LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3

BUILD A SCROLLABLE LIST

Oleh:

Muhammad Fauzan Ahsani NIM. 2310817310009

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2025

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 3: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Prakitkum ini dikerjakan oleh:

Nama Praktikan : Muhammad Fauzan Ahsani
NIM : 2310817310009

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Zulfa Auliya Akbar NIM. 2210817210026	Muti`a Maulida S.Kom M.T.I NIP. 19881027 201903 20 13

DAFTAR ISI

LEMBAR PENGESAHAN 2
DAFTAR ISI 3
DAFTAR GAMBAR 4
DAFTAR TABEL 5
SOAL 1 6
A. Source Code 6
B. Output Program 22
C. Pembahasan 26
D. Tautan Git 31
SOAL 2 32
A. Pembahasan 32

DAFTAR GAMBAR

Gambar 1. Tampilan Halaman Utama dengan Light Mode 22
Gambar 2. Tampilan Halaman Detail dengan Light Mode 23
Gambar 3. Tampilan Halaman Utama dengan Dark Mode 24
Gambar 4. Tampilan Halaman Detail dengan Dark Mode 25

DAFTAR TABEL

Tabel 1. Source Code MainActivity.kt 6
Tabel 2. Source Code Club.kt 7
Tabel 3. Source Code Competitions.kt 7
Tabel 4. Source Code ClubConstant.kt 8
Tabel 5. Source Code CompetitionsConstant.kt 9
Tabel 6. Source Code AppNavHost.kt 9
Tabel 7. Source Code ClubViewModel.kt 11
Tabel 8. Source Code ClubListScreen.kt 12
Tabel 9. Source Code ClubDetailsScreen.kt 15

SOAL 1

Buatlah sebuah aplikasi Android menggunakan XML atau Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:

1. List menggunakan fungsi RecyclerView (XML) atau LazyColumn (Compose)
2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
3. Item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah
4. Terdapat 2 button dalam list, dengan fungsi berikut:
   a. Button pertama menggunakan intent eksplisit untuk membuka aplikasi atau browser lain.
   b. Button kedua menggunakan Navigation component/intent untuk membuka laman detail item.
5. Sudut item pada list dan gambar di dalam list melengkung atau rounded corner
6. menggunakan Radius
7. Saat orientasi perangkat berubah/dirotasi, baik ke portrait maupun landscape, aplikasi
8. responsif dan dapat menunjukkan list dengan baik. Data di dalam list tidak boleh hilang
9. Aplikasi menggunakan arsitektur single activity (satu activity memiliki beberapa fragment)
10. Aplikasi berbasis XML harus menggunakan ViewBindingSource Code

A. Source Code

1. MainActivity.kt

Tabel 1. Source Code MainActivity.kt

```
 1 package com.example.treblewinner
 2
 3 import android.os.Bundle
 4 import androidx.activity.ComponentActivity
 5 import androidx.activity.compose.setContent
 6 import androidx.activity.enableEdgeToEdge
 7 import com.example.treblewinner.ui.theme.TrebleWinnerTheme
 8 import androidx.navigation.compose.rememberNavController
 9 import
   com.bumptech.glide.integration.compose.ExperimentalGlideCompo
   seApi
10 import com.example.treblewinner.navigation.AppNavHost
11
12 class MainActivity : ComponentActivity() {
13     @OptIn(ExperimentalGlideComposeApi::class)
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
```

```
16          enableEdgeToEdge()
17          setContent {
18              TrebleWinnerTheme {
19                  val navController = rememberNavController()
20                  AppNavHost(navController = navController)
21              }
22          }
23      }
24  }
```

2. Club.kt

Tabel 2. Source Code Club.kt

```
 1  package com.example.treblewinner.model
 2
 3  import android.os.Parcelable
 4  import kotlinx.parcelize.Parcelize
 5
 6  @Parcelize
 7  data class Club (
 8      val name: String,
 9      val country: String,
10      val confederation: String,
11      val trebleYears: List<String>,
12      val competitions: List<Competition>,
13      val logoUrl: String,
14      val webUrl: String,
15      val description: String
16  ): Parcelable
```

3. Competition.kt

Tabel 3. Source Code Competitions.kt

```
 1  package com.example.treblewinner.model
 2
 3  import android.os.Parcelable
 4  import kotlinx.parcelize.Parcelize
 5
 6  @Parcelize
 7  data class Competition (
 8      val name: String,
 9      val country: String,
10      val confederation: String,
11      val logoUrl: String,
12      val logoUrlDark: String
13  ): Parcelable
```

4. ClubConstant.kt

Tabel 4. Source Code ClubConstant.kt

```
 1 package com.example.treblewinner.constants
 2
 3 import com.example.treblewinner.model.Club
 4 import kotlin.reflect.full.memberProperties
 5
 6 object ClubConstant {
 7     val BARCELONA = Club(
 8         name = "FC Barcelona",
 9         country = "Spain",
10         confederation = "UEFA",
11         trebleYears = listOf("2009", "2015"),
12         competitions = listOf(
13             CompetitionsConstant.LA_LIGA,
14             CompetitionsConstant.COPA_DEL_REY,
15             CompetitionsConstant.UEFA_CHAMPIONS_LEAGUE
16         ),
17         logoUrl =
   "https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
   VvXsEh4xprb5TfHqTe6hCvl4hiV6pdlgPfiG_722ZGkfNOPbK1K7bWr
   klpdZ2wMR_qvSuCSpXuLKMSGAH7IhB9PY61vG5ctNQ4-R-Je18Uq5-
   oYEN8pfP0z7c7-EtQE_gjr-
   iDR2D3t6F26mr8/s16000/FC+Barcelona.png",
18         webUrl = "https://www.fcbarcelona.com",
19         description = """
               In 2009, FC Barcelona, under the management
   of Pep Guardiola, achieved a historic treble, becoming
   the first Spanish club to do so. The team secured the
   La Liga title with a blend of attacking flair and
   defensive solidity. In the Copa del Rey final,
   Barcelona dominated Athletic Bilbao with a 4-1 victory,
   showcasing their superiority in domestic competitions.
   The pinnacle of their season was the UEFA Champions
   League final, where they faced Manchester United.
   Barcelona triumphed 2-0, with goals from Samuel Eto'o
   and Lionel Messi, completing a season that redefined
   modern football with their tiki-taka style and cohesive
   team play.
20
21             Barcelona replicated their treble success
   in 2015 under manager Luis Enrique, becoming the first
   European club to achieve this feat twice. They clinched
   the La Liga title, demonstrating consistency and
   resilience throughout the season. In the Copa del Rey
   final, Barcelona defeated Athletic Bilbao 3-1, with
```

```
    Lionel Messi delivering a standout performance. The
    UEFA Champions League final saw Barcelona overcome
    Juventus 3-1, with goals from Ivan Rakitić, Luis
    Suárez, and Neymar. This treble was marked by the
    formidable attacking trio of Messi, Suárez, and Neymar,
    who were instrumental in Barcelona's dominance across
    all competitions.
        """.trimIndent()
    )
}
```

