Jetpack Compose ile Android Uygulama Geliştirme Kursu

Tasarım

Kasım ADALAN
Elektronik ve Haberleşme Mühendisi
Android - IOS Developer and Trainer

Eğitim İçeriği

- Tasarım Giriş
- Row – Column – Box
- Arrangement
- Alignment
- FillMax
- Align
- Size
- Spacer
- Padding
- Weight
- Özelleştirilmiş Widget
- Resim Ekleme
- Çoklu Dil Desteği
- Çoklu Ekran Desteği
- Icon Oluşturma

Tasarım

Row – Column - Box

Row

Column

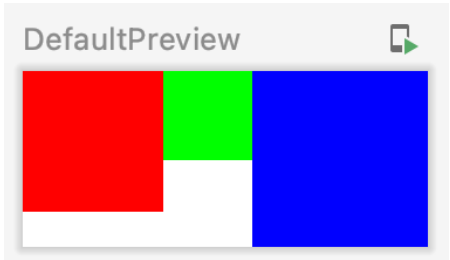
Box

Row

Row

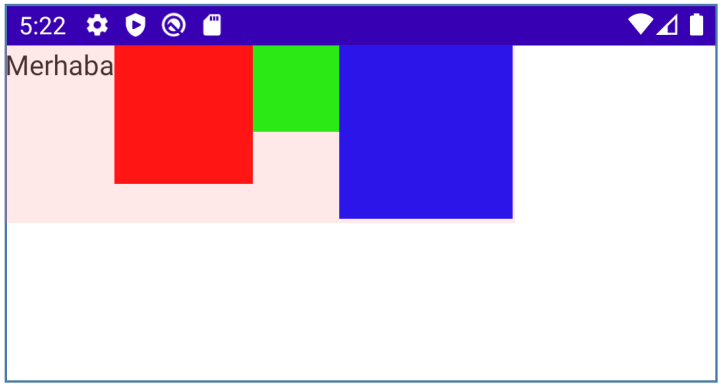
- Widgetların yan yana durmasını sağlar.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.

```
@Composable
fun RowKullanimi(){
    Row{ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

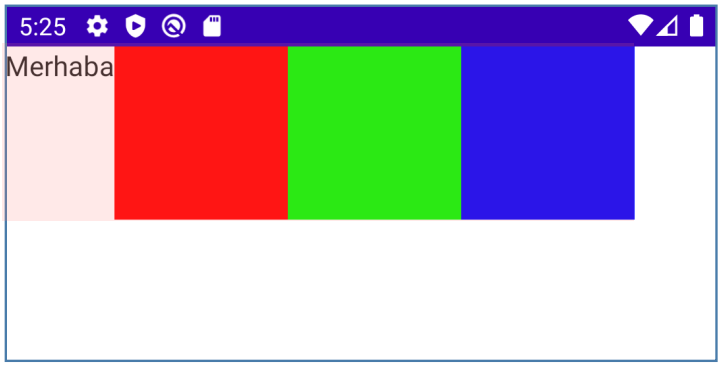


Row

