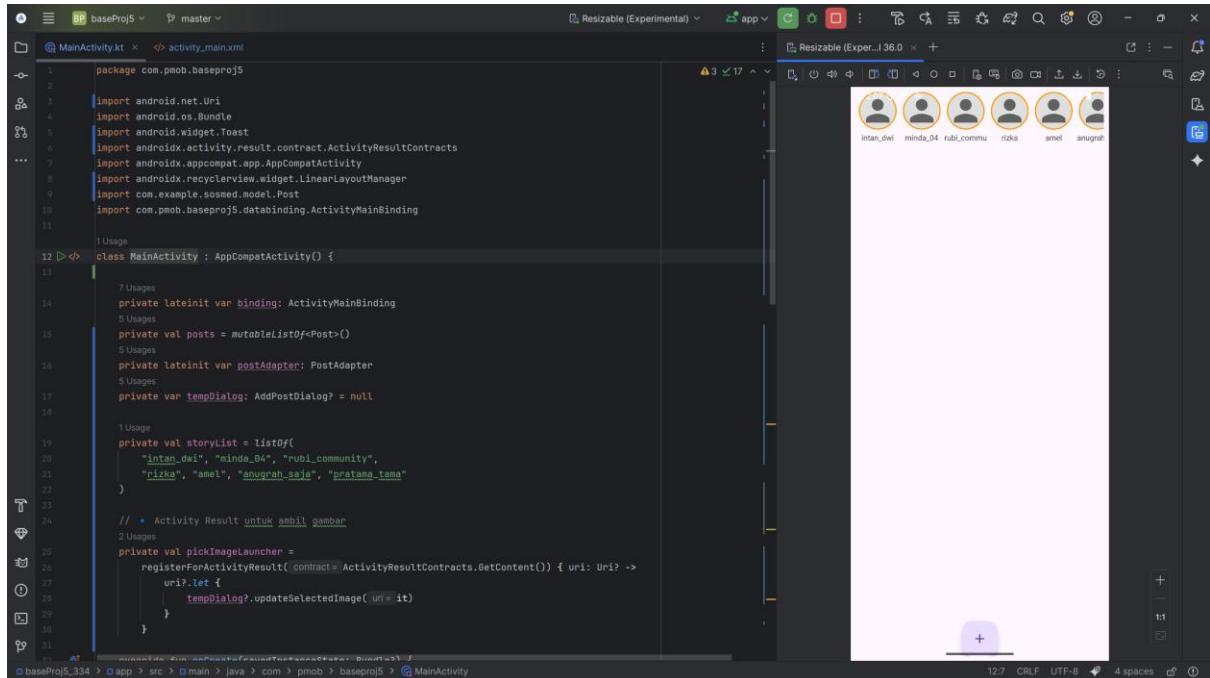


Nama : Muhammad Luthfindi Anugrah Pratama

NIM : 2300018334

Slot : Senin, 16:45

Scroll story



Screenshot of Android Studio showing the `MainActivity.kt` file. The code is as follows:

```
package com.pmob.baseproj5

import android.net.Uri
import android.os.Bundle
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.sosmed.model.Post
import com.pmob.baseproj5.databinding.ActivityMainBinding

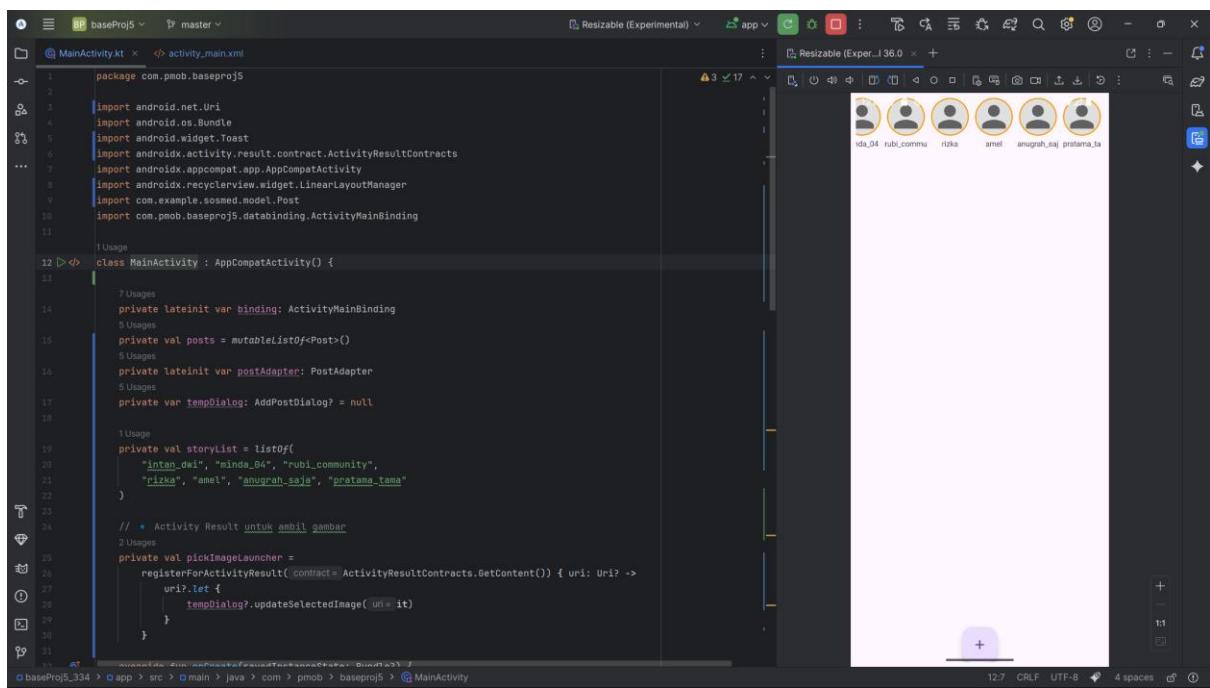
class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private val posts = mutableListOf<Post>()
    private lateinit var postAdapter: PostAdapter
    private var tempDialog: AddPostDialog? = null

    private val storyList = listOf(
        "intan_dwi", "minde_04", "rubi_community",
        "rizka", "amel", "anugrah_sala", "pratama_tama"
    )

    private val pickImageLauncher =
        registerForActivityResult(contract = ActivityResultContracts.GetContent()) { uri: Uri? ->
            uri?.let {
                tempDialog?.updateSelectedImage(uri = it)
            }
        }
}
```

The right side of the screen shows a `Resizable (Experimental) 36.0` window containing a list of user profiles with names: intan_dwi, minde_04, rubi_community, rizka, amel, and anugrah.