5. CompetitionsConstant.kt

Tabel 5. Source Code CompetitionsConstant.kt

```
package com.example.treblewinner.constants

import com.example.treblewinner.model.Competition

object CompetitionsConstant{
    val UEFA_CHAMPIONS_LEAGUE = Competition(
        name = "UEFA Champions League",
        country = "Europe",
        confederation = "UEFA",
        logoUrl =
    "https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
    VvXsEiWR16FIxnI46HJeFwoEVVS7S8D_3XXpHo0LYEPEM-
    6DFks_dqRDrixc1LC065bDaKrLo9Rbh-
    AlY67dr7kQrH_zdzetMZ_bGDW686hZJXo1RBsS-
    X_xOjauC6QyXkKI09euk88wrxphFg/s16000/UEFACL.png",
        logoUrlDark =
    "https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
    VvXsEg7QlZm7AO2tnSmuc3FkayP98OGUbZ2ly2t5idIn9NtmX6r3Iph
    S1ifIASwkMkEJUYfuRofPHC8c3RPQ9xxpcYUnAE8-
    5dhTPCVOhfn8p6gKG1GHQrHJ59BAUQJYw7uKAZPXt6jpREPDfY/s512
    /UCL.png"
    )
}
```

6. AppNavHost.kt

Tabel 6. Source Code AppNavHost.kt

```
package com.example.treblewinner.navigation

import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxSize
```

```
 5 import androidx.compose.material3.Text
 6 import androidx.compose.runtime.Composable
 7 import androidx.compose.ui.Alignment
 8 import androidx.compose.ui.Modifier
 9 import androidx.lifecycle.viewmodel.compose.viewModel
10 import androidx.navigation.NavHostController
11 import androidx.navigation.compose.NavHost
12 import androidx.navigation.compose.composable
13 import
   com.example.treblewinner.screen.clubdetails.ClubDetailS
   creen
14 import
   com.example.treblewinner.screen.clublist.ClubListScreen
15 import com.example.treblewinner.screen.ClubViewModel
16
17 @Composable
18 fun AppNavHost(navController: NavHostController) {
19
20     val clubVM: ClubViewModel = viewModel()
21
22     NavHost(
23         navController = navController,
24         startDestination = "club_list"
25     ) {
26         composable("club_list") {
27             ClubListScreen(navController =
   navController, viewModel = clubVM)
28         }
29
30         composable("club_detail") {
31             if (clubVM.selectedClub != null) {
32                 ClubDetailScreen(navController =
   navController, clubVM = clubVM)
33             } else {
34                 Box(modifier = Modifier.fillMaxSize(),
   contentAlignment = Alignment.Center) {
35                     Text("No club selected")
36                 }
37             }
38         }
39     }
40 }
```

7. ClubViewModel.kt

Tabel 7. Source Code ClubViewModel.kt

```
 1 package com.example.treblewinner.screen
 2
 3 import androidx.compose.runtime.getValue
 4 import androidx.compose.runtime.mutableStateOf
 5 import androidx.compose.runtime.setValue
 6 import androidx.lifecycle.ViewModel
 7 import androidx.lifecycle.viewModelScope
 8 import com.example.treblewinner.constants.ClubConstant
 9 import com.example.treblewinner.model.Club
10 import kotlinx.coroutines.flow.MutableStateFlow
11 import kotlinx.coroutines.flow.StateFlow
12 import kotlinx.coroutines.flow.asStateFlow
13 import kotlinx.coroutines.launch
14
15 class ClubViewModel: ViewModel() {
16     private val _clubs =
   MutableStateFlow<List<Club>>(emptyList())
17     val clubs: StateFlow<List<Club>> get() =
   _clubs.asStateFlow()
18
19     var selectedClub by mutableStateOf<Club?>(null)
20         private set
21
22     fun selectClub(club: Club){
23         selectedClub = club
24     }
25
26     init {
27         loadData()
28     }
29
30     private fun loadData() {
31         viewModelScope.launch {
32             _clubs.value = ClubConstant.ALL
33         }
34     }
35 }
```

8. ClubListScreen.kt

Tabel 8. Source Code ClubListScreen.kt

```
 1 package com.example.treblewinner.screen.clublist
 2
 3 import android.content.ActivityNotFoundException
 4 import android.content.Intent
 5 import android.widget.Toast
 6 import androidx.compose.foundation.layout.Arrangement
 7 import androidx.compose.foundation.layout.Column
 8 import androidx.compose.foundation.layout.Row
 9 import androidx.compose.foundation.layout.Spacer
10 import androidx.compose.foundation.layout.aspectRatio
11 import androidx.compose.foundation.layout.fillMaxWidth
12 import androidx.compose.foundation.layout.height
13 import androidx.compose.foundation.layout.padding
14 import androidx.compose.foundation.layout.size
15 import androidx.compose.foundation.lazy.LazyColumn
16 import androidx.compose.foundation.lazy.items
17 import androidx.compose.material3.Button
18 import androidx.compose.material3.Card
19 import
   androidx.compose.material3.ExperimentalMaterial3Api
20 import androidx.compose.material3.Text
21 import androidx.compose.material3.MaterialTheme
22 import androidx.compose.material3.Scaffold
23 import androidx.compose.material3.TopAppBar
24 import androidx.compose.runtime.Composable
25 import androidx.compose.runtime.getValue
26 import androidx.compose.runtime.collectAsState
27 import androidx.compose.ui.Alignment
28 import androidx.compose.ui.Modifier
29 import androidx.compose.ui.platform.LocalContext
30 import androidx.compose.ui.tooling.preview.Preview
31 import androidx.compose.ui.unit.dp
32 import androidx.lifecycle.viewmodel.compose.viewModel
33 import androidx.navigation.NavController
34 import
   androidx.navigation.compose.rememberNavController
35 import
   com.bumptech.glide.integration.compose.ExperimentalGli
   deComposeApi
36 import
   com.bumptech.glide.integration.compose.GlideImage
37 import com.example.treblewinner.screen.ClubViewModel
38 import android.net.Uri
39
```

```
@OptIn(ExperimentalGlideComposeApi::class,
ExperimentalMaterial3Api::class)
@Composable
fun ClubListScreen(
    viewModel: ClubViewModel = viewModel(),
    navController: NavController
) {
    val clubs by
viewModel.clubs.collectAsState(emptyList())
    val context = LocalContext.current

    Scaffold(
        topBar = {
            TopAppBar(
                title = {
                    Text(text = "Treble Winner Clubs
List")
                }
            )
        }
    ) { paddingValues ->
        LazyColumn(
            contentPadding = paddingValues,
            verticalArrangement =
Arrangement.spacedBy(4.dp)
        ) {
            items(clubs) { club ->
                Card(
                    modifier = Modifier
                        .fillMaxWidth()
                        .padding(6.dp)
                ) {
                    Row(
                        modifier = Modifier
                            .fillMaxWidth()
                            .padding(8.dp),
                        verticalAlignment =
Alignment.CenterVertically
                    ) {
                        GlideImage(
                            model = club.logoUrl,
                            contentDescription =
club.name,
                            modifier = Modifier
                                .size(150.dp)
                                .aspectRatio(1f)
                                .padding(end = 8.dp)
```

```
 81                          )
 82
 83                          Column(
 84                              modifier =
    Modifier.weight(1f)
 85                          ) {
 86                              Text(
 87                                  text = club.name,
 88                                  style =
    MaterialTheme.typography.headlineSmall
 89                              )
 90                              Text(
 91                                  text =
    club.trebleYears.joinToString(", "),
 92                                  style =
    MaterialTheme.typography.bodyMedium
 93                              )
 94                              Spacer(modifier =
    Modifier.height(8.dp))
 95                              Row(
 96                                  horizontalArrangement
    = Arrangement.spacedBy(8.dp)
 97                              ) {
 98                                  Button(
 99                                      onClick = {
100
    viewModel.selectClub(club)
101
    navController.navigate("club_detail") {
102
    navController.currentBackStackEntry?.arguments?.putPar
    celable(
103
    "club",
104                                                     club
105                                                 )
106                                             }
107                                 }) {
108                                     Text(text =
    "Details")
109                                 }
110                                 Button(
111                                     onClick = {
112                                         try {
113                                             val intent
    = Intent(Intent.ACTION_VIEW, Uri.parse(club.webUrl))
```

```
114
    context.startActivity(intent)
115                                         } catch (e:
    ActivityNotFoundException) {
116
    Toast.makeText(context, "No browser available",
    Toast.LENGTH_SHORT).show()
117                                         }
118                                     },
119                                 ) {
120                                     Text(text = "Visit
    Site")
121                                 }
122                             }
123                         }
124                     }
125                 }
126             }
127         }
128     }
129 }
130
131 @OptIn(ExperimentalGlideComposeApi::class)
132 @Preview(showBackground = true, showSystemUi = true)
133 @Composable
134 fun ClubListPreview(){
135     val fakeViewModel = ClubViewModel()
136     ClubListScreen(viewModel = fakeViewModel,
    navController = rememberNavController())
137 }
```