```
Row{ this: RowScope
    Text(text="Merhaba")
    Box(Modifier.size(80.dp).background(Color.Red))
    Box(Modifier.size(50.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```




```
Row{ this: RowScope
    Text(text="Merhaba")
    Box(Modifier.size(100.dp).background(Color.Red))
    Box(Modifier.size(100.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```

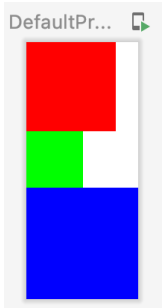


Column

Column

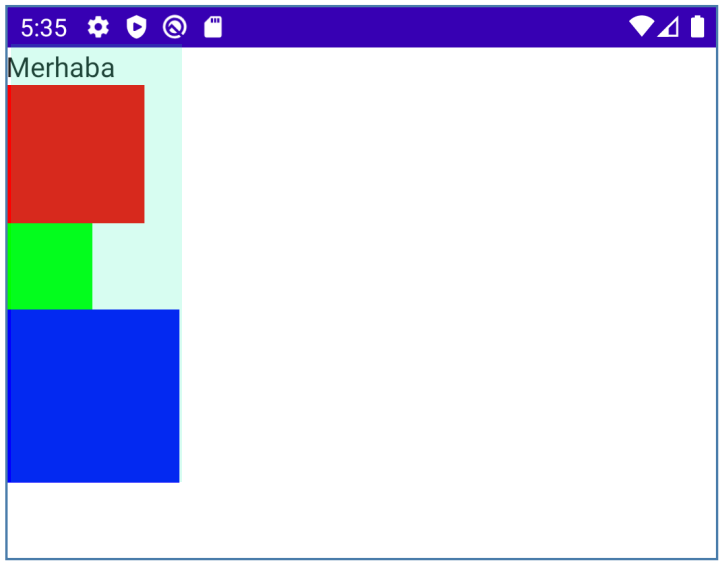
- Widgetların alt alta durmasını sağlar.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.

```
@Composable
fun ColumnKullanimi(){
    Column{ this: ColumnScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

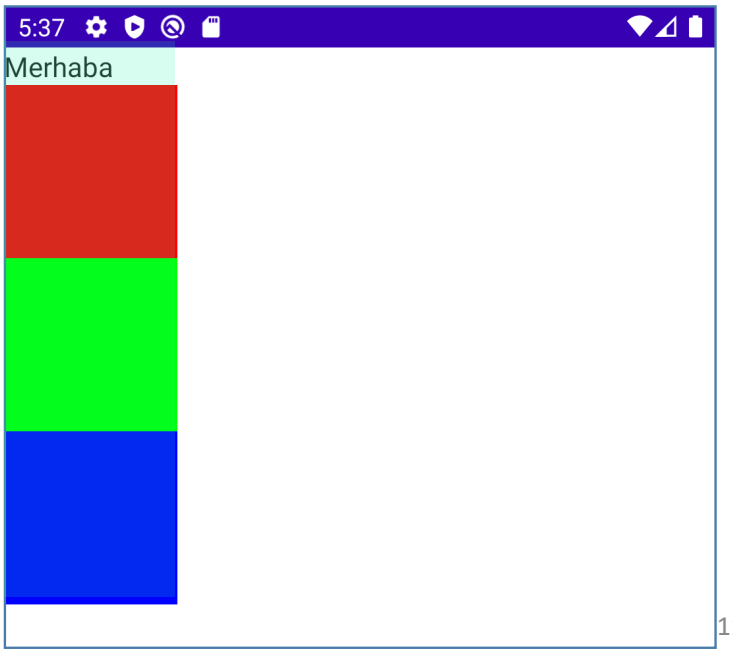


Column

```
Column{ this: ColumnScope
    Text(text="Merhaba")
    Box(Modifier.size(80.dp).background(Color.Red))
    Box(Modifier.size(50.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```



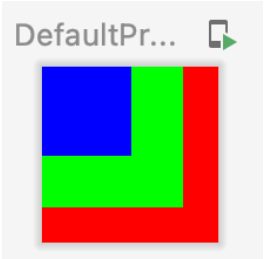

```
Column{ this: ColumnScope
    Text(text="Merhaba")
    Box(Modifier.size(100.dp).background(Color.Red))
    Box(Modifier.size(100.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```



Box

Box

- Widgetların üst üste durmasını sağlar.
- İlk eklenen en altta yer alır.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.
- Box görsel bir kutu olarakta kullanılabilir.

