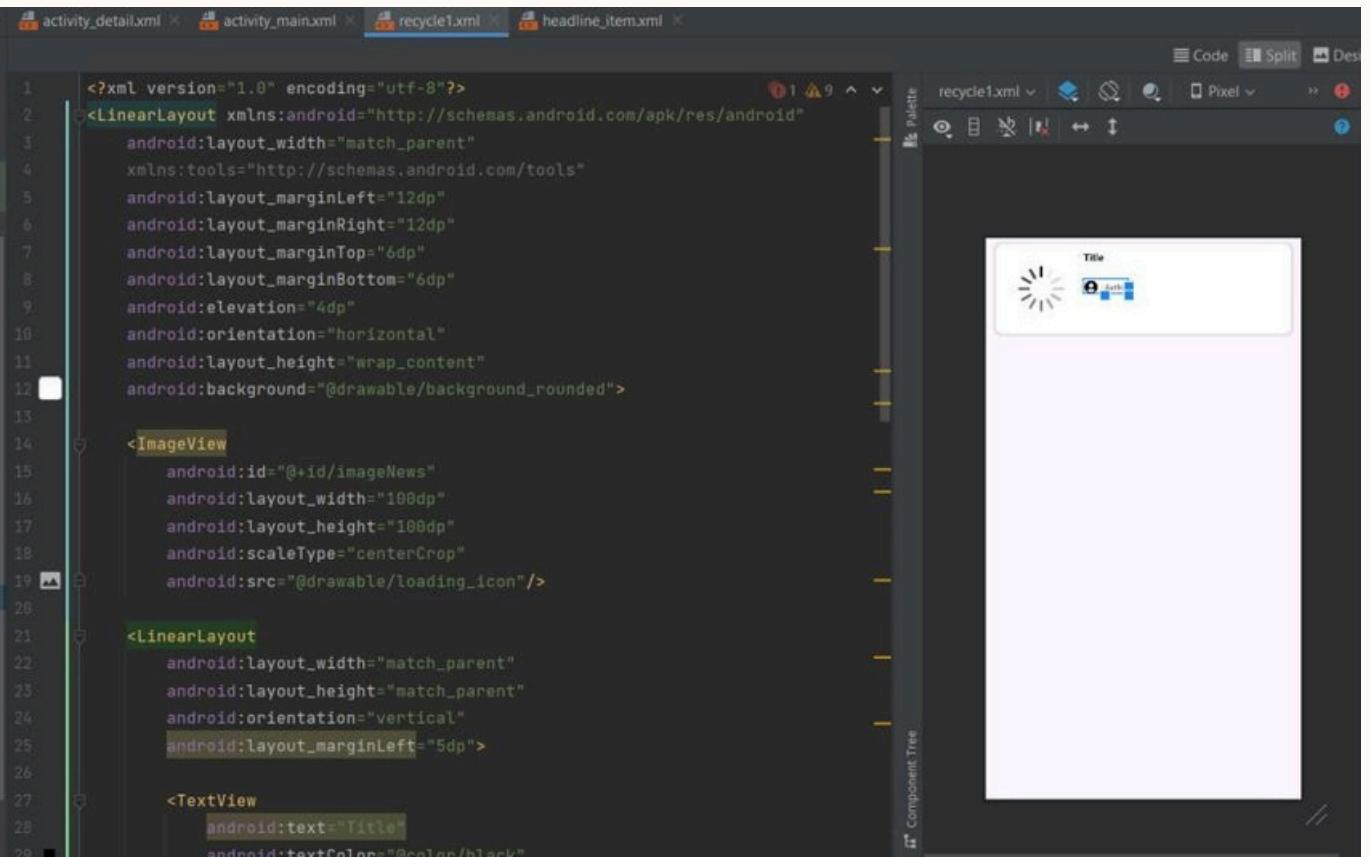
Menambahkan Logo dan komponen lainnya di drawable



Membuat Theme untuk splashscreen



Membuat layout Desain recycle1.xml



The screenshot shows the Android Studio interface with the XML code for `recycle1.xml`. The code defines a layout structure with a title bar, an image view, a linear layout containing a title text view, and another linear layout with an author icon and text view. The preview window shows a white card with a circular loading icon at the top, followed by a section with a blue icon and some placeholder text, and a bottom section with a user icon and text.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_marginLeft="12dp"
    android:layout_marginRight="12dp"
    android:layout_marginTop="6dp"
    android:layout_marginBottom="6dp"
    android:elevation="4dp"
    android:orientation="horizontal"
    android:layout_height="wrap_content"
    android:background="@drawable/background_rounded">

    <ImageView
        android:id="@+id/imageNews"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:scaleType="centerCrop"
        android:src="@drawable/loading_icon"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_marginLeft="5dp">

        <TextView
            android:text="Title"
            android:textColor="@color/black"/>

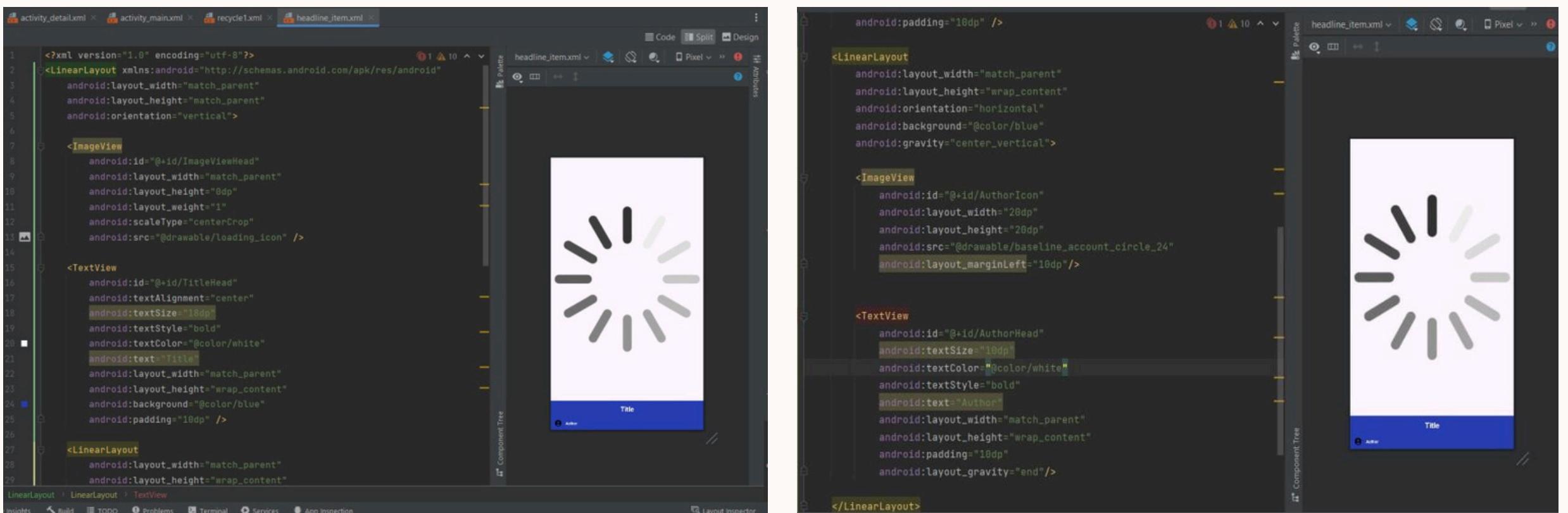
        <LinearLayout
            android:id="@+id/AuthorLayout"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="20dp"
            android:gravity="center_vertical">

            <ImageView
                android:id="@+id/AuthorIcon"
                android:layout_width="20dp"
                android:layout_height="20dp"
                android:src="@drawable/baseline_account_circle_24"/>