9. ClubDetailsScreen.kt

Tabel 9. Source Code ClubDetailsScreen.kt

```
  1 package com.example.treblewinner.screen.clubdetails
  2
  3 import android.content.Context
  4 import android.content.res.Configuration
  5 import  ndroid.compose.foundation.BorderStroke
  6 import  ndroid.compose.foundation.horizontalScroll
  7 import  ndroid.compose.foundation.layout.*
  8 import  ndroid.compose.foundation.rememberScrollState
  9 import  ndroid.compose.foundation.verticalScroll
 10 import  ndroid.compose.material.icons.Icons
 11 import  ndroid.compose.material.icons.filled.ArrowBack
    import  ndroid.compose.material3.*
```

```
12 import  ndroid.compose.runtime.Composable
13 import  ndroid.compose.ui.Alignment
14 import  ndroid.compose.ui.Modifier
15 import  ndroid.compose.ui.layout.ContentScale
16 import  ndroid.compose.ui.text.font.FontWeight
17 import  ndroid.compose.ui.tooling.preview.Preview
18 import  ndroid.compose.ui.unit.dp
19 import  ndroid.navigation.NavController
20 import
21 com.bumptech.glide.integration.compose.ExperimentalGli
   deComposeApi
   import
22 com.bumptech.glide.integration.compose.GlideImage
   import com.example.treblewinner.constants.ClubConstant
23 import com.example.treblewinner.model.Club
24 import com.example.treblewinner.screen.ClubViewModel
25 import  ndroid.compose.foundation.layout.Arrangement
26 import
27  ndroid.compose.foundation.shape.RoundedCornerShape
   import
28  ndroid.compose.material.icons.automirrored.filled.Arr
   owBack
   import  ndroid.compose.material.icons.filled.ArrowBack
29 import  ndroid.compose.ui.graphics.Color
   import  ndroid.compose.ui.text.style.TextAlign
30 import  ndroid.compose.ui.platform.LocalContext
31 import  ndroid.compose.ui.unit.sp
32
33 @Composable
34 fun ClubDetailScreen(
35     navController: NavController,
36     clubVM: ClubViewModel
37 ) {
38     val club = clubVM.selectedClub
39
40     if (club == null) {
41         Text("No club selected")
42     } else {
43         ClubDetailContent(club = club) {
44             navController.popBackStack()
45         }
46     }
47 }
48
49 fun Context.isDarkTheme(): Boolean {
50
51
```

```
    return (resources.configuration.uiMode and
Configuration.UI_MODE_NIGHT_MASK) ==
Configuration.UI_MODE_NIGHT_YES
}

@OptIn(ExperimentalMaterial3Api::class,
ExperimentalGlideComposeApi::class)
@Composable
fun ClubDetailContent(
    club: Club,
    onBack: () -> Unit = {}
) {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text(text = club.name) },
                navigationIcon = {
                    IconButton(onClick = { onBack() })
{
                        Icon(imageVector =
Icons.AutoMirrored.Filled.ArrowBack,
contentDescription = "Back")
                    }
                }
            )
        }
    ) { padding ->
        Column(
            modifier = Modifier
                .verticalScroll(rememberScrollState())
                .padding(16.dp)
                .padding(padding)
        ) {
            GlideImage(
                model = club.logoUrl,
                contentDescription = "${club.name}
logo",
                modifier = Modifier
                    .fillMaxWidth()
                    .height(200.dp),
                contentScale = ContentScale.Fit
            )

            Spacer(modifier = Modifier.height(24.dp))

            Text(
                text = club.name,
```