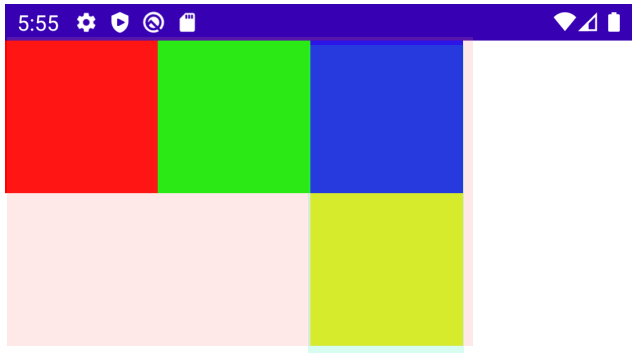


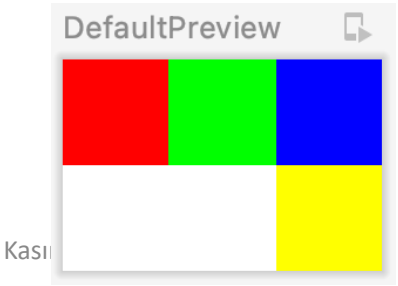

```
@Composable
fun BoxKullanimi(){
    Box{ this: BoxScope
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(80.dp).background(Color.Green))
        Box(Modifier.size(50.dp).background(Color.Blue))
    }
}
```

Karışık Kullanım

Row & Column

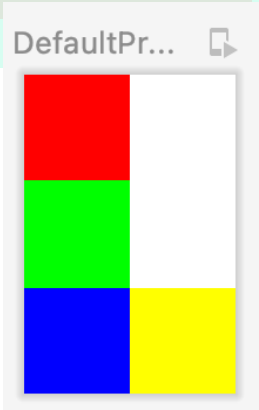
```
@Composable
fun KarisikKullanimi(){
    Row{ this: RowScope
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(100.dp).background(Color.Green))
        Column{ this: ColumnScope
            Box(Modifier.size(100.dp).background(Color.Blue))
            Box(Modifier.size(100.dp).background(Color.Yellow))
        }
    }
}
```