Screenshot of Android Studio showing the `MainActivity.kt` file. The code is identical to the previous screenshot:

```
package com.pmob.baseproj5

import android.net.Uri
import android.os.Bundle
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.sosmed.model.Post
import com.pmob.baseproj5.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

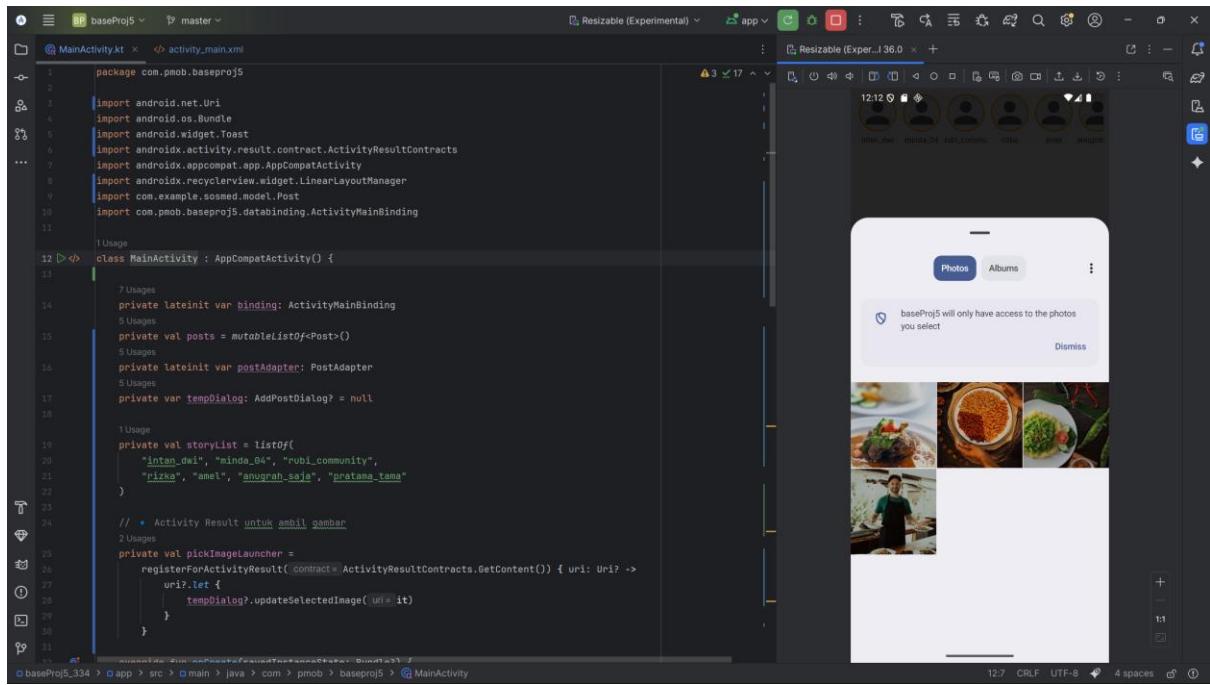
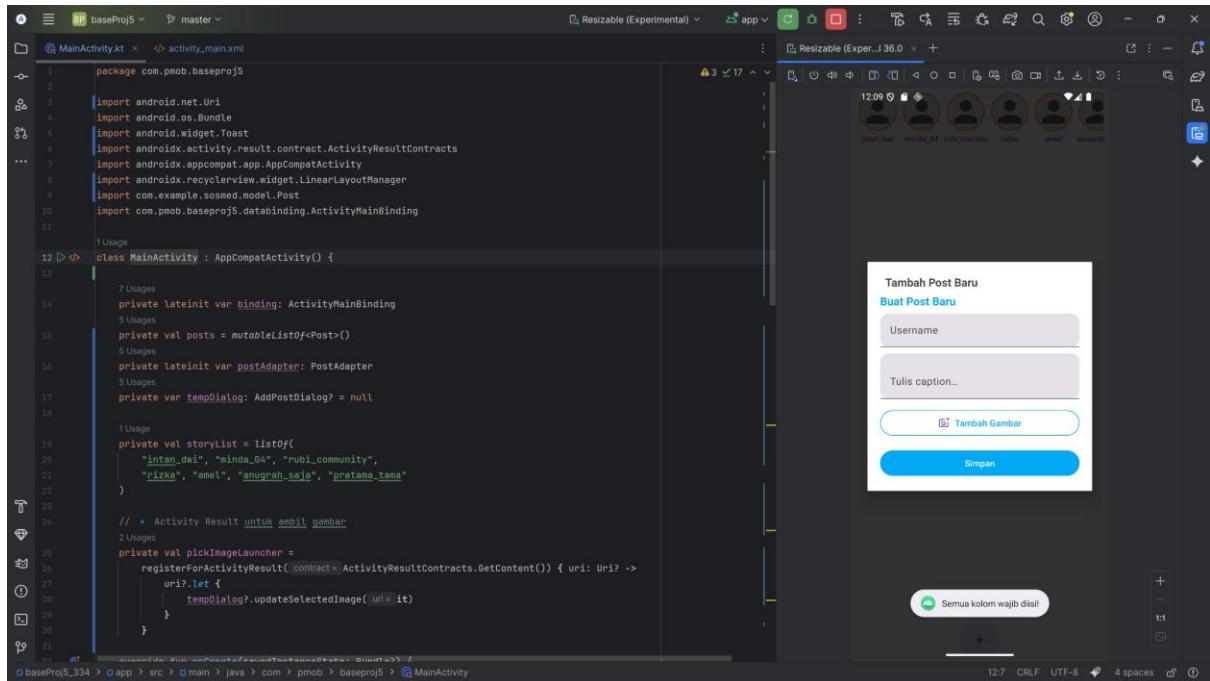
    private lateinit var binding: ActivityMainBinding
    private val posts = mutableListOf<Post>()
    private lateinit var postAdapter: PostAdapter
    private var tempDialog: AddPostDialog? = null