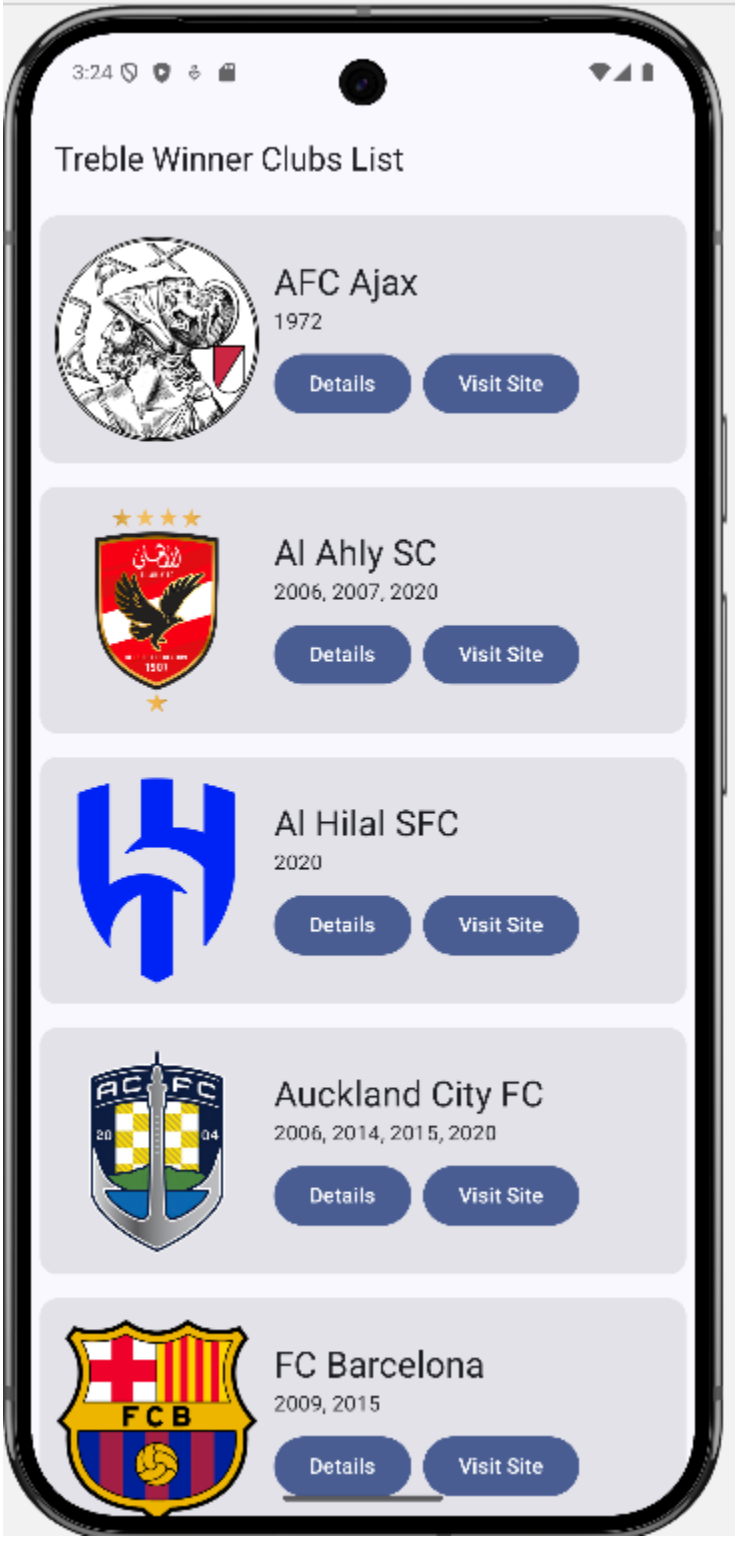
```
 91                 style =
 92 MaterialTheme.typography.headlineMedium,
                    fontWeight = FontWeight.Bold
 93             )
 94
 95             Spacer(modifier = Modifier.height(8.dp))
 96
 97             Row(
 98                 horizontalArrangement =
 99 Arrangement.spacedBy(8.dp),
                    verticalAlignment =
100 Alignment.CenterVertically
               ) {
101                 Text(text = club.country, style =
102 MaterialTheme.typography.bodyLarge)
                    Text(text = "•", style =
103 MaterialTheme.typography.bodyLarge)
                    Text(text = club.confederation, style
104 = MaterialTheme.typography.bodyLarge)
               }
105
106            Spacer(modifier = Modifier.height(16.dp))
107
108            Text(
109                text = "Treble Years:",
110                style =
111 MaterialTheme.typography.titleMedium,
                   fontWeight = FontWeight.Bold
112            )
113            Spacer(modifier = Modifier.height(4.dp))
114
115            Row(
116                modifier = Modifier
117                    .fillMaxWidth(),
118                horizontalArrangement =
    Arrangement.spacedBy(16.dp,
119 Alignment.CenterHorizontally)
               ){
                   club.trebleYears.forEach { year ->
120                    YearTagOutlined(year)
121                }
122            }
123
124
125            Spacer(modifier = Modifier.height(16.dp))
126
127            Text(
```

```
                text = "Competitions Won:",
                style =
MaterialTheme.typography.titleMedium,
                fontWeight = FontWeight.Bold
            )
            Spacer(modifier = Modifier.height(4.dp))

            Row (
                modifier = Modifier
                    .fillMaxWidth(),
                horizontalArrangement =
Arrangement.spacedBy(16.dp,
Alignment.CenterHorizontally)
            ){
                club.competitions.forEach {
competition ->
                    Card (
                        modifier = Modifier
                            .width(100.dp)
                            .height(200.dp),
                        shape =
MaterialTheme.shapes.medium,
                        elevation =
CardDefaults.cardElevation(defaultElevation = 4.dp)
                    ) {
                        Column (
                            modifier = Modifier
                                .fillMaxSize(),
                            verticalArrangement =
Arrangement.Top,
                            horizontalAlignment =
Alignment.CenterHorizontally

                        ) {