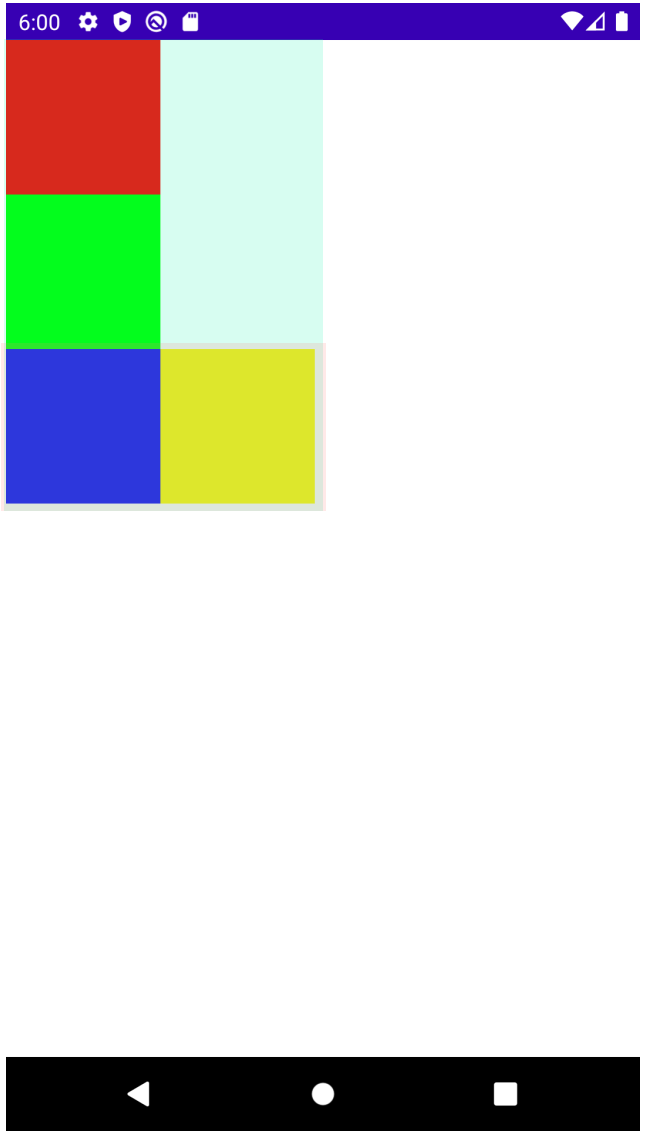




Row & Column

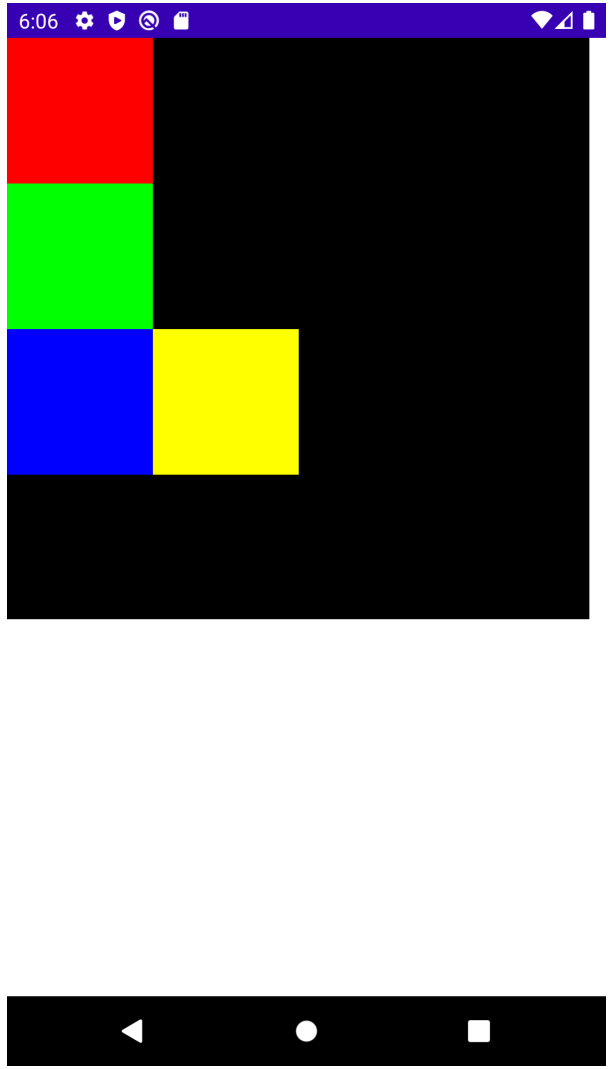
```
@Composable
fun KarisikKullanimi(){
    Column{ this: ColumnScope
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(100.dp).background(Color.Green))
        Row{ this: RowScope
            Box(Modifier.size(100.dp).background(Color.Blue))
            Box(Modifier.size(100.dp).background(Color.Yellow))
        }
    }
}
```





Row – Column - Box

```
@Composable
fun KarisikKullanimi(){
    Box{ this: BoxScope
        Box(Modifier.size(400.dp).background(Color.Black))
        Column{ this: ColumnScope
            Box(Modifier.size(100.dp).background(Color.Red))
            Box(Modifier.size(100.dp).background(Color.Green))
            Row{ this: RowScope
                Box(Modifier.size(100.dp).background(Color.Blue))
                Box(Modifier.size(100.dp).background(Color.Yellow))
            }
        }
    }
}
```



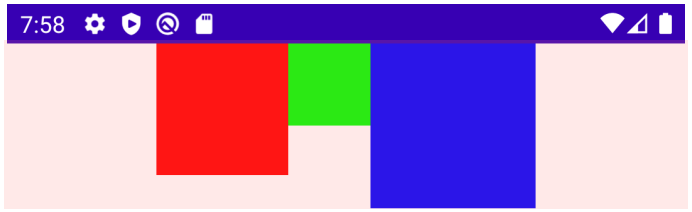
Row & Column

İçeriği Hizalama İşlemleri

#arangement
#alignment
#fillMax

Arrangement

Arrangement özelliği



- Bulunduğu tasarım alanın varsayılan hizalama özelliğidir.
- Row veya Column içinde satırsa yatayda, sütun ise dikeyde hizalama yapılır.
- Bunu sağlamak için row veya column içinde hareket alanı olmalıdır.