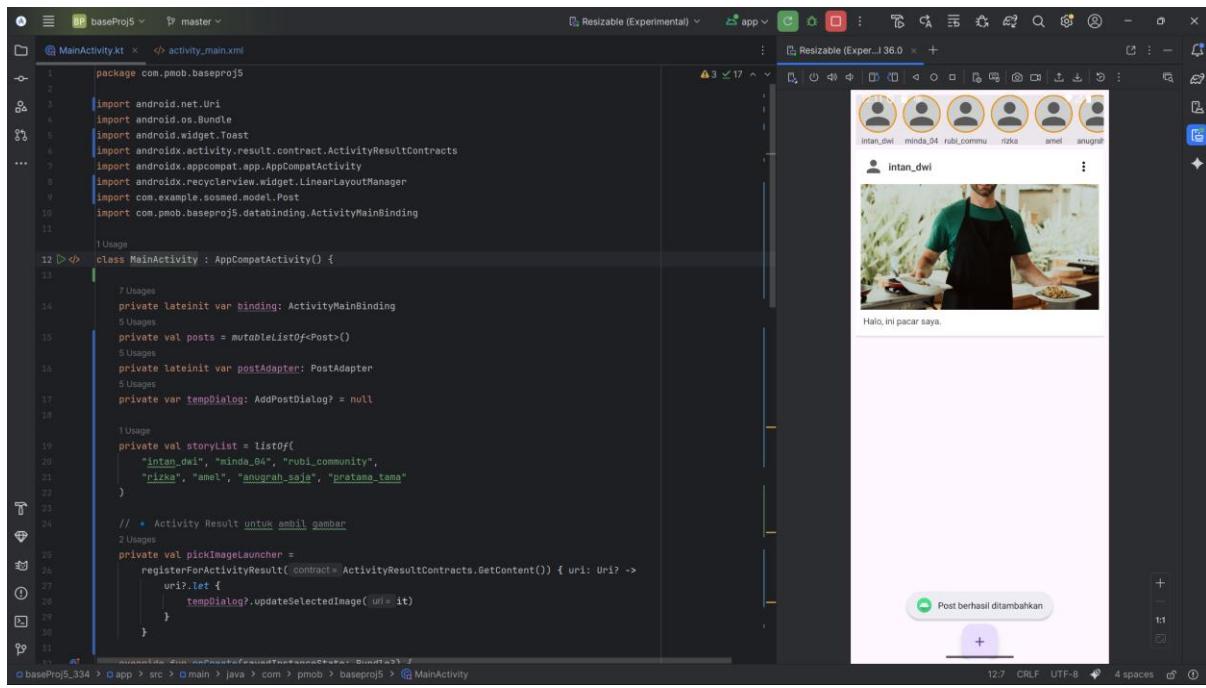
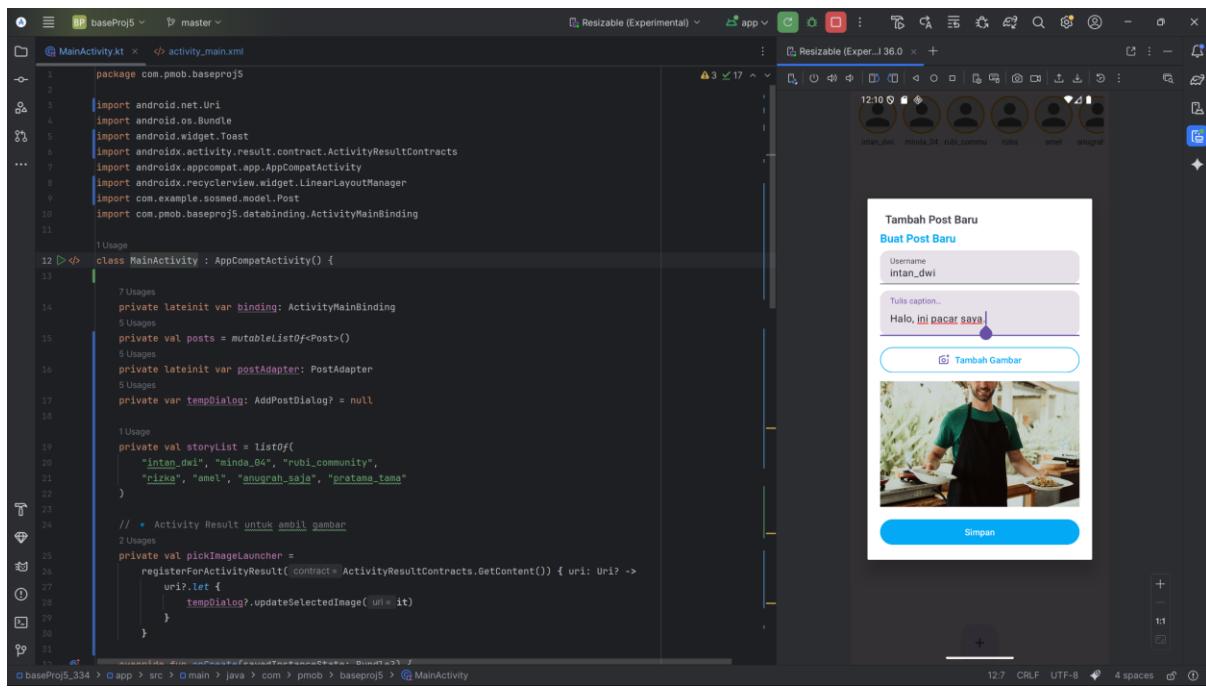
    private val storyList = listOf(
        "intan_dwi", "minde_04", "rubi_community",
        "rizka", "amel", "anugrah_sala", "pratama_tama"
    )

    private val pickImageLauncher =
        registerForActivityResult(contract = ActivityResultContracts.GetContent()) { uri: Uri? ->
            uri?.let {
                tempDialog?.updateSelectedImage(uri = it)
            }
        }
}
```

The right side of the screen shows a `Resizable (Experimental) 36.0` window containing a list of user profiles with names: vda_04, rubi_community, rizka, amel, anugrah_sala, and pratama_tama.