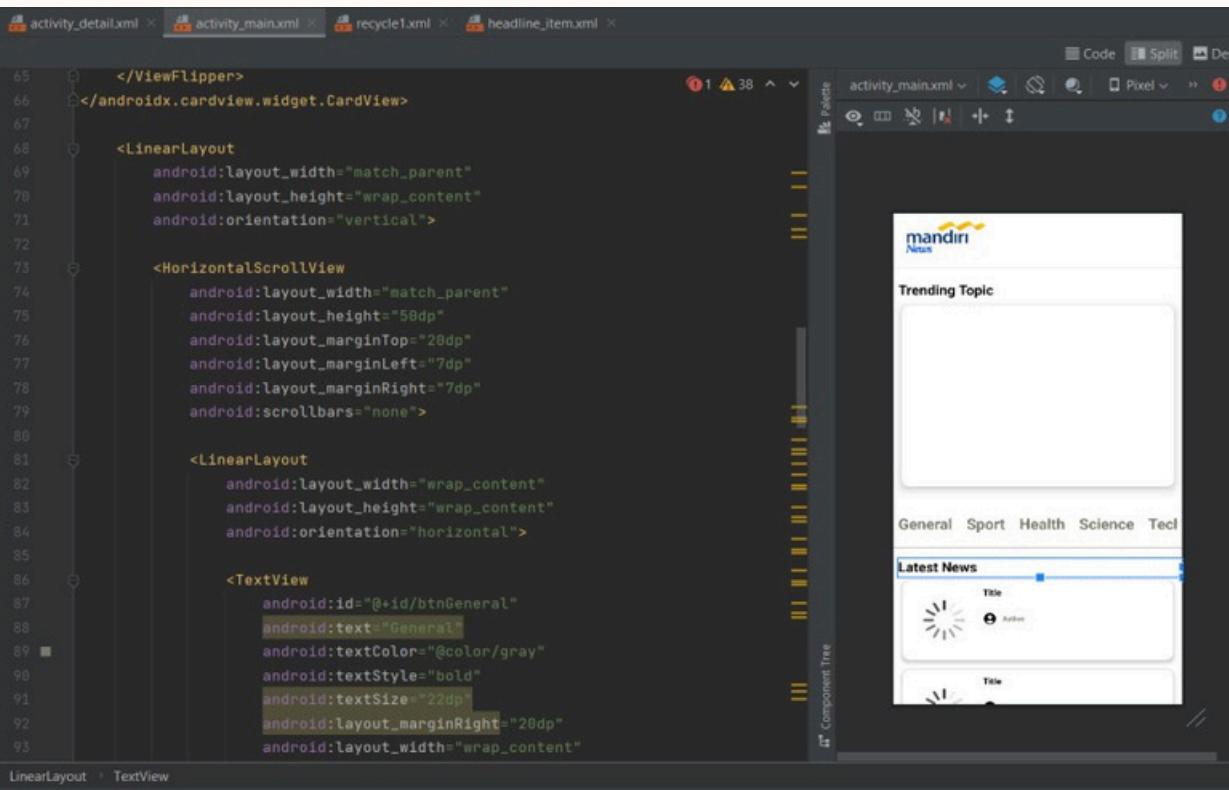
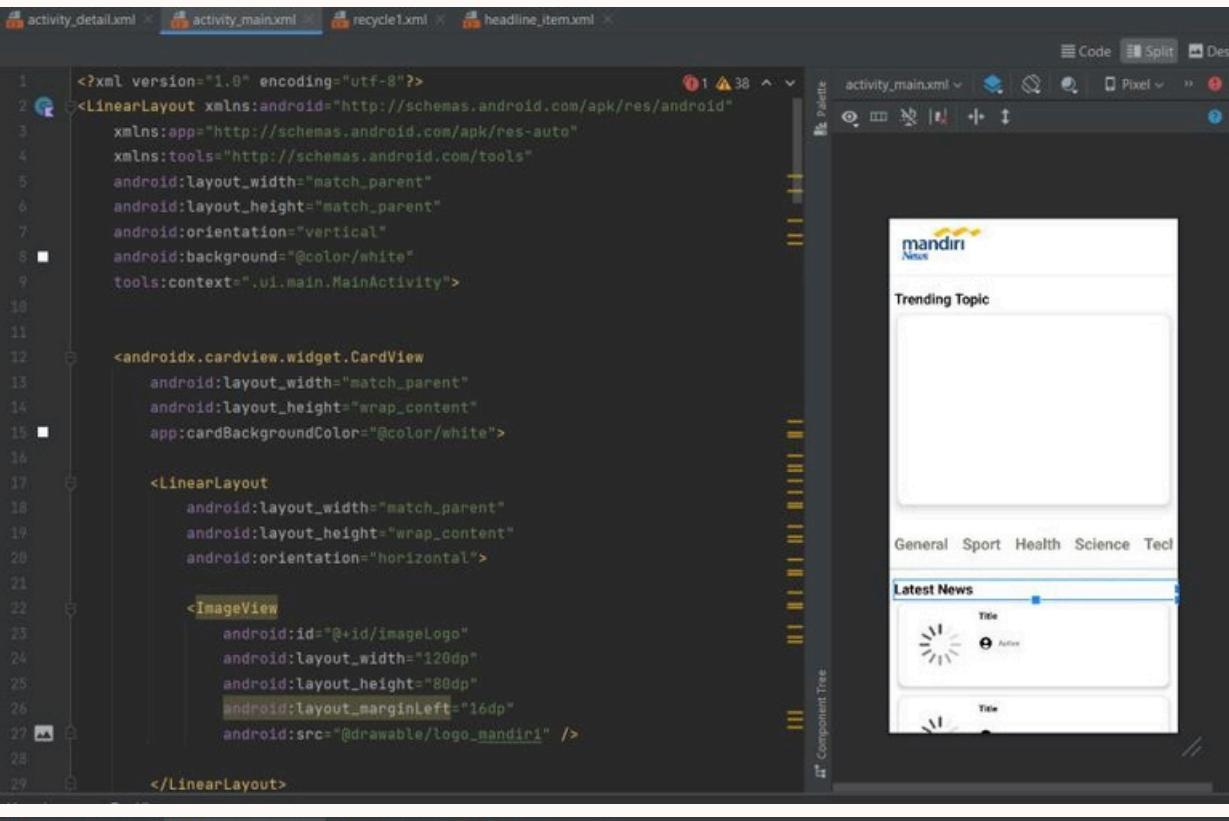
            <TextView
                android:text="Author"
                android:textColor="@color/gray"
                android:id="@+id/tvSource"/>
        
    

```

Membuat layout Desain headline_item.xml



Membuat layout Desain activity_main.xml



```

</LinearLayout>
</androidx.cardview.widget.CardView>

<TextView
    android:text="Trending Topic"
    android:textColor="@color/black"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="10dp"
    android:textSize="20dp"
    android:textStyle="bold"
    android:layout_marginLeft="7dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<androidx.cardview.widget.CardView
    android:id="@+id/Headline"
    android:layout_width="match_parent"
    android:layout_height="270dp"
    android:layout_marginBottom="20dp"
    android:layout_marginLeft="12dp"
    android:layout_marginRight="12dp"
    app:cardCornerRadius="10dp"
    app:cardBackgroundColor="@color/white"
    app:cardElevation="10dp">

    <ViewFlipper
        android:id="@+id/viewFlipper">

```



mandiri

Membuat layout Desain activity_detail.xml

A screenshot of the Android Studio interface. On the left is the XML code editor for 'activity_detail.xml'. The code defines a CardView containing a WebView and an ImageView. On the right is the 'Preview' tab, showing a white screen with a small 'mandiri News' logo at the top. The bottom right corner shows a 'Component Tree' icon.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ui.detail.DetailActivity"
    android:background="@color/white">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/navbar"
        app:cardBackgroundColor="@color/white">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageLogo"
                android:layout_width="128dp"
                android:layout_height="88dp"
                android:layout_marginLeft="16dp"
                android:src="@drawable/logo_mandiri" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <WebView
        android:id="@+id/webView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/navbar"
        android:visibility="gone" />
</LinearLayout>
```