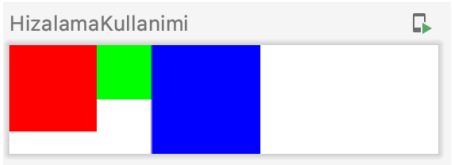
```
@Composable
fun HizalamaKullanimi(){
    Row(horizontalArrangement = Arrangement.Center,
        modifier = Modifier.fillMaxWidth(){ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Not : Modifier özellik olarak tek başına kullanılacaksa modifier = yazmaya gerek yoktur.

fillMax özelliği ile yatay veya dikeyde bulunduğu alan kadar yayılmasını sağlar.Match parent özelliğidir.Bu özellik olmadan hizalama için alan kalmaz bundan dolayı kullanmalıyız.

horizontalArrangement - Row

```
@Composable
fun HizalamaKullanimi(){
    Row(horizontalArrangement = Arrangement.Center,
        modifier = Modifier.fillMaxWidth(){ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```



Start varsayılan

End

Center

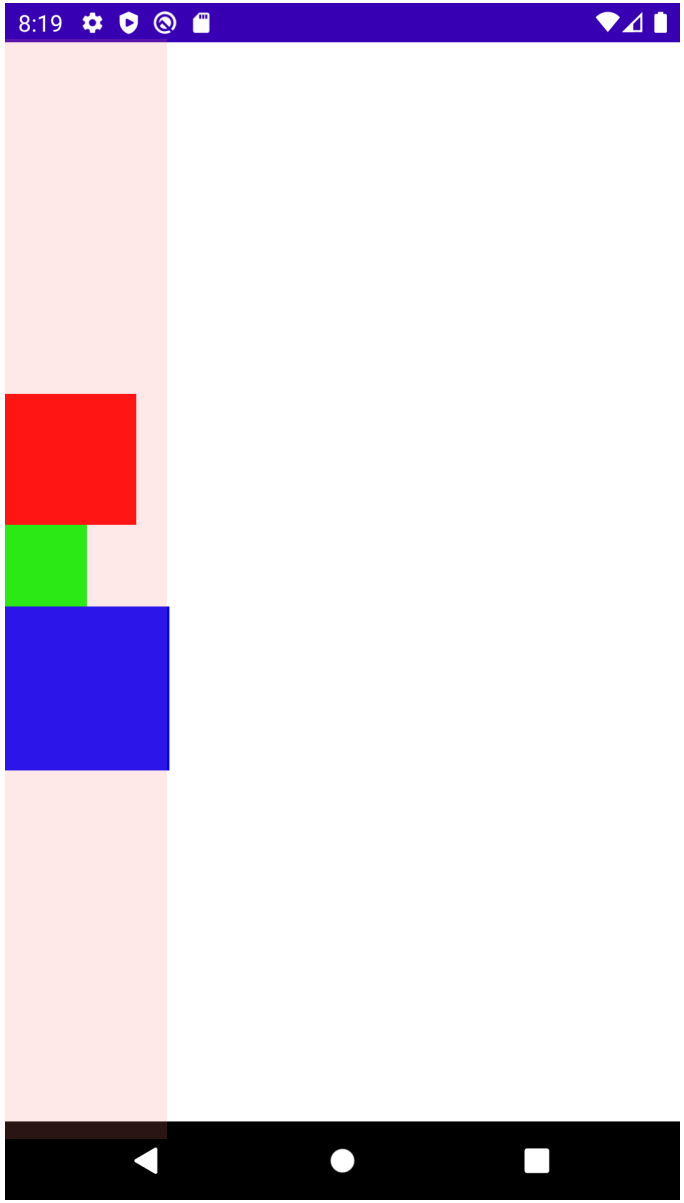
SpaceBetween

SpaceEvenly

verticalArrangement - Column

```