Tambah postingan





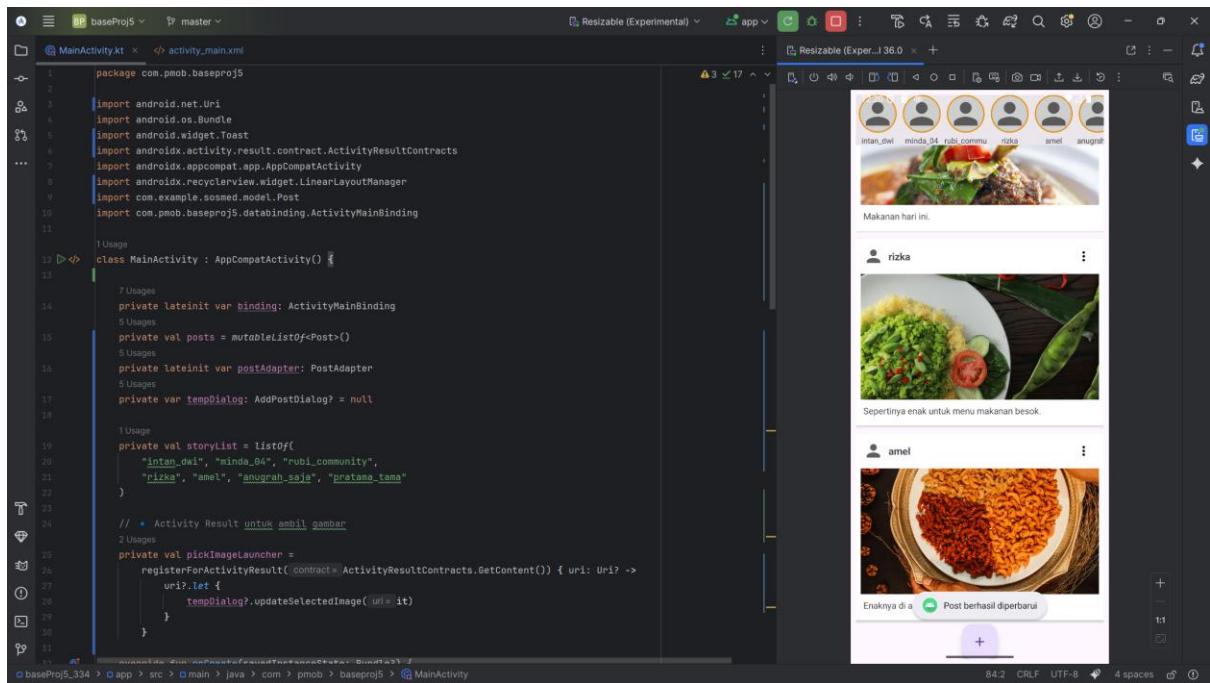
Screenshot of an Android application interface. On the left, the code editor shows the `MainActivity.kt` file with Java code. On the right, the app's main screen displays a feed of six posts. Each post consists of a user's profile picture, a small thumbnail image, and a caption. The users are intan_dwi, minda_04, rubi_commu, rizka, amel, and anugrah.

```
1 package com.pmob.baseproj5
2
3 import android.net.Uri
4 import android.os.Bundle
5 import android.widget.Toast
6 import androidx.activity.result.contract.ActivityResultContracts
7 import androidx.appcompat.app.AppCompatActivity
8 import androidx.recyclerview.widget.LinearLayoutManager
9 import com.example.sosmed.model.Post
10 import com.pmob.baseproj5.databinding.ActivityMainBinding
11
12 class MainActivity : AppCompatActivity() {
13
14     private lateinit var binding: ActivityMainBinding
15
16     private val posts = mutableListOf<Post>()
17
18     private lateinit var postAdapter: PostAdapter
19
20     private var tempDialog: AddPostDialog? = null
21
22     private val storyList = listOf(
23         "intan_dwi", "mindy_04", "rubi_community",
24         "rizka", "amel", "anugrah_sale", "pratama_tama"
25     )
26
27     // * Activity Result untuk ambil gambar
28     private val pickImageLauncher =
29         registerForActivityResult(contract = ActivityResultContracts.GetContent()) { uri: Uri? ->
30             uri?.let {
31                 tempDialog?.updateSelectedImage(uri = it)
32             }
33         }
34
35     override fun onCreate(savedInstanceState: Bundle?) {
36         super.onCreate(savedInstanceState)
37         binding = ActivityMainBinding.inflate(layoutInflater)
38         setContentView(binding.root)
39
40         binding.recyclerView.layoutManager = LinearLayoutManager(this)
41         postAdapter = PostAdapter(posts)
42         binding.recyclerView.adapter = postAdapter
43
44         binding.fab.setOnClickListener {
45             tempDialog = AddPostDialog()
46             tempDialog?.show(supportFragmentManager, "temp")
47         }
48     }
49
50     private fun addPost(post: Post) {
51         posts.add(post)
52         postAdapter.notifyDataSetChanged()
53     }
54
55     private fun removePost(post: Post) {
56         posts.remove(post)
57         postAdapter.notifyDataSetChanged()
58     }
59
60     private fun updatePost(post: Post) {
61         posts[posts.indexOf(post)] = post
62         postAdapter.notifyDataSetChanged()
63     }
64
65     private fun checkPost(post: Post) {
66         if (post.isEdited) {
67             updatePost(post)
68         } else {
69             removePost(post)
70         }
71     }
72
73     private fun checkPost(post: Post) {
74         if (post.isEdited) {
75             updatePost(post)
76         } else {
77             removePost(post)
78         }
79     }
80
81     private fun checkPost(post: Post) {
82         if (post.isEdited) {
83             updatePost(post)
84         } else {
85             removePost(post)
86         }
87     }
88
89     private fun checkPost(post: Post) {
90         if (post.isEdited) {
91             updatePost(post)
92         } else {
93             removePost(post)
94         }
95     }
96
97     private fun checkPost(post: Post) {
98         if (post.isEdited) {
99             updatePost(post)
100        } else {
101            removePost(post)
102        }
103    }
104
105    private fun checkPost(post: Post) {
106        if (post.isEdited) {
107            updatePost(post)
108        } else {
109            removePost(post)
110        }
111    }
112
113    private fun checkPost(post: Post) {
114        if (post.isEdited) {
115            updatePost(post)
116        } else {
117            removePost(post)
118        }
119    }
120
121    private fun checkPost(post: Post) {
122        if (post.isEdited) {
123            updatePost(post)
124        } else {
125            removePost(post)
126        }
127    }
128
129    private fun checkPost(post: Post) {
130        if (post.isEdited) {
131            updatePost(post)
132        } else {
133            removePost(post)
134        }
135    }
136
137    private fun checkPost(post: Post) {
138        if (post.isEdited) {
139            updatePost(post)
140        } else {
141            removePost(post)
142        }
143    }
144
145    private fun checkPost(post: Post) {
146        if (post.isEdited) {
147            updatePost(post)
148        } else {
149            removePost(post)
150        }
151    }
152
153    private fun checkPost(post: Post) {
154        if (post.isEdited) {
155            updatePost(post)
156        } else {
157            removePost(post)
158        }
159    }
160
161    private fun checkPost(post: Post) {
162        if (post.isEdited) {
163            updatePost(post)
164        } else {
165            removePost(post)
166        }
167    }
168
169    private fun checkPost(post: Post) {
170        if (post.isEdited) {
171            updatePost(post)
172        } else {
173            removePost(post)
174        }
175    }
176
177    private fun checkPost(post: Post) {
178        if (post.isEdited) {
179            updatePost(post)
180        } else {
181            removePost(post)
182        }
183    }
184
185    private fun checkPost(post: Post) {
186        if (post.isEdited) {
187            updatePost(post)
188        } else {
189            removePost(post)
190        }
191    }
192
193    private fun checkPost(post: Post) {
194        if (post.isEdited) {
195            updatePost(post)
196        } else {
197            removePost(post)
198        }
199    }
200
201    private fun checkPost(post: Post) {
202        if (post.isEdited) {
203            updatePost(post)
204        } else {
205            removePost(post)
206        }
207    }
208
209    private fun checkPost(post: Post) {
210        if (post.isEdited) {
211            updatePost(post)
212        } else {
213            removePost(post)
214        }
215    }
216
217    private fun checkPost(post: Post) {
218        if (post.isEdited) {
219            updatePost(post)
220        } else {
221            removePost(post)
222        }
223    }
224
225    private fun checkPost(post: Post) {
226        if (post.isEdited) {
227            updatePost(post)
228        } else {
229            removePost(post)
230        }
231    }
232
233    private fun checkPost(post: Post) {
234        if (post.isEdited) {