A detailed view of the XML code from the 'activity_detail.xml' file, focusing on the logo section. It shows the ImageView with its specific dimensions, margins, and source image. The code is presented in a monospaced font with some parts highlighted in green and yellow.

```
        android:layout_width="128dp"
        android:layout_height="88dp"
        android:layout_marginLeft="16dp"
        android:src="@drawable/logo_mandiri" />

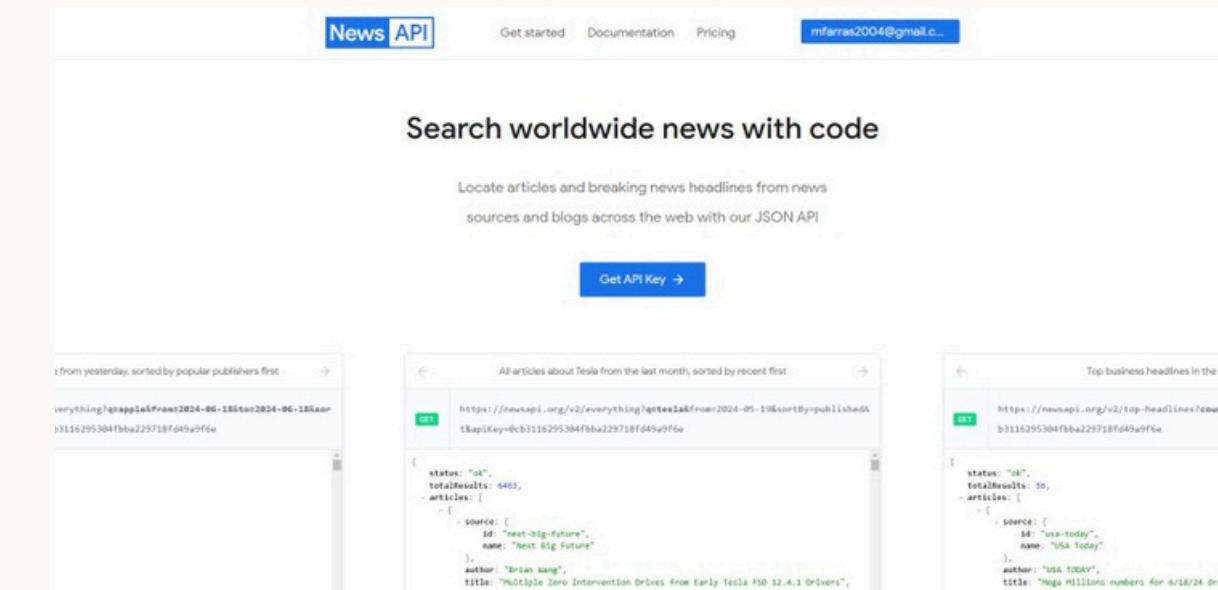
    </LinearLayout>
</androidx.cardview.widget.CardView>

<WebView
    android:id="@+id/webView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/navbar"
    android:visibility="gone" />

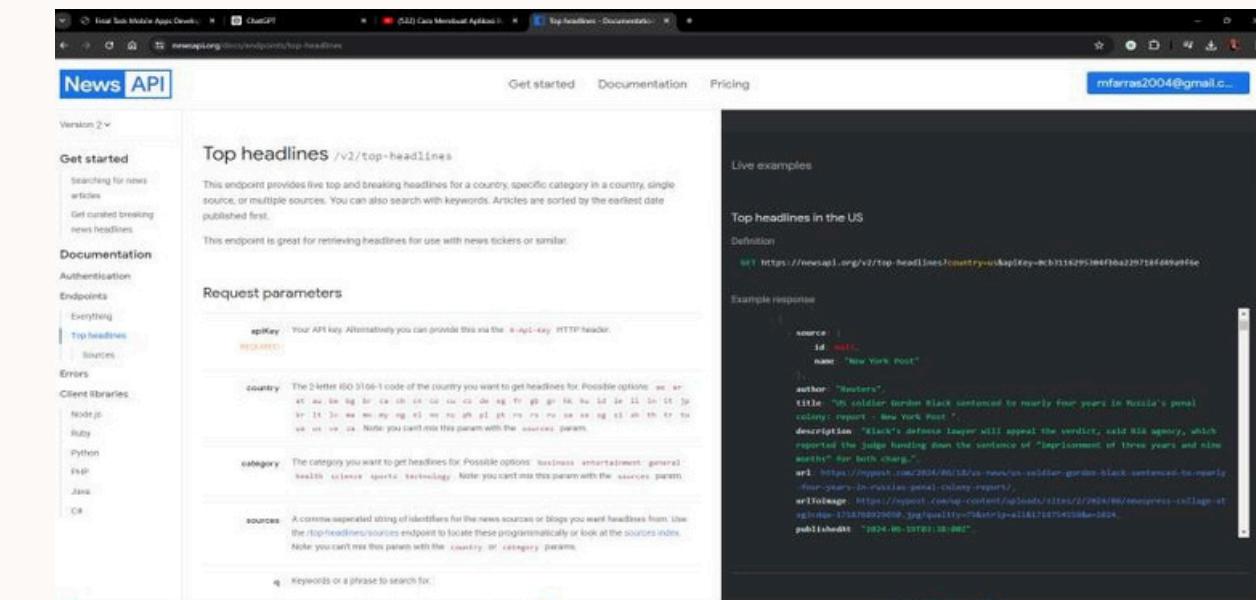
</LinearLayout>
```

Memperoleh berita dari newsAPI

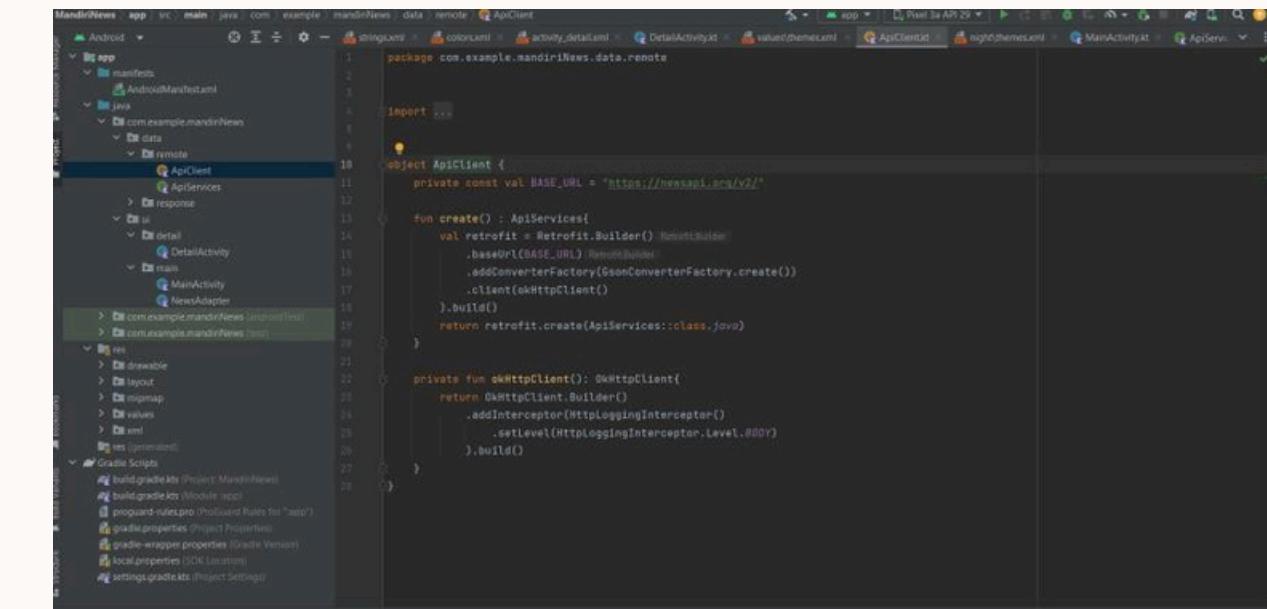
1. Login di newAPI dengan email dan setelah login klik documentation yang ada di navbar