                            Spacer(modifier =
Modifier.height(10.dp))

                            val context =
LocalContext.current
                            val logo = if
(context.isDarkTheme() &&
!competition.logoUrlDark.isNullOrBlank()) {

competition.logoUrlDark
                            } else {
                                competition.logoUrl
                            }
```

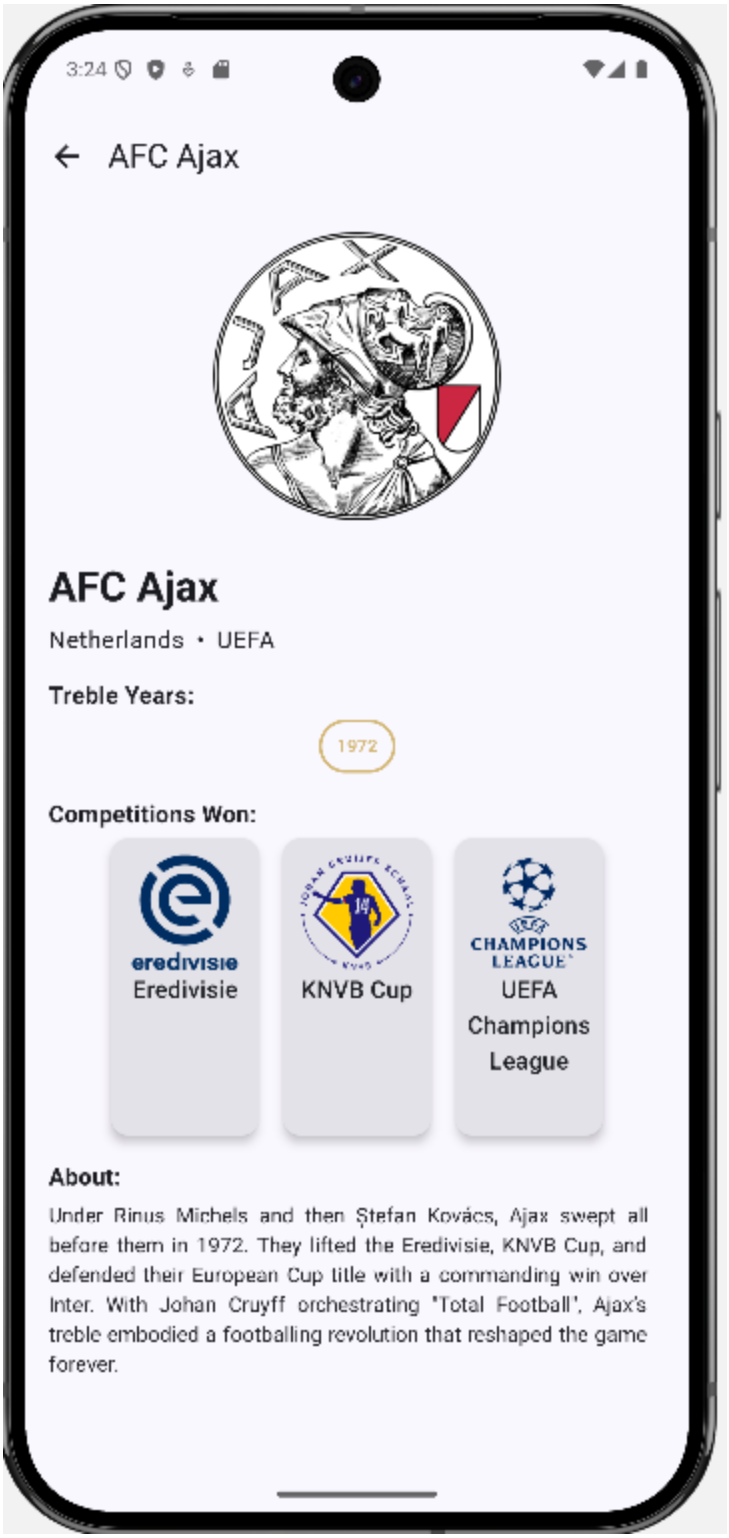
```
162
163                                 GlideImage(
164                                     model = logo,
165                                     contentDescription =
166 competition.name,
167                                     modifier =
    Modifier.size(80.dp),
168                                     contentScale =
    ContentScale.Fit
169                                 )
                                    Text(
170                                     text =
171 competition.name,
172                                     style =
    MaterialTheme.typography.titleMedium,
173                                     textAlign =
    TextAlign.Center
174                                 )
                                }
175                         }
176                     }
177                 }
178
179                 Spacer(modifier = Modifier.height(16.dp))
180
181                 Text(
182                     text = "About:",
183                     style =
184 MaterialTheme.typography.titleMedium,
185                     fontWeight = FontWeight.Bold
                    )
186                 Spacer(modifier = Modifier.height(4.dp))
187                 Text(
188                     text = club.description,
189                     textAlign = TextAlign.Justify,
190                     style =
191 MaterialTheme.typography.bodyMedium
192                 )
            }
193     }
194 }
195
196 @Composable
197 fun YearTagOutlined(year: String) {
198     val tagColor = Color(0xFFD2B571)
199     Surface(
200         shape = RoundedCornerShape(50),
```

```
201         border = BorderStroke(2.dp, tagColor),
202         color = Color.Transparent,
203         tonalElevation = 1.dp // Optional
204     ) {
205         Text(
206             text = year,
207             style =
208 MaterialTheme.typography.titleMedium,
209             modifier = Modifier.padding(horizontal =
    12.dp, vertical = 6.dp),
210             fontSize = 12.sp,
                color = tagColor
211         )
212     }
213 }
214
215 @Preview(showBackground = true)
216 @Composable
217 fun ClubDetailContentPreview() {
218     val mockClub = ClubConstant.BARCELONA
219     ClubDetailContent(club = mockClub)
220 }
221
222
```