235            updatePost(post)
236        } else {
237            removePost(post)
238        }
239    }
240
241    private fun checkPost(post: Post) {
242        if (post.isEdited) {
243            updatePost(post)
244        } else {
245            removePost(post)
246        }
247    }
248
249    private fun checkPost(post: Post) {
250        if (post.isEdited) {
251            updatePost(post)
252        } else {
253            removePost(post)
254        }
255    }
256
257    private fun checkPost(post: Post) {
258        if (post.isEdited) {
259            updatePost(post)
260        } else {
261            removePost(post)
262        }
263    }
264
265    private fun checkPost(post: Post) {
266        if (post.isEdited) {
267            updatePost(post)
268        } else {
269            removePost(post)
270        }
271    }
272
273    private fun checkPost(post: Post) {
274        if (post.isEdited) {
275            updatePost(post)
276        } else {
277            removePost(post)
278        }
279    }
280
281    private fun checkPost(post: Post) {
282        if (post.isEdited) {
283            updatePost(post)
284        } else {
285            removePost(post)
286        }
287    }
288
289    private fun checkPost(post: Post) {
290        if (post.isEdited) {
291            updatePost(post)
292        } else {
293            removePost(post)
294        }
295    }
296
297    private fun checkPost(post: Post) {
298        if (post.isEdited) {
299            updatePost(post)
300        } else {
301            removePost(post)
302        }
303    }
304
305    private fun checkPost(post: Post) {
306        if (post.isEdited) {
307            updatePost(post)
308        } else {
309            removePost(post)
310        }
311    }
312
313    private fun checkPost(post: Post) {
314        if (post.isEdited) {
315            updatePost(post)
316        } else {
317            removePost(post)
318        }
319    }
320
321    private fun checkPost(post: Post) {
322        if (post.isEdited) {
323            updatePost(post)
324        } else {
325            removePost(post)
326        }
327    }
328
329    private fun checkPost(post: Post) {
330        if (post.isEdited) {
331            updatePost(post)
332        } else {
333            removePost(post)
334        }
335    }
336
337    private fun checkPost(post: Post) {
338        if (post.isEdited) {
339            updatePost(post)
340        } else {
341            removePost(post)
342        }
343    }
344
345    private fun checkPost(post: Post) {
346        if (post.isEdited) {
347            updatePost(post)
348        } else {
349            removePost(post)
350        }
351    }
352
353    private fun checkPost(post: Post) {
354        if (post.isEdited) {
355            updatePost(post)
356        } else {
357            removePost(post)
358        }
359    }
360
361    private fun checkPost(post: Post) {
362        if (post.isEdited) {
363            updatePost(post)
364        } else {
365            removePost(post)
366        }
367    }
368
369    private fun checkPost(post: Post) {
370        if (post.isEdited) {
371            updatePost(post)
372        } else {
373            removePost(post)
374        }
375    }
376
377    private fun checkPost(post: Post) {
378        if (post.isEdited) {
379            updatePost(post)
380        } else {
381            removePost(post)
382        }
383    }
384
385    private fun checkPost(post: Post) {
386        if (post.isEdited) {
387            updatePost(post)
388        } else {
389            removePost(post)
390        }
391    }
392
393    private fun checkPost(post: Post) {
394        if (post.isEdited) {
395            updatePost(post)
396        } else {
397            removePost(post)
398        }
399    }
400
401    private fun checkPost(post: Post) {
402        if (post.isEdited) {
403            updatePost(post)
404        } else {
405            removePost(post)
406        }
407    }
408
409    private fun checkPost(post: Post) {
410        if (post.isEdited) {
411            updatePost(post)
412        } else {
413            removePost(post)
414        }
415    }
416
417    private fun checkPost(post: Post) {
418        if (post.isEdited) {
419            updatePost(post)
420        } else {
421            removePost(post)
422        }
423    }
424
425    private fun checkPost(post: Post) {
426        if (post.isEdited) {
427            updatePost(post)
428        } else {
429            removePost(post)
430        }
431    }
432
433    private fun checkPost(post: Post) {
434        if (post.isEdited) {
435            updatePost(post)
436        } else {
437            removePost(post)
438        }
439    }
440
441    private fun checkPost(post: Post) {
442        if (post.isEdited) {
443            updatePost(post)
444        } else {
445            removePost(post)
446        }
447    }
448
449    private fun checkPost(post: Post) {
450        if (post.isEdited) {
451            updatePost(post)
452        } else {
453            removePost(post)
454        }
455    }
456
457    private fun checkPost(post: Post) {
458        if (post.isEdited) {
459            updatePost(post)
460        } else {
461            removePost(post)
462        }
463    }
464
465    private fun checkPost(post: Post) {
466        if (post.isEdited) {
467            updatePost(post)
468        } else {
469            removePost(post)
470        }
471    }
472
473    private fun checkPost(post: Post) {
474        if (post.isEdited) {
475            updatePost(post)
476        } else {
477            removePost(post)
478        }
479    }
480
481    private fun checkPost(post: Post) {
482        if (post.isEdited) {
483            updatePost(post)
484        } else {
485            removePost(post)
486        }
487    }
488
489    private fun checkPost(post: Post) {
490        if (post.isEdited) {
491            updatePost(post)
492        } else {
493            removePost(post)
494        }
495    }
496
497    private fun checkPost(post: Post) {
498        if (post.isEdited) {
499            updatePost(post)
500        } else {
501            removePost(post)
502        }
503    }
504
505    private fun checkPost(post: Post) {
506        if (post.isEdited) {
507            updatePost(post)
508        } else {
509            removePost(post)
510        }
511    }
512
513    private fun checkPost(post: Post) {
514        if (post.isEdited) {
515            updatePost(post)