2. Klik top headline untuk mendapatkan source api key



3. Membuat object ApiClient



```

package com.example.mandiriNews.data.remote

import ...

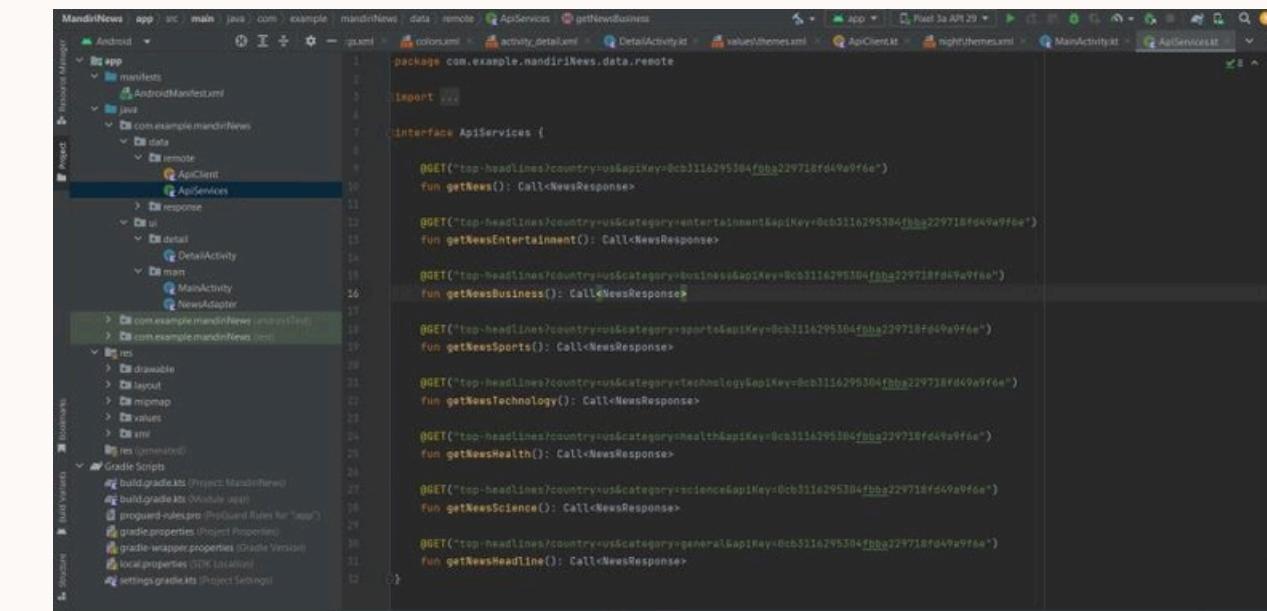
object ApiClient {
    private const val BASE_URL = "https://newsapi.org/v2/"

    fun create(): ApiServices {
        val retrofit = Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .client(okHttpClient)
        ).build()
        return retrofit.create(ApiServices::class.java)
    }

    private fun okHttpClient(): OkHttpClient {
        return OkHttpClient.Builder()
            .addInterceptor(HttpLoggingInterceptor())
            .setLevel(HttpLoggingInterceptor.Level.BODY)
        ).build()
    }
}

```

4. Membuat Interface ApiService untuk mendapatkan link top headline dan bisa didapatkan berdasarkan country dan category.



```

package com.example.mandiriNews.data.remote

interface ApiService {
    @GET("top-headlines?country=us&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNews(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=entertainment&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsEntertainment(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=business&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsBusiness(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=sports&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsSports(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=technology&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsTechnology(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=health&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsHealth(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=science&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsScience(): Call<NewsResponse>

    @GET("top-headlines?country=us&category=general&apiKey=0cb3116295304f0ba229718fd49a9f6e")
    fun getNewsHeadline(): Call<NewsResponse>
}

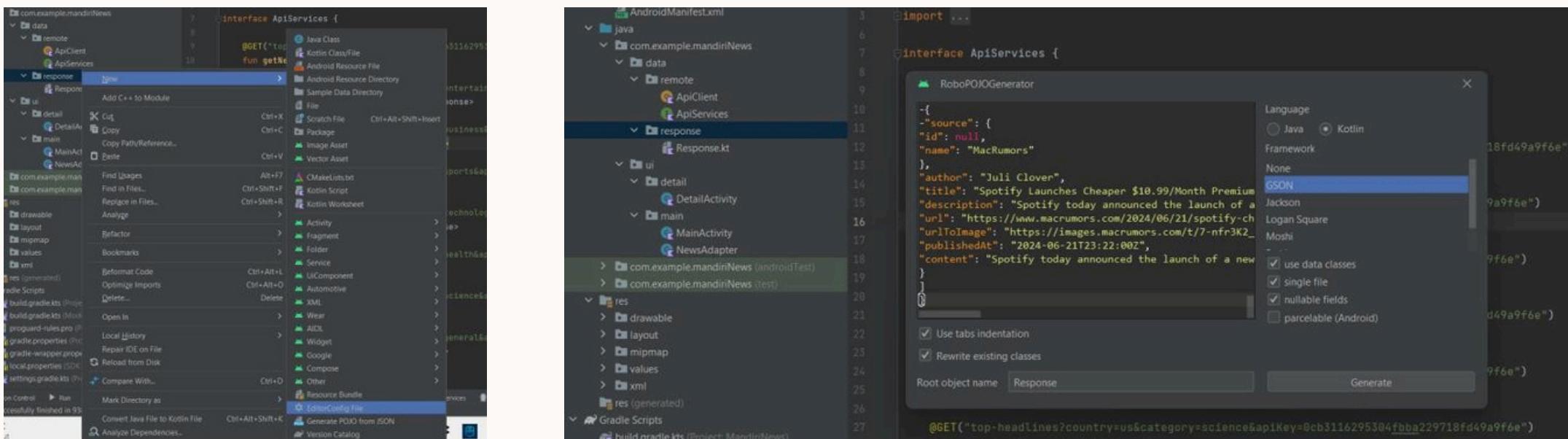
```