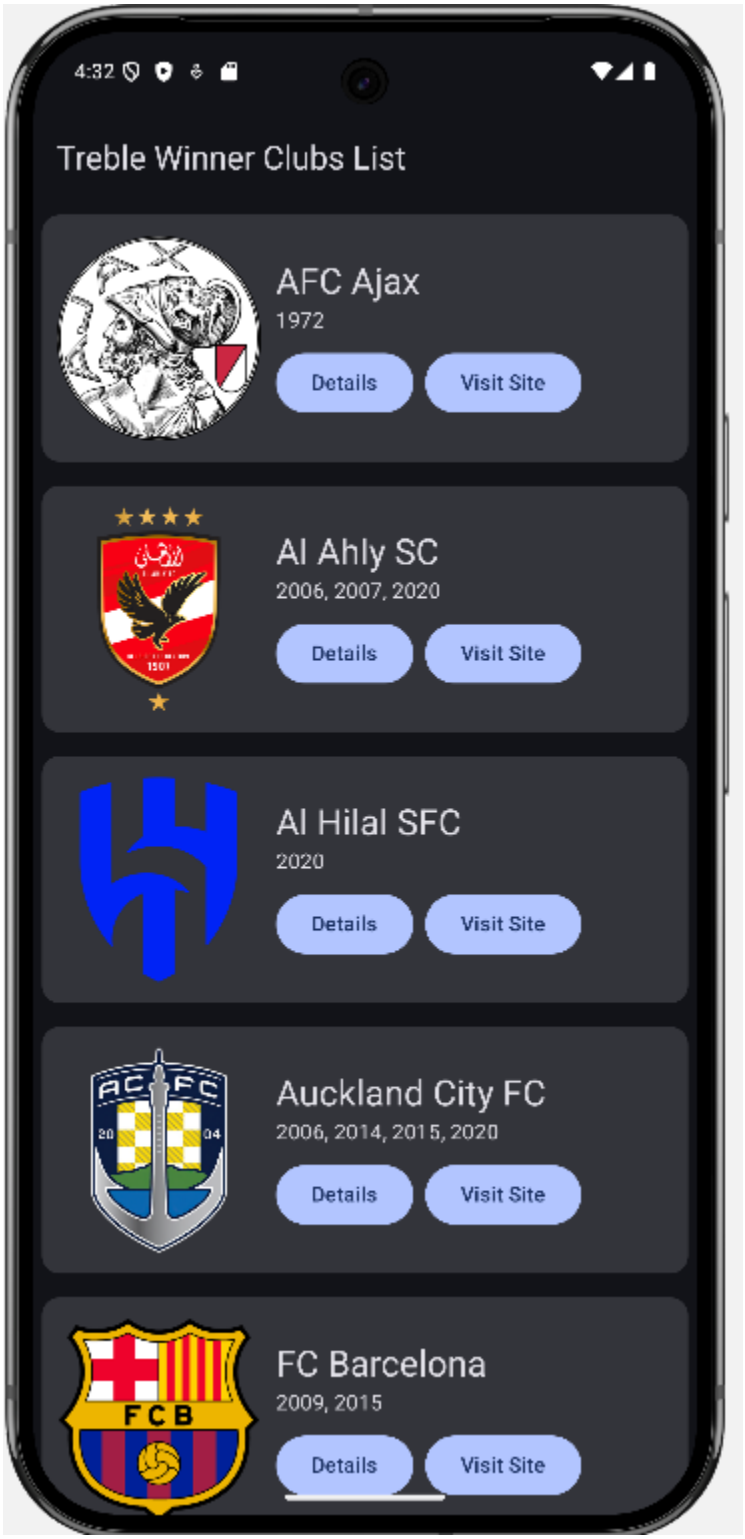
B. Output Program



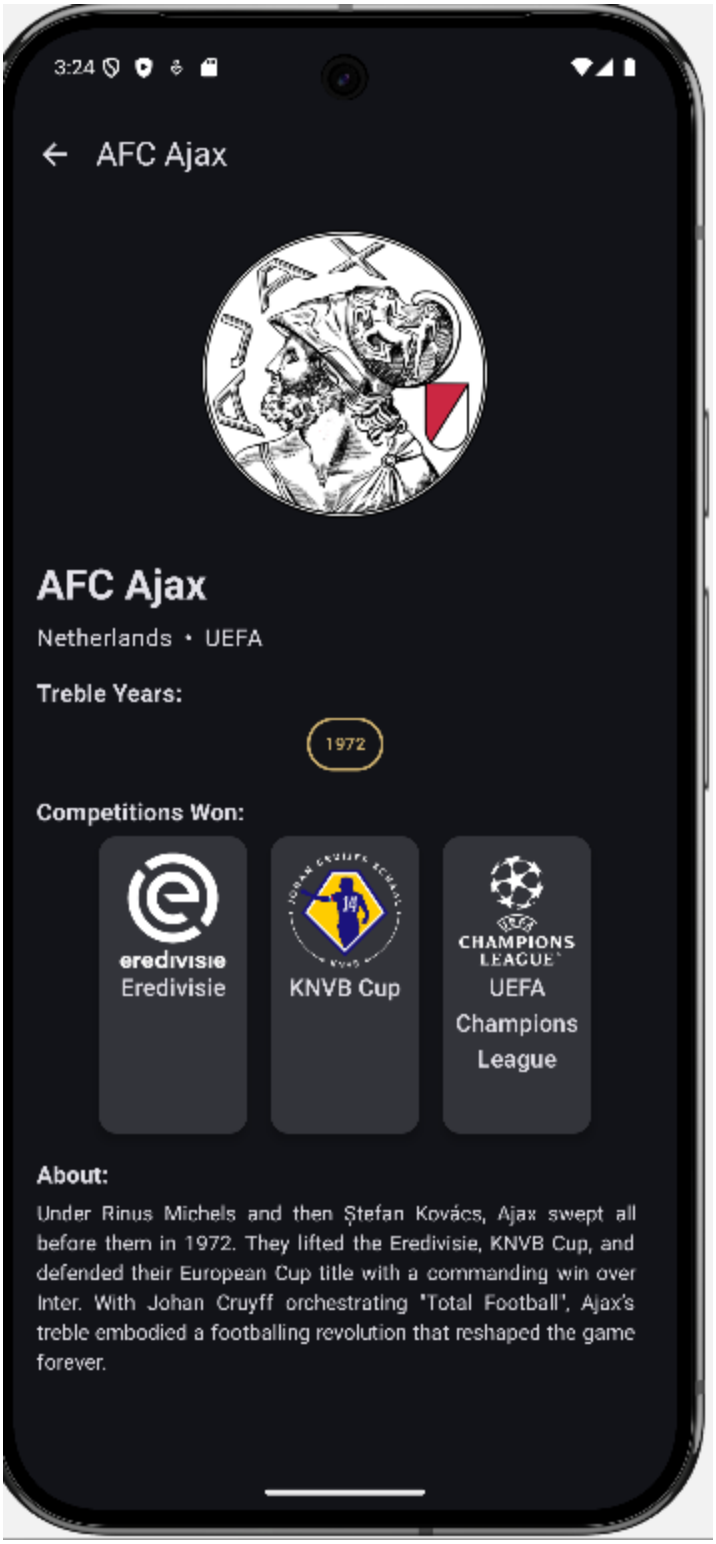
Gambar 1. Tampilan Halaman Utama dengan Light Mode



Gambar 2. Tampilan Halaman Detail dengan Light Mode



Gambar 3. Tampilan Halaman Utama dengan Dark Mode



Gambar 4. Tampilan Halaman Detail dengan Dark Mode

C. Pembahasan

Saya membuat scrollable list yang berisikan klub-klub sepakbola yang pernah Treble Continental. Treble Continental adalah pencapaian yang sangat langka dalam dunia sepakbola, di mana sebuah klub berhasil memenangkan tiga gelar utama dalam satu musim: liga domestik, piala domestik utama, dan kompetisi antarklub benua.

Logo-logo klub tersebut saya dapatkan melalui situs komunitas modder game PES yang menyediakan logo-logo kluub dengan kualitas tinggi. Beberapa pengecualian karena terdapat beberapa klub atau kompetisi yang kurang popular sehingga perlu menyelam ke DeepWeb demi mendapatkan aset grafis yang layak. Beberepa logo kompetisi memiliki versi resmi khusus untuk mode gelap (dark mode), dan saya pun menambahkan atribut tersebut dalam model agar tampil optimal dalam berbagai tema aplikasi. Meskipun demikian, klub-klub yang berhasil meraih Treble kebetulan tidak memiliki versi logo khusus untuk dark mode, sehingga tidak ditambahkan atribut untuk kasus logo mode gelap.

1. MainActivity.kt:

Pertama, mendeklarasikan nama package `com.example.treblewinner`, lalu mengimpor berbagai library yang dibutuhkan, termasuk untuk komponen Activity, Jetpack Compose, Navigation, dan Glide Compose. Kelas `MainActivity` meng-extend `ComponentActivity`, yang merupakan turunan dari Activity.

Di dalam metode `onCreate()`, fungsi `enableEdgeToEdge()` dipanggil terlebih dahulu untuk mengaktifkan tampilan layar penuh hingga ke tepi layar, termasuk area status bar dan navigasi. Kemudian, UI aplikasi diatur melalui `setContent`. Di dalamnya, seluruh konten UI dibungkus oleh `TrebleWinnerTheme`.

Selanjutnya, sebuah instance `NavController` dibuat menggunakan `rememberNavController()` yang berfungsi untuk mengelola navigasi antar layar (screen) dalam Compose. Objek ini kemudian diteruskan ke fungsi `AppNavHost`, sebuah composable khusus yang mendefinisikan struktur navigasi aplikasi, termasuk daftar screen yang tersedia dan bagaimana perpindahan antar screen dilakukan. Anotasi `@OptIn(ExperimentalGlideComposeApi::class)` ditambahkan untuk mengizinkan penggunaan API eksperimental dari Glide Compose (karena Glide masih beta di Compose), yang digunakan untuk menampilkan gambar logo klub dan kompetisi sepakbola dalam daftar yang ditampilkan aplikasi.

2. Club.kt

Pertama, mendeklarasikan nama package `com.example.treblewinner.model`, lalu mengimpor library yang diperlukan seperti `Parcelable` dari Android dan anotasi `@Parcelize` dari Kotlin untuk mempermudah proses pengiriman (passing) data antar komponen Android. Di dalam file ini, dibuat sebuah `data class` bernama `Club` yang merepresentasikan informasi lengkap mengenai klub sepakbola yang pernah meraih Treble Continental. Anotasi `@Parcelize` digunakan agar objek `Club` dapat diubah menjadi `parcel` dan ditransfer antar `Activity` atau `Composable` dengan mudah tanpa harus menuliskan implementasi `Parcelable` secara manual.

Kelas `Club` memiliki beberapa properti penting yang mendukung penyajian data secara menyeluruh. Properti `name` menyimpan nama klub, `country` menunjukkan asal negara klub, dan `confederation` menjelaskan konfederasi sepakbola tempat klub tersebut bernaung seperti UEFA (Eropa) atau CONMEBOL (Amerika Selatan). Informasi tahun-tahun di mana klub meraih Treble Continental disimpan dalam bentuk list string pada properti `trebleYears`. Selain itu, properti `competitions` menyimpan daftar kompetisi yang dimenangkan klub dalam musim treble, yang ditulis dalam bentuk list dari objek `Competition`. Properti `logoUrl` yang menunjuk ke gambar logo klub, serta `webUrl` sebagai tautan ke situs resmi klub. Terakhir, properti `description` digunakan untuk menyimpan penjelasan singkat atau latar belakang dari klub tersebut.