516        } else {
517            removePost(post)
518        }
519    }
520
521    private fun checkPost(post: Post) {
522        if (post.isEdited) {
523            updatePost(post)
524        } else {
525            removePost(post)
526        }
527    }
528
529    private fun checkPost(post: Post) {
530        if (post.isEdited) {
531            updatePost(post)
532        } else {
533            removePost(post)
534        }
535    }
536
537    private fun checkPost(post: Post) {
538        if (post.isEdited) {
539            updatePost(post)
540        } else {
541            removePost(post)
542        }
543    }
544
545    private fun checkPost(post: Post) {
546        if (post.isEdited) {
547            updatePost(post)
548        } else {
549            removePost(post)
550        }
551    }
552
553    private fun checkPost(post: Post) {
554        if (post.isEdited) {
555            updatePost(post)
556        } else {
557            removePost(post)
558        }
559    }
560
561    private fun checkPost(post: Post) {
562        if (post.isEdited) {
563            updatePost(post)
564        } else {
565            removePost(post)
566        }
567    }
568
569    private fun checkPost(post: Post) {
570        if (post.isEdited) {
571            updatePost(post)
572        } else {
573            removePost(post)
574        }
575    }
576
577    private fun checkPost(post: Post) {
578        if (post.isEdited) {
579            updatePost(post)
580        } else {
581            removePost(post)
582        }
583    }
584
585    private fun checkPost(post: Post) {
586        if (post.isEdited) {
587            updatePost(post)
588        } else {
589            removePost(post)
590        }
591    }
592
593    private fun checkPost(post: Post) {
594        if (post.isEdited) {
595            updatePost(post)
596        } else {
597            removePost(post)
598        }
599    }
599
600    private fun checkPost(post: Post) {
601        if (post.isEdited) {
602            updatePost(post)
603        } else {
604            removePost(post)
605        }
606    }
606
607    private fun checkPost(post: Post) {
608        if (post.isEdited) {
609            updatePost(post)
610        } else {
611            removePost(post)
612        }
613    }
613
614    private fun checkPost(post: Post) {
615        if (post.isEdited) {
616            updatePost(post)
617        } else {
618            removePost(post)
619        }
620    }
620
621    private fun checkPost(post: Post) {
622        if (post.isEdited) {
623            updatePost(post)
624        } else {
625            removePost(post)
626        }
627    }
627
628    private fun checkPost(post: Post) {
629        if (post.isEdited) {
630            updatePost(post)
631        } else {
632            removePost(post)
633        }
634    }
634
635    private fun checkPost(post: Post) {
636        if (post.isEdited) {
637            updatePost(post)
638        } else {
639            removePost(post)
640        }
641    }
641
642    private fun checkPost(post: Post) {
643        if (post.isEdited) {
644            updatePost(post)
645        } else {
646            removePost(post)
647        }
648    }
648
649    private fun checkPost(post: Post) {
650        if (post.isEdited) {
651            updatePost(post)
652        } else {
653            removePost(post)
654        }
655    }
655
656    private fun checkPost(post: Post) {
657        if (post.isEdited) {
658            updatePost(post)
659        } else {
660            removePost(post)
661        }
662    }
662
663    private fun checkPost(post: Post) {
664        if (post.isEdited) {
665            updatePost(post)
666        } else {
667            removePost(post)
668        }
669    }
669
670    private fun checkPost(post: Post) {
671        if (post.isEdited) {
672            updatePost(post)
673        } else {
674            removePost(post)
675        }
676    }
676
677    private fun checkPost(post: Post) {
678        if (post.isEdited) {
679            updatePost(post)
680        } else {
681            removePost(post)
682        }
683    }
683
684    private fun checkPost(post: Post) {
685        if (post.isEdited) {
686            updatePost(post)
687        } else {
688            removePost(post)
689        }
690    }
690
691    private fun checkPost(post: Post) {
692        if (post.isEdited) {
693            updatePost(post)
694        } else {
695            removePost(post)
696        }
697    }
697
698    private fun checkPost(post: Post) {
699        if (post.isEdited) {
700            updatePost(post)
701        } else {
702            removePost(post)
703        }
704    }
704
705    private fun checkPost(post: Post) {
706        if (post.isEdited) {
707            updatePost(post)
708        } else {
709            removePost(post)
710        }
711    }
711
712    private fun checkPost(post: Post) {
713        if (post.isEdited) {
714            updatePost(post)
715        } else {
716            removePost(post)
717        }
718    }
718
719    private fun checkPost(post: Post) {
720        if (post.isEdited) {
721            updatePost(post)
722        } else {
723            removePost(post)
724        }
725    }
725
726    private fun checkPost(post: Post) {
727        if (post.isEdited) {
728            updatePost(post)
729        } else {
730            removePost(post)
731        }
732    }
732
733    private fun checkPost(post: Post) {
734        if (post.isEdited) {
735            updatePost(post)
736        } else {
737            removePost(post)
738        }
739    }
739
740    private fun checkPost(post: Post) {
741        if (post.isEdited) {
742            updatePost(post)
743        } else {
744            removePost(post)
745        }
746    }
746
747    private fun checkPost(post: Post) {
748        if (post.isEdited) {
749            updatePost(post)
750        } else {
751            removePost(post)
752        }
753    }
753
754    private fun checkPost(post: Post) {
755        if (post.isEdited) {
756            updatePost(post)
757        } else {
758            removePost(post)
759        }
760    }
760
761    private fun checkPost(post: Post) {
762        if (post.isEdited) {
763            updatePost(post)
764        } else {
765            removePost(post)
766        }
767    }
767
768    private fun checkPost(post: Post) {
769        if (post.isEdited) {
770            updatePost(post)