5. Copy semua example response

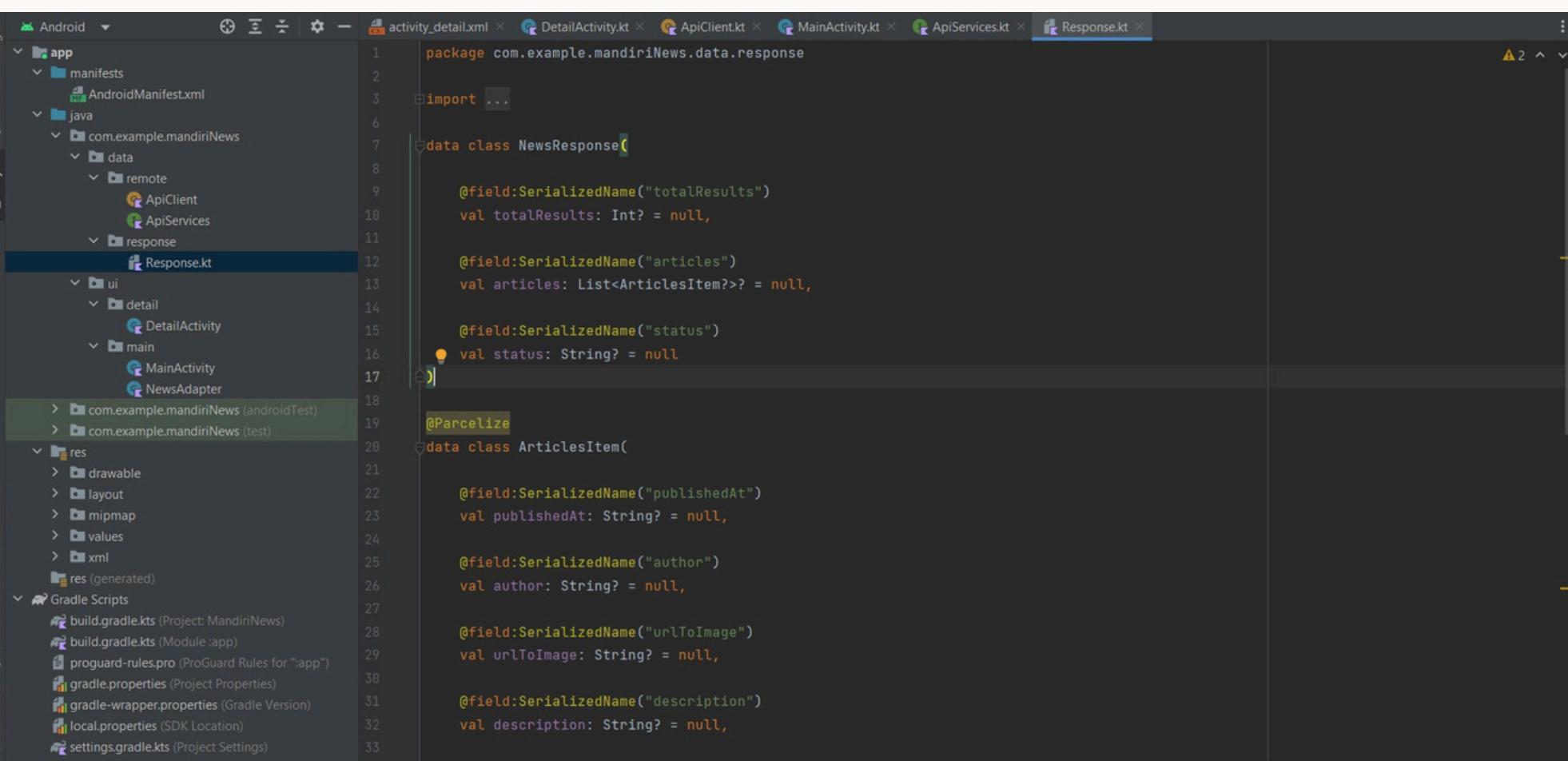
Example response

```
{
  status: "ok",
  totalResults: 36,
  articles: [
    {
      source: {
        id: null,
        name: "BBC News"
      },
      author: null,
      title: "ICRC says 22 killed in strike near its Gaza office - BBC.com",
      description: "The International Committee of the Red Cross said the \"grave security incident\" was the latest of several."
    }
  ]
}
```

6. Kemudian paste di generate to pojo from json



7. Maka hasil dari generatenya seperti ini



The screenshot shows the Android Studio interface with the code editor open to the `Response.kt` file. The code defines two data classes: `NewsResponse` and `ArticlesItem`. The `NewsResponse` class has fields for totalResults, articles, and status. The `ArticlesItem` class has fields for publishedAt, author, urlToImage, and description. Both classes are annotated with `@Parcelize`.

```
1 package com.example.mandiriNews.data.response
2
3 import ...
4
5 data class NewsResponse(
6     @field:SerializedName("totalResults")
7     val totalResults: Int? = null,
8
9     @field:SerializedName("articles")
10    val articles: List<ArticlesItem?>? = null,
11
12     @field:SerializedName("status")
13     val status: String? = null
14 )
15
16
17 @Parcelize
18 data class ArticlesItem(
19
20     @field:SerializedName("publishedAt")
21     val publishedAt: String? = null,
22
23     @field:SerializedName("author")
24     val author: String? = null,
25
26     @field:SerializedName("urlToImage")
27     val urlToImage: String? = null,
28
29     @field:SerializedName("description")
30     val description: String? = null,
31
32
33 )
```

8. Buat function getNews di mainActivity untuk mengambil link berita dari interface apiServices

```
private fun getNews(category: String) {
    val call: Call<NewsResponse> = when (category) {
        "sports" -> ApiClient.create().getNewsSports()
        "health" -> ApiClient.create().getNewsHealth()
        "science" -> ApiClient.create().getNewsScience()
        "technology" -> ApiClient.create().getNewsTechnology()
        "business" -> ApiClient.create().getNewsBusiness()
        "entertainment" -> ApiClient.create().getNewsEntertainment()
        else -> ApiClient.create().getNews() // Default to general news
    }

    call.enqueue(object : Callback<NewsResponse> {
        override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {
            if (response.isSuccessful) {
                val articles: List<ArticlesItem> = response.body()?.articles as List<ArticlesItem>
                adapter.setNews(articles)
            }
        }

        override fun onFailure(call: Call<NewsResponse>, t: Throwable) {
            // Handle failure
        }
    })
}
```

9. Buat function getNews di mainActivity untuk mengambil link berita dari interface apiServices

```
private fun getNews(category: String) {
    val call: Call<NewsResponse> = when (category) {
        "sports" -> ApiClient.create().getNewsSports()
        "health" -> ApiClient.create().getNewsHealth()
        "science" -> ApiClient.create().getNewsScience()
        "technology" -> ApiClient.create().getNewsTechnology()
        "business" -> ApiClient.create().getNewsBusiness()
        "entertainment" -> ApiClient.create().getNewsEntertainment()
        else -> ApiClient.create().getNews() // Default to general news
    }

    call.enqueue(object : Callback<NewsResponse> {
        override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {
            if (response.isSuccessful) {
                val articles: List<ArticlesItem> = response.body()?.articles as List<ArticlesItem>
                adapter.setNews(articles)
            }
        }

        override fun onFailure(call: Call<NewsResponse>, t: Throwable) {
            // Handle failure
        }
    })
}
```