3. Competition.kt

Pertama, mendeklarasikan nama package `com.example.treblewinner.model`, kemudian mengimpor library yang dibutuhkan, yaitu `Parcelable` dari Android dan anotasi `@Parcelize` dari Kotlin Extensions. Di dalam file ini didefinisikan sebuah `data class` bernama `Competition` yang digunakan untuk merepresentasikan informasi dari kompetisi sepakbola yang menjadi bagian dari Treble Continental yang diraih oleh sebuah klub. Dengan menggunakan anotasi `@Parcelize`, objek dari kelas ini dapat secara otomatis dikonversi menjadi parcel, sehingga memudahkan proses pengiriman data antar komponen dalam aplikasi Android, seperti saat berpindah dari satu layar ke layar lainnya.

Kelas `Competition` terdiri dari beberapa properti yang menyimpan data penting tentang kompetisi tersebut. Properti `name` menyimpan nama kompetisi, `country` menyimpan nama negara penyelenggara atau lokasi, dan `confederation` menunjukkan konfederasi sepakbola yang mengatur kompetisi tersebut. Properti `logoUrl` dan `logoUrlDark` masing-masing menyimpan URL untuk logo kompetisi dalam versi standar dan versi gelap, yang bisa digunakan sesuai tema terang atau gelap yang sedang aktif di aplikasi.

4. ClubConstant.kt

Untuk file ini, jumlah semua objek sebenarnya sangatlah banyak dan memakan hampir 500 baris. Maka diambilah salah satu contoh representasi dari objek untuk mempermudah penulisan.

Pertama, mendeklarasikan nama package `com.example.treblewinner.constants`, lalu mengimpor kelas `Club` dari `package model`. Di dalam file ini dibuat sebuah object bernama `ClubConstant`, yang berfungsi sebagai tempat penyimpanan data konstan atau data statis terkait klub sepakbola. Salah satu entri penting di dalamnya adalah `BARCELONA`, yaitu sebuah instance dari kelas `Club` yang merepresentasikan klub FC Barcelona beserta informasi lengkap terkait raihan Treble Continental mereka.

Objek `BARCELONA` menyimpan berbagai properti yang menjelaskan pencapaian klub. Nama klub diisi dengan "FC Barcelona", negara asalnya adalah Spanyol, dan konfederasinya adalah UEFA. Daftar tahun ketika mereka meraih Treble Continental adalah 2009 dan 2015. Informasi tentang kompetisi yang dimenangkan pada musim-musim tersebut direpresentasikan dalam bentuk list berisi konstanta dari `CompetitionsConstant`, yakni `LA_LIGA`, `COPA_DEL_REY`, dan `UEFA_CHAMPIONS_LEAGUE`. Properti `logoUrl` berisi tautan ke gambar logo klub, sementara `webUrl` menyimpan alamat situs resmi FC Barcelona. `description`, berisi narasi historis lengkap tentang dua musim luar biasa ketika Barcelona meraih treble. Deskripsi ini menjelaskan bagaimana klub mencapai keberhasilan mereka pada tahun 2009 di bawah Pep Guardiola dan tahun 2015 di bawah Luis Enrique, termasuk rincian pertandingan final, gaya bermain, serta pemain-pemain kunci yang berkontribusi pada kejayaan mereka.

5. CompetitionsConstant.kt

Untuk file ini, jumlah semua objek sebenarnya sangatlah banyak dan memakan hampir 500 baris. Maka diambilah salah satu contoh representasi dari objek untuk mempermudah penulisan.

Pertama, mendeklarasikan nama package `com.example.treblewinner.constants`, kemudian mengimpor kelas `Competition` dari package `model`. File ini berfungsi sebagai tempat penyimpanan data konstan terkait kompetisi sepakbola yang menjadi bagian dari Treble Continental. Di dalamnya didefinisikan sebuah `object` bernama `CompetitionsConstant`, yang menyimpan data kompetisi dalam bentuk properti `immutable` agar dapat digunakan secara global di berbagai bagian aplikasi.

Salah satu data yang dideklarasikan dalam objek ini adalah `UEFA_CHAMPIONS_LEAGUE`, yaitu sebuah instance dari kelas `Competition` yang merepresentasikan Liga Champions UEFA, kompetisi antarklub paling prestisius di Eropa. Properti `name` diisi dengan "UEFA Champions League", sementara `country` menunjukkan wilayah cakupannya yaitu Eropa, dan `confederation` adalah UEFA sebagai badan pengatur kompetisi. Untuk mendukung tampilan visual yang sesuai dengan tema aplikasi, disediakan dua URL logo, yakni `logoUrl` untuk tampilan normal dan `logoUrlDark` untuk tampilan mode gelap (dark mode).

6. AppNavHost.kt

Pertama, mendeklarasikan nama package `com.example.treblewinner.navigation`, lalu mengimpor berbagai komponen dari Jetpack Compose dan Android Navigation untuk mendukung navigasi antar tampilan dalam aplikasi. Di dalam file ini terdapat sebuah fungsi `@Composable` bernama `AppNavHost`, yang berfungsi sebagai pusat navigasi (navigation host) aplikasi. Fungsi ini menerima parameter `navController` bertipe `NavHostController`, yang digunakan untuk mengatur perpindahan antar halaman UI (screen).

Di dalam `AppNavHost`, sebuah instance dari `ClubViewModel` diinisialisasi menggunakan fungsi `viewModel()`, yang secara otomatis menghubungkan `ViewModel` dengan recycle composable. Navigasi didefinisikan melalui `NavHost`, dengan `startDestination` diatur ke `"club_list"` sebagai layar pertama yang ditampilkan saat aplikasi dijalankan. Kemudian, dua rute composable didefinisikan: `"club_list"` dan `"club_detail"`.

Rute `"club_list"` akan menampilkan tampilan `ClubListScreen`, yaitu layar utama berisi daftar klub yang pernah meraih Treble Continental, di mana `navController` dan `viewModel` diteruskan sebagai parameter. Sementara itu, rute `"club_detail"` digunakan untuk menampilkan detail klub. Sebelum menampilkan `ClubDetailScreen`, dilakukan pengecekan apakah properti `selectedClub` di dalam `clubVM` sudah tidak `null`, artinya pengguna telah memilih klub dari daftar. Jika belum, maka akan ditampilkan teks "No club selected" yang diposisikan di tengah layar menggunakan `Box` dengan `Modifier.fillMaxSize()` dan `contentAlignment = Alignment.Center`.

7. ClubViewModel.kt

File ini mendefinisikan `ClubViewModel`, sebuah `class` yang meng-extend `ViewModel`, yang merupakan bagian dari arsitektur MVVM pada Android dan digunakan untuk menyimpan serta mengelola data UI yang bersifat tahan terhadap perubahan siklus hidup. Pertama, file ini mendeklarasikan package `com.example.treblewinner.screen`, lalu mengimpor berbagai komponen Compose dan coroutine yang diperlukan untuk manajemen state dan data asynchronous.

Di dalam `ClubViewModel`, terdapat properti `private _clubs` yang bertipe `MutableStateFlow<List<Club>>`, digunakan untuk menyimpan daftar klub secara reaktif. Properti `public clubs` mengekspos versi read-only dari `_clubs` menggunakan `asStateFlow()`, sehingga hanya `ViewModel` yang dapat memodifikasi nilainya, sementara UI hanya dapat mengobservasinya.