771        } else {
772            removePost(post)
773        }
774    }
774
775    private fun checkPost(post: Post) {
776        if (post.isEdited) {
777            updatePost(post)
778        } else {
779            removePost(post)
780        }
781    }
781
782    private fun checkPost(post: Post) {
783        if (post.isEdited) {
784            updatePost(post)
785        } else {
786            removePost(post)
787        }
788    }
788
789    private fun checkPost(post: Post) {
790        if (post.isEdited) {
791            updatePost(post)
792        } else {
793            removePost(post)
794        }
795    }
795
796    private fun checkPost(post: Post) {
797        if (post.isEdited) {
798            updatePost(post)
799        } else {
800            removePost(post)
801        }
802    }
802
803    private fun checkPost(post: Post) {
804        if (post.isEdited) {
805            updatePost(post)
806        } else {
807            removePost(post)
808        }
809    }
809
810    private fun checkPost(post: Post) {
811        if (post.isEdited) {
812            updatePost(post)
813        } else {
814            removePost(post)
815        }
816    }
816
817    private fun checkPost(post: Post) {
818        if (post.isEdited) {
819            updatePost(post)
820        } else {
821            removePost(post)
822        }
823    }
823
824    private fun checkPost(post: Post) {
825        if (post.isEdited) {
826            updatePost(post)
827        } else {
828            removePost(post)
829        }
830    }
830
831    private fun checkPost(post: Post) {
832        if (post.isEdited) {
833            updatePost(post)
834        } else {
835            removePost(post)
836        }
837    }
837
838    private fun checkPost(post: Post) {
839        if (post.isEdited) {
840            updatePost(post)
841        } else {
842            removePost(post)
843        }
844    }
844
845    private fun checkPost(post: Post) {
846        if (post.isEdited) {
847            updatePost(post)
848        } else {
849            removePost(post)
850        }
851    }
851
852    private fun checkPost(post: Post) {
853        if (post.isEdited) {
854            updatePost(post)
855        } else {
856            removePost(post)
857        }
858    }
858
859    private fun checkPost(post: Post) {
860        if (post.isEdited) {
861            updatePost(post)
862        } else {
863            removePost(post)
864        }
865    }
865
866    private fun checkPost(post: Post) {
867        if (post.isEdited) {
868            updatePost(post)
869        } else {
870            removePost(post)
871        }
872    }
872
873    private fun checkPost(post: Post) {
874        if (post.isEdited) {
875            updatePost(post)
876        } else {
877            removePost(post)
878        }
879    }
879
880    private fun checkPost(post: Post) {
881        if (post.isEdited) {
882            updatePost(post)
883        } else {
884            removePost(post)
885        }
886    }
886
887    private fun checkPost(post: Post) {
888        if (post.isEdited) {
889            updatePost(post)
890        } else {
891            removePost(post)
892        }
893    }
893
894    private fun checkPost(post: Post) {
895        if (post.isEdited) {
896            updatePost(post)
897        } else {
898            removePost(post)
899        }
900    }
900
901    private fun checkPost(post: Post) {
902        if (post.isEdited) {
903            updatePost(post)
904        } else {
905            removePost(post)
906        }
907    }
907
908    private fun checkPost(post: Post) {
909        if (post.isEdited) {
910            updatePost(post)
911        } else {
912            removePost(post)
913        }
914    }
914
915    private fun checkPost(post: Post) {
916        if (post.isEdited) {
917            updatePost(post)
918        } else {
919            removePost(post)
920        }
921    }
921
922    private fun checkPost(post: Post) {
923        if (post.isEdited) {
924            updatePost(post)
925        } else {
926            removePost(post)
927        }
928    }
928
929    private fun checkPost(post: Post) {
930        if (post.isEdited) {
931            updatePost(post)
932        } else {
933            removePost(post)
934        }
935    }
935
936    private fun checkPost(post: Post) {
937        if (post.isEdited) {
938            updatePost(post)
939        } else {
940            removePost(post)
941        }
942    }
942
943    private fun checkPost(post: Post) {
944        if (post.isEdited) {
945            updatePost(post)
946        } else {
947            removePost(post)
948        }
949    }
949
950    private fun checkPost(post: Post) {
951        if (post.isEdited) {
952            updatePost(post)
953        } else {
954            removePost(post)
955        }
956    }
956
957    private fun checkPost(post: Post) {
958        if (post.isEdited) {
959            updatePost(post)
960        } else {
961            removePost(post)
962        }
963    }
963
964    private fun checkPost(post: Post) {
965        if (post.isEdited) {
966            updatePost(post)
967        } else {
968            removePost(post)
969        }
970    }
970
971    private fun checkPost(post: Post) {
972        if (post.isEdited) {
973            updatePost(post)
974        } else {
975            removePost(post)
976        }
977    }
977
978    private fun checkPost(post: Post) {
979        if (post.isEdited) {
980            updatePost(post)
981        } else {
982            removePost(post)
983        }
984    }
984
985    private fun checkPost(post: Post) {
986        if (post.isEdited) {
987            updatePost(post)
988        } else {
989            removePost(post)
990        }
991    }
991
992    private fun checkPost(post: Post) {
993        if (post.isEdited) {
994            updatePost(post)
995        } else {
996            removePost(post)
997        }
998    }
998
999    private fun checkPost(post: Post) {
1000       if (post.isEdited) {
1001           updatePost(post)
1002       } else {
1003           removePost(post)
1004       }
1005   }
```

Screenshot of an Android application interface. On the left, the code editor shows the `MainActivity.kt` file with Java code. On the right, the app's main screen displays a feed of two posts. Each post consists of a user's profile picture, a small thumbnail image, and a caption. The users are rizka and amel.