Menampilkan berita di desain yang sudah dibuat di xml

1. Membuat class NewsAdapter untuk menampilkan daftar artikel berita dalam recycle view

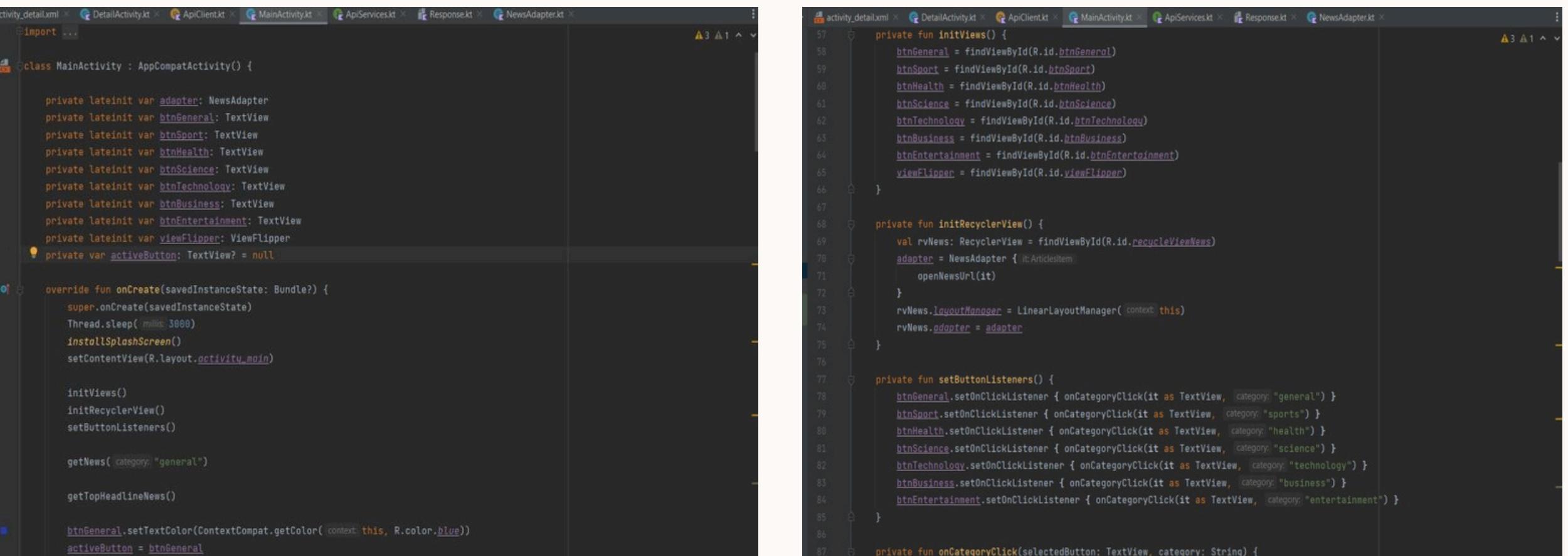
```
3 import ...
13
14 class NewsAdapter(private val listener: (ArticlesItem) -> Unit) :
15     RecyclerView.Adapter<NewsAdapter.ViewHolder>() {
16
17     private var news = listOf<ArticlesItem>()
18
19     fun setNews(news: List<ArticlesItem>) {
20         this.news = news
21         notifyDataSetChanged()
22     }
23
24    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
25        val view =
26            LayoutInflater.from(parent.context).inflate(R.layout.recycle1, parent, attachToRoot: false)
27        return ViewHolder(view)
28    }
29
30    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
31        holder.bind(news[position])
32    }
33
34    override fun getItemCount(): Int {
35        return news.size
36    }
37
38    inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
39
40        val tvTitle: TextView = itemView.findViewById(R.id.tvTitle)
41        val ImageNews: ImageView = itemView.findViewById(R.id.imageNews)
42        val tvSource: TextView = itemView.findViewById(R.id.tvSource)
43
44        fun bind(news: ArticlesItem) {
45            tvTitle.text = news.title
46            tvSource.text = news.author
47
48            Glide.with(itemView.context).load(news.urlToImage)
49                .apply(RequestOptions().dontTransform().placeholder(R.drawable.loading_icon))
50                .into(ImageNews)
51
52            itemView.setOnClickListener { it: View ->
53                listener(news)
54            }
55        }
56    }
57}
```

2. Membuat function di class main_activity untuk memperoleh id recycleView dari activity_main.xml dan disambungkan ke class newsAdapter

```
recyclerView() {
    recyclerView = findViewById(R.id.recycleView)
    sAdapter = ArticlesItem
    rl(it)

    Manager = LinearLayoutManager(context: t
    r = adapter
```

3. Membuat textView di horizontalScrollView berfungsi agar jika salah satu di klik maka list berita yang tampil di recycle view akan sesuai dengan category di link berita yang di apiClient



```

activity_detail.xml  DetailActivity.kt  ApiClient.kt  MainActivity.kt  ApiServices.kt  Response.kt  NewsAdapter.kt
3 import ...
25
26 class MainActivity : AppCompatActivity() {
27
28     private lateinit var adapter: NewsAdapter
29     private lateinit var btnGeneral: TextView
30     private lateinit var btnSport: TextView
31     private lateinit var btnHealth: TextView
32     private lateinit var btnScience: TextView
33     private lateinit var btnTechnology: TextView
34     private lateinit var btnBusiness: TextView
35     private lateinit var btnEntertainment: TextView
36     private lateinit var viewFlipper: ViewFlipper
37     private var activeButton: TextView? = null
38
39     override fun onCreate(savedInstanceState: Bundle?) {
40         super.onCreate(savedInstanceState)
41         Thread.sleep(millis 3000)
42         installSplashScreen()
43         setContentView(R.layout.activity_main)
44
45         initViews()
46         initRecyclerView()
47         setButtonListeners()
48
49         getNews(category "general")
50
51         getTopHeadlineNews()
52
53         btnGeneral.setTextColor(ContextCompat.getColor(context this, R.color.blue))
54         activeButton = btnGeneral

```

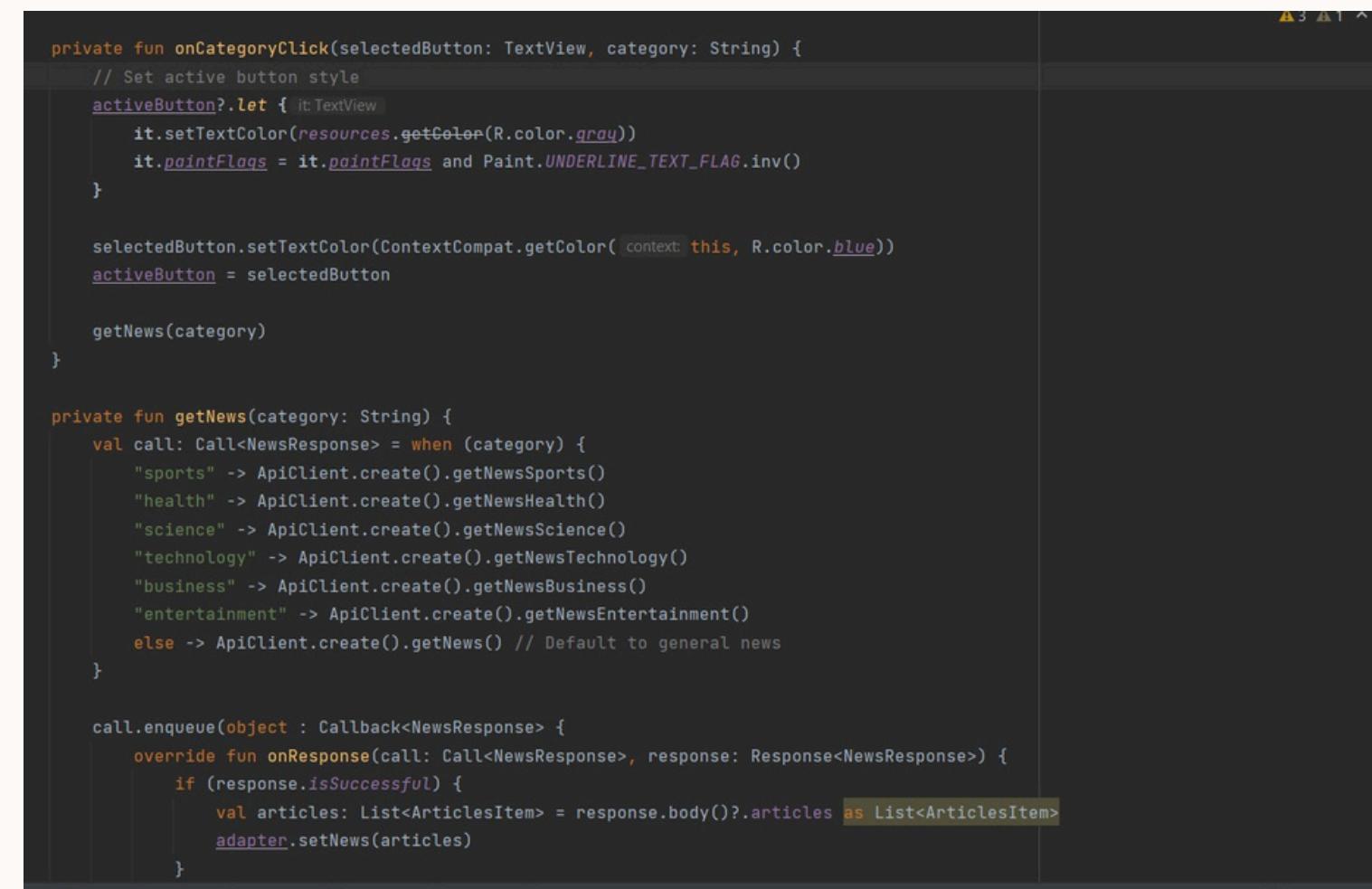


```

activity_detail.xml  DetailActivity.kt  ApiClient.kt  MainActivity.kt  ApiServices.kt  Response.kt  NewsAdapter.kt
57     private fun initViews() {
58         btnGeneral = findViewById(R.id.btnGeneral)
59         btnSport = findViewById(R.id.btnSport)
60         btnHealth = findViewById(R.id.btnHealth)
61         btnScience = findViewById(R.id.btnScience)
62         btnTechnology = findViewById(R.id.btnTechnology)
63         btnBusiness = findViewById(R.id.btnBusiness)
64         btnEntertainment = findViewById(R.id.btnEntertainment)
65         viewFlipper = findViewById(R.id.viewFlipper)
66     }
67
68     private fun initRecyclerView() {
69         val rvNews: RecyclerView = findViewById(R.id.recycleViewNews)
70         adapter = NewsAdapter { item: ArticleItem -
71             openNewsUrl(it)
72         }
73         rvNews.layoutManager = LinearLayoutManager(context this)
74         rvNews.adapter = adapter
75     }
76
77     private fun setButtonListeners() {
78         btnGeneral.setOnClickListener { onCategoryClick(it as TextView, category "general") }
79         btnSport.setOnClickListener { onCategoryClick(it as TextView, category "sports") }
80         btnHealth.setOnClickListener { onCategoryClick(it as TextView, category "health") }
81         btnScience.setOnClickListener { onCategoryClick(it as TextView, category "science") }
82         btnTechnology.setOnClickListener { onCategoryClick(it as TextView, category "technology") }
83         btnBusiness.setOnClickListener { onCategoryClick(it as TextView, category "business") }
84         btnEntertainment.setOnClickListener { onCategoryClick(it as TextView, category "entertainment") }
85     }
86
87     private fun onCategoryClick(selectedButton: TextView, category: String) {