Selain itu, terdapat properti `selectedClub` yang bertipe `mutableStateOf<Club?>`, digunakan untuk menyimpan klub yang sedang dipilih oleh pengguna. Properti ini dapat berubah dan otomatis memicu

recomposition pada composable yang mengamatinya. Metode `selectClub(club: Club)` disediakan agar UI dapat menetapkan klub yang dipilih secara eksplisit, namun tidak dapat mengubah langsung nilai `selectedClub` dari luar `ViewModel` karena properti ini memiliki `private set`.

Ketika instance dari `ClubViewModel` dibuat, blok `init` akan langsung memanggil fungsi `loadData()`, yang menjalankan coroutine di dalam `viewModelScope` untuk memuat data klub dari `ClubConstant.ALL`. Data yang dimuat ini akan disimpan ke dalam `_clubs`, dan perubahan ini akan otomatis terpantau oleh komponen UI yang berreferensi ke `clubs`.

8. ClubListScreen.kt

File ini mendefinisikan tampilan utama dari daftar klub sepakbola yang pernah meraih Treble Continental, yaitu `ClubListScreen`. Fungsi ini merupakan composable yang mengambil `ClubViewModel` dan `NavController` sebagai parameter. `ClubViewModel` berperan menyediakan data daftar klub melalui `StateFlow`, yang kemudian dikonversi ke dalam bentuk state Compose menggunakan `collectAsState`. Data ini bersifat reaktif sehingga UI akan diperbarui secara otomatis saat data berubah.

Struktur UI dibangun menggunakan `Scaffold`, dengan `TopAppBar` yang menampilkan judul aplikasi. Konten utama berupa `LazyColumn`, yang digunakan untuk menampilkan daftar klub secara efisien dalam bentuk scrollable list. Setiap klub ditampilkan dalam sebuah `Card` yang di dalamnya memuat gambar logo klub (melalui `GlideImage` dari Glide Compose), nama klub, dan daftar tahun saat klub meraih treble. Layout `card` dibagi menjadi dua kolom utama, yaitu logo di kiri dan teks beserta tombol di kanan.

Dua tombol disediakan untuk setiap klub: tombol "Details" dan "Visit Site". Tombol "Details" akan memanggil fungsi `selectClub()` pada `ViewModel` untuk menyimpan klub yang dipilih, kemudian melakukan navigasi ke halaman detail melalui `NavController`. Sedangkan tombol "Visit Site" akan membuka halaman web resmi klub menggunakan `Intent.ACTION_VIEW`. Jika tidak ada browser yang tersedia, aplikasi akan menampilkan `Toast` untuk memberi tahu pengguna.

Di bagian bawah terdapat fungsi `ClubListPreview`, sebuah composable yang dianotasi dengan `@Preview` untuk menampilkan pratinjau tampilan langsung di Android Studio. Fungsi ini membuat instance sementara `ClubViewModel` dan `NavController`.

9. ClubDetailsScreen.kt

File `ClubDetailScreen.kt` merupakan bagian dari aplikasi untuk menampilkan halaman detail sebuah klub sepak bola yang meraih treble winner. Halaman ini bekerja dengan mengambil data dari `ClubViewModel`, khususnya properti `selectedClub` yang menyimpan klub yang dipilih oleh pengguna dari layar sebelumnya. Bila `selectedClub` bernilai `null`, tampilan hanya

menampilkan teks sederhana “No club selected”, tetapi jika klub telah dipilih, maka komponen `ClubDetailContent` akan digunakan untuk menyusun antarmuka detail klub tersebut secara lengkap dan informatif.

Fungsi `ClubDetailContent` dibangun menggunakan struktur `Scaffold` yang menyediakan `TopAppBar` dengan judul berupa nama klub dan ikon panah kembali yang memungkinkan pengguna kembali ke layar sebelumnya. Di dalam tubuh `Scaffold`, terdapat kolom utama yang dapat discroll secara vertikal, berisi elemen-elemen visual dan teks yang disusun secara proporsional dan estetis. Logo klub ditampilkan di bagian atas dalam ukuran besar menggunakan `GlideImage`, diikuti dengan nama klub dalam ukuran font besar dan gaya bold. Informasi asal negara dan konfederasi klub ditampilkan dalam satu baris menggunakan `Row`, diikuti oleh daftar tahun-tahun di mana klub tersebut berhasil meraih treble. Tahun-tahun ini ditampilkan dalam bentuk tag kecil berbentuk oval yang memiliki border berwarna emas, dibuat melalui fungsi composable `YearTagOutlined`.

Selanjutnya, aplikasi menampilkan bagian kompetisi yang dimenangkan oleh klub tersebut pada musim treble. Masing-masing kompetisi divisualisasikan dalam kartu vertikal kecil dengan logo kompetisi dan nama kompetisinya. Logo yang digunakan dapat berubah sesuai dengan mode terang atau gelap dari sistem, yang ditentukan menggunakan fungsi ekstensi `Context.isDarkTheme()`. Setelah bagian kompetisi, terdapat juga paragraf deskripsi klub yang ditampilkan dalam gaya teks justify, memberikan latar belakang atau cerita singkat mengenai sejarah klub tersebut. Seluruh desain didasarkan pada prinsip-prinsip Material 3 dengan penggunaan komponen modern seperti `Surface`, `Card`, dan `Text` dengan pengaturan padding dan jarak antar elemen yang konsisten.

Selain itu, terdapat juga fungsi pratinjau `ClubDetailContentPreview` yang memungkinkan pengembang melihat tampilan layar ini di Android Studio menggunakan data dari `ClubConstant.BARCELONA`.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/MinamotoYuki46/MeineStudienArbeit/tree/main/MobileDevelopment/Codex-Practicus/Modul%203/TrebleWinner

SOAL 2

Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang Panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?

A. Pembahasan

Ada beberapa alasan. Pertama, `RecyclerView` merupakan komponen mapan dalam pengembangan Android berbasis XML dan View System yang sudah lama dan banyak digunakan. Banyak proyek, librari pihak ketiga, tutorial, dan dokumentasi yang masih mengandalkan `RecyclerView`. Selain itu biaya untuk melakukan migrasi ke Compose tidaklah sedikit.

Kemudian, `RecyclerView` memiliki fleksibilitas tingkat rendah dan kontrol yang sangat detail, seperti penggunaan `DiffUtil`, `ItemAnimator`, `SnapHelper`, dan `ItemDecoration`. Fitur-fitur ini memungkinkan optimisasi performa dan interaksi kompleks yang belum sepenuhnya disediakan atau distabilkan pada `LazyColumn`. Meskipun `LazyColumn` unggul dalam hal kesederhanaan, `RecyclerView` tetap unggul dalam aspek kustomisasi ekstrem yang masih umum di industri.

Kemudian, Compose belum sepenuhnya diadopsi secara universal dan masih berkembang. Meskipun stabil dan direkomendasikan oleh Google, Compose belum dianggap pengganti total View System oleh semua developer. Di banyak perusahaan, terutama yang menerapkan arsitektur hybrid Compose-View, `RecyclerView` tetap menjadi pilihan utama untuk menjaga konsistensi antar layar lama dan baru.