```
1 package com.pmob.baseproj5
2
3 import android.net.Uri
4 import android.os.Bundle
5 import android.widget.Toast
6 import androidx.activity.result.contract.ActivityResultContracts
7 import androidx.appcompat.app.AppCompatActivity
8 import androidx.recyclerview.widget.LinearLayoutManager
9 import com.example.sosmed.model.Post
10 import com.pmob.baseproj5.databinding.ActivityMainBinding
11
12 class MainActivity : AppCompatActivity() {
13
14     private lateinit var binding: ActivityMainBinding
15
16     private val posts = mutableListOf<Post>()
17
18     private lateinit var postAdapter: PostAdapter
19
20     private var tempDialog: AddPostDialog? = null
21
22     private val storyList = listOf(
23         "intan_dwi", "mindy_04", "rubi_community",
24         "rizka", "amel", "anugrah_sale", "pratama_tama"
25     )
26
27     // * Activity Result untuk ambil gambar
28     private val pickImageLauncher =
29         registerForActivityResult(contract = ActivityResultContracts.GetContent()) { uri: Uri? ->
30             uri?.let {
31                 tempDialog?.updateSelectedImage(uri = it)
32             }
33         }
34
35     override fun onCreate(savedInstanceState: Bundle?) {
36         super.onCreate(savedInstanceState)
37         binding = ActivityMainBinding.inflate(layoutInflater)
38         setContentView(binding.root)
39
40         binding.recyclerView.layoutManager = LinearLayoutManager(this)
41         postAdapter = PostAdapter(posts)
42         binding.recyclerView.adapter = postAdapter
43
44         binding.fab.setOnClickListener {
45             tempDialog = AddPostDialog()
46             tempDialog?.show(supportFragmentManager, "temp")
47         }
48     }
49
50     private fun addPost(post: Post) {
51         posts.add(post)
52         postAdapter.notifyDataSetChanged()
53     }
54
55     private fun removePost(post: Post) {
56         posts.remove(post)
57         postAdapter.notifyDataSetChanged()
58     }
59
60     private fun updatePost(post: Post) {
61         posts[posts.indexOf(post)] = post
62         postAdapter.notifyDataSetChanged()
63     }
64
65     private fun checkPost(post: Post) {
66         if (post.isEdited) {
67             updatePost(post)
68         } else {
69             removePost(post)
70         }
71     }
72
73     private fun checkPost(post: Post) {
74         if (post.isEdited) {
75             updatePost(post)
76         } else {
77             removePost(post)
78         }
79     }
80
81     private fun checkPost(post: Post) {
82         if (post.isEdited) {
83             updatePost(post)
84         } else {
85             removePost(post)
86         }
87     }
88
89     private fun checkPost(post: Post) {
90         if (post.isEdited) {
91             updatePost(post)
92         } else {
93             removePost(post)
94         }
95     }
96
97     private fun checkPost(post: Post) {
98         if (post.isEdited) {
99             updatePost(post)
100        } else {
101            removePost(post)
102        }
103    }
104
105    private fun checkPost(post: Post) {
106        if (post.isEdited) {
107            updatePost(post)
108        } else {
109            removePost(post)
110       }
111    }
112
113    private fun checkPost(post: Post) {
114        if (post.isEdited) {
115            updatePost(post)
116        } else {
117            removePost(post)
118       }
119    }
120
121    private fun checkPost(post: Post) {
122        if (post.isEdited) {
123            updatePost(post)
124        } else {
125            removePost(post)
126       }
127    }
128
129    private fun checkPost(post: Post) {
130        if (post.isEdited) {
131            updatePost(post)
132        } else {
133            removePost(post)
134       }
135    }
136
137    private fun checkPost(post: Post) {
138        if (post.isEdited) {
139            updatePost(post)
140        } else {
141            removePost(post)
142       }
143    }
144
145    private fun checkPost(post: Post) {
146        if (post.isEdited) {
147            updatePost(post)
148        } else {
149            removePost(post)
150       }
151    }
152
153    private fun checkPost(post: Post) {
154        if (post.isEdited) {
155            updatePost(post)
156        } else {
157            removePost(post)
158       }
159    }
160
161    private fun checkPost(post: Post) {
162        if (post.isEdited) {
163            updatePost(post)
164        } else {
165            removePost(post)
166       }
167    }
168
169    private fun checkPost(post: Post) {
170        if (post.isEdited) {
171            updatePost(post)
172        } else {
173            removePost(post)
174       }
175    }
176
177    private fun checkPost(post: Post) {
178        if (post.isEdited) {
179            updatePost(post)
180        } else {
181            removePost(post)
182       }
183    }
184
185    private fun checkPost(post: Post) {
186        if (post.isEdited) {
187            updatePost(post)
188        } else {
189            removePost(post)
190       }
191    }
192
193    private fun checkPost(post: Post) {
194        if (post.isEdited) {
195            updatePost(post)
196        } else {
197            removePost(post)
198       }
199    }
199
200    private fun checkPost(post: Post) {
201        if (post.isEdited) {
202            updatePost(post)
203        } else {
204            removePost(post)
205       }
206    }
206
207    private fun checkPost(post: Post) {
208        if (post.isEdited) {
209            updatePost(post)
210        } else {
211            removePost(post)
212       }
213    }
213
214    private fun checkPost(post: Post) {
215        if (post.isEdited) {
216            updatePost(post)
217        } else {
218            removePost(post)
219       }
220    }
220
221    private fun checkPost(post: Post) {
222        if (post.isEdited) {
223            updatePost(post)
224        } else {
225            removePost(post)
226       }
227    }
227
228    private fun checkPost(post: Post) {
229        if (post.isEdited) {
230            updatePost(post)
231        } else {
232            removePost(post)
233       }
234    }
234
235    private fun checkPost(post: Post) {
236        if (post.isEdited) {
237            updatePost(post)
238        } else {
239            removePost(post)
240       }
241    }
241
242    private fun checkPost(post: Post) {
243        if (post.isEdited) {
244            updatePost(post)
245        } else {
246            removePost(post)
247       }
248    }
248
249    private fun checkPost(post: Post) {
250        if (post.isEdited) {
251            updatePost(post)
252        } else {
253            removePost(post)
254       }
255    }
255
256    private fun checkPost(post: Post) {
257        if (post.isEdited) {
258            updatePost(post)
259        } else {
260            removePost(post)
261       }
262    }
262
263    private fun checkPost(post: Post) {
264        if (post.isEdited) {
265            updatePost(post)
266        } else {
267            removePost(post)
268       }
269    }
269
270    private fun checkPost(post: Post) {
271        if (post.isEdited) {
272            updatePost(post)
273        } else {
274            removePost(post)
275       }
276    }
276
277    private fun checkPost(post: Post) {
278        if (post.isEdited) {
279            updatePost(post)
280        } else {
281            removePost(post)
282       }
283    }
283
284    private fun checkPost(post: Post) {
285        if (post.isEdited) {
286            updatePost(post)
287        } else {
288            removePost(post)
289       }
290    }
290
291    private fun checkPost(post: Post) {
292        if (post.isEdited) {
293            updatePost(post)
294        } else {
295            removePost(post)
296       }
297    }
297
298    private fun checkPost(post: Post) {
299        if (post.isEdited) {
300            updatePost(post)
301        } else {
302            removePost(post)
303       }
304    }
304
305    private fun checkPost(post: Post) {
306        if (post.isEdited) {
307            updatePost(post)
308        } else {
309            removePost(post)
310       }
311    }
311
312    private fun checkPost(post: Post) {
313        if (post.isEdited) {
314            updatePost(post)
315        } else {
316            removePost(post)
317       }
318    }
318
319    private fun checkPost(post: Post) {
320        if (post.isEdited) {
321            updatePost(post)
322        } else {
323            removePost(post)
324       }
325    }
325
326    private fun checkPost(post: Post) {
327        if (post.isEdited) {
328            updatePost(post)
329        } else {
330            removePost(post)
331       }
332    }
332
333    private fun checkPost(post: Post) {
334        if (post.isEdited) {
335            updatePost(post)
336        } else {
337            removePost(post)
338       }
339    }
3
```

Edit Postingan



Hapus Postingan