```

4. Membuat textView di horizontalScrollView berfungsi agar jika salah satu di klik maka list berita yang tampil di recycle view akan sesuai dengan category di link berita yang di apiClient



```
private fun onCategoryClick(selectedButton: TextView, category: String) {
    // Set active button style
    activeButton?.let { it: TextView ->
        it.setTextColor(resources.getColor(R.color.gray))
        it.paintFlags = it.paintFlags and Paint.UNDERLINE_TEXT_FLAG.inv()
    }

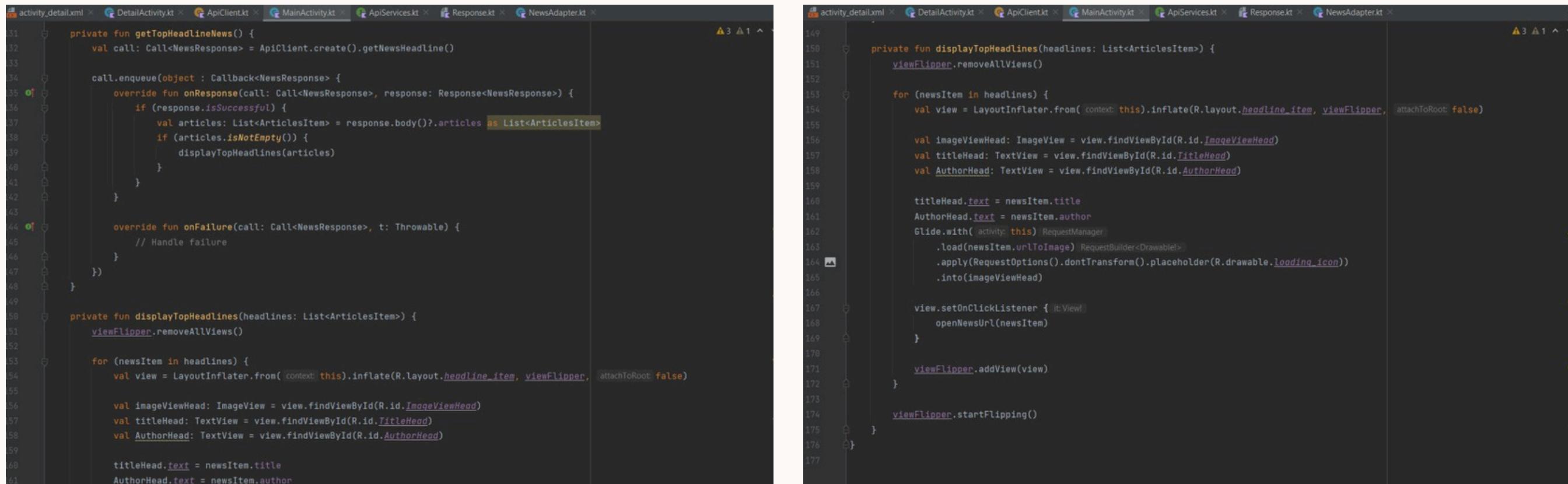
    selectedButton.setTextColor(ContextCompat.getColor(context, R.color.blue))
    activeButton = selectedButton

    getNews(category)
}

private fun getNews(category: String) {
    val call: Call<NewsResponse> = when (category) {
        "sports" -> ApiClient.create().getNewsSports()
        "health" -> ApiClient.create().getNewsHealth()
        "science" -> ApiClient.create().getNewsScience()
        "technology" -> ApiClient.create().getNewsTechnology()
        "business" -> ApiClient.create().getNewsBusiness()
        "entertainment" -> ApiClient.create().getNewsEntertainment()
        else -> ApiClient.create().getNews() // Default to general news
    }

    call.enqueue(object : Callback<NewsResponse> {
        override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {
            if (response.isSuccessful) {
                val articles: List<ArticlesItem> = response.body()?.articles as List<ArticlesItem>
                adapter.setNews(articles)
            }
        }
    })
}
```

5. Membuat topHeadline dengan ViewFlipper

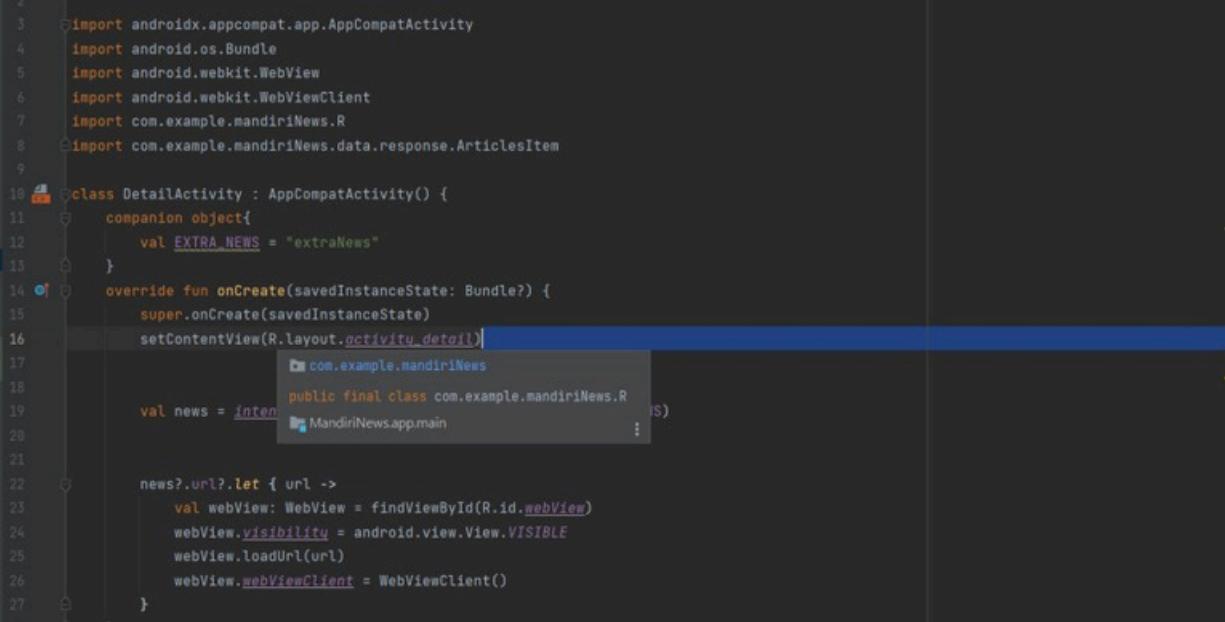


```

activity_detail.xml x DetailActivity.kt x ApiClient.kt x MainActivity.kt x ApiService.kt x Response.kt x NewsAdapter.kt x
149 private fun displayTopHeadlines(headlines: List<ArticlesItem>) {
150     viewFlipper.removeAllViews()
151
152     for (newsItem in headlines) {
153         val view = LayoutInflater.from(context).inflate(R.layout.headline_item, viewFlipper, attachToRoot: false)
154
155         val imageViewHead: ImageView = view.findViewById(R.id.ImageViewHead)
156         val titleHead: TextView = view.findViewById(R.id.TitleHead)
157         val AuthorHead: TextView = view.findViewById(R.id.AuthorHead)
158
159         titleHead.text = newsItem.title
160         AuthorHead.text = newsItem.author
161
162         Glide.with(activity).RequestManager
163             .load(newsItem.urlToImage).RequestBuilder<Drawable>
164             .apply(RequestOptions().dontTransform().placeholder(R.drawable.loading_icon))
165             .into(imageViewHead)
166
167         view.setOnClickListener { it: View? ->
168             openNewsUrl(newsItem)
169         }
170
171         viewFlipper.addView(view)
172     }
173
174     viewFlipper.startFlipping()
175 }
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
625
626
627
627
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
699
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
799
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
899
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
12
```

6. Membuat DetailActivity untuk menampilkan WebView dari berita yang di klik di card berita di mainActivity dengan membuat function openNewsUrl untuk menyambungkan ke DetailActivity

```
113     if (response.isSuccessful) {
114         val articles: List<ArticlesItem> = response.body()?.articles as List<ArticlesItem>
115         adapter.setNews(articles)
116     }
117 }
118
119 override fun onFailure(call: Call<NewsResponse>, t: Throwable) {
120     // Handle failure
121 }
122 })
123 }
124
125 private fun openNewsUrl(newsItem: ArticlesItem) {
126     val intent = Intent(packageContext: this, DetailActivity::class.java)
127     intent.putExtra(DetailActivity.EXTRA_NEWS, newsItem)
128     startActivity(intent)
129 }
130
131 private fun getTopHeadlineNews() {
132     val call: Call<NewsResponse> = ApiClient.create().getNewsHeadline()
133
134     call.enqueue(object : Callback<NewsResponse> {
135         override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {
136             if (response.isSuccessful) {
137                 val articles: List<ArticlesItem> = response.body()?.articles as List<ArticlesItem>
138                 if (articles.isNotEmpty()) {
139                     displayTopHeadlines(articles)
140                 }
141             }
142         }
143     })
144 }
```



The screenshot shows the Android Studio code editor with the file `DetailActivity.kt` open. The cursor is at the end of the line `setContentView(R.layout.activity_detail)`. A dropdown menu is open, showing suggestions for the layout file. The top suggestion is `R.layout.activity_detail`, which is highlighted with a blue border. Other suggestions include `com.example.mandiriNews`, `public final class com.example.mandiriNews.R`, and `MandiriNews.app.main`. The code editor interface includes tabs for other files like `activity_detail.xml`, `DetailActivity.kt`, `ApiClient.kt`, etc., at the top. The status bar at the bottom right shows "Android Studio | MandiriNews" and "File | Project Structure".

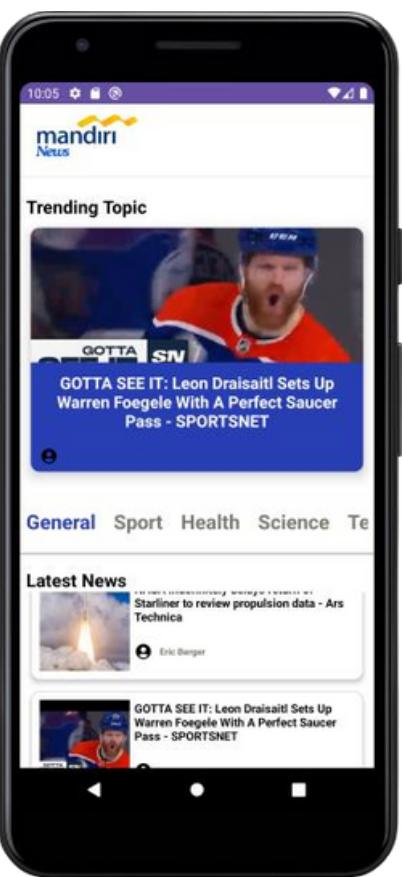
```
1 package com.example.mandiriNews.ui.detail
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.webkit.WebView
6 import android.webkit.WebViewClient
7 import com.example.mandiriNews.R
8 import com.example.mandiriNews.data.response.ArticlesItem
9
10 class DetailActivity : AppCompatActivity() {
11     companion object{
12         val EXTRA_NEWS = "extraNews"
13     }
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_detail)
17             // com.example.mandiriNews
18
19         val news = intent
20             // MandiriNews.app.main
21
22         news?.url?.let { url ->
23             val webView: WebView = findViewById(R.id.webView)
24             webView.visibility = android.view.View.VISIBLE
25             webView.loadUrl(url)
26             webView.webViewClient = WebViewClient()
27         }
28     }
29 }
```



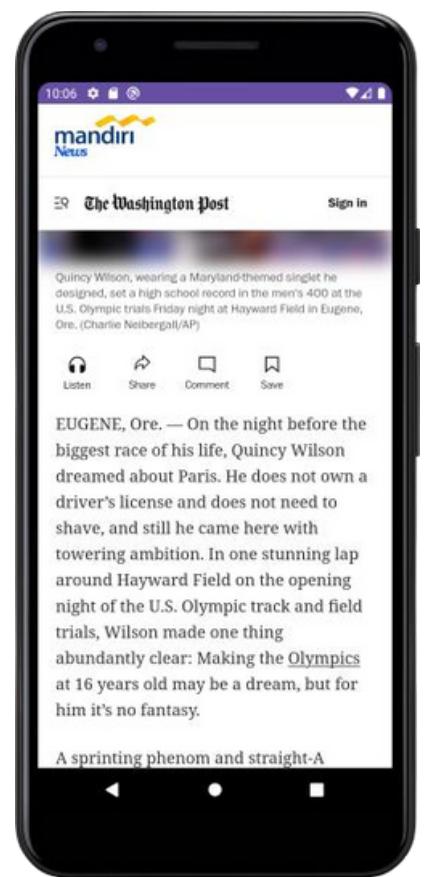
Tampilan Aplikasi



SplashScreen



Main Activity



Detail Activity



GITHUB

<https://github.com/Etsbagas/MandiriNews>

THANK YOU