@Composable
fun HizalamaKullanimi(){
    Column(verticalArrangement = Arrangement.Center,
        modifier = Modifier.fillMaxHeight()){ this: ColumnScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

fillMax özelliği ile yatay veya dikeyde bulunduğu alan kadar yayılmasını sağlar.Match parent özelliğidir.Bu özellik olmadan hizalama için alan kalmaz bundan dolayı kullanmalıyız.

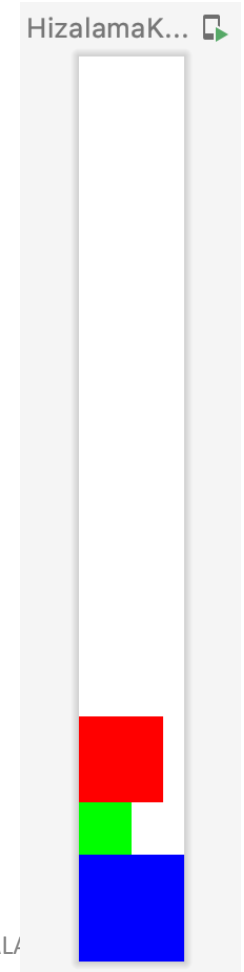


verticalArrangement - Column

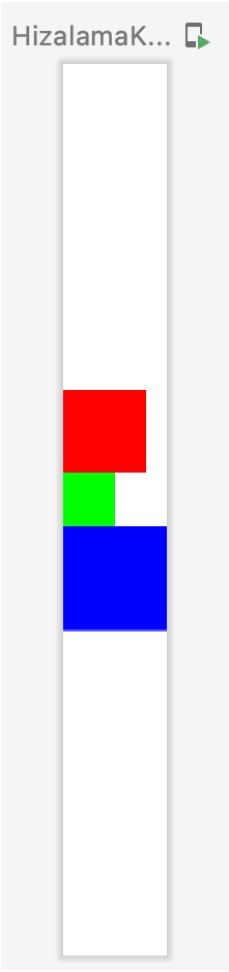
```
@Composable
fun HizalamaKullanimi(){
    Column(verticalArrangement = Arrangement.Center,
        modifier = Modifier.fillMaxHeight(){ this: ColumnScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Top
varsayılan

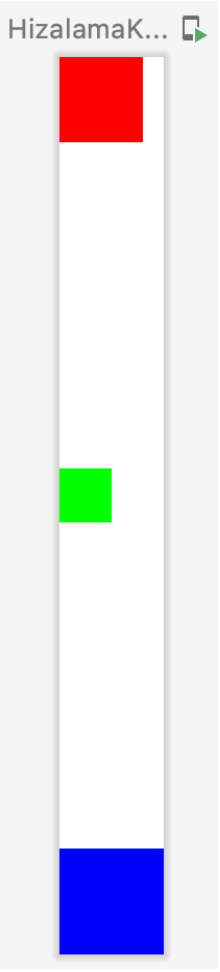
Bottom



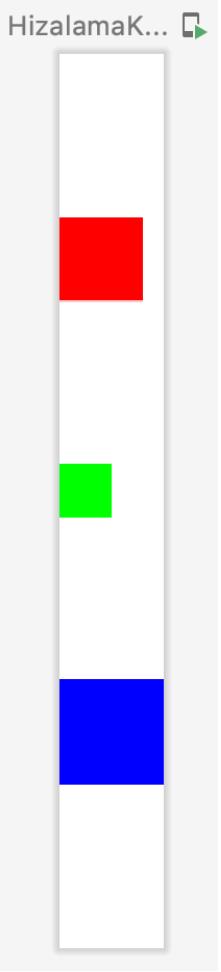
Center



SpaceBetween



SpaceEvenly



Alignment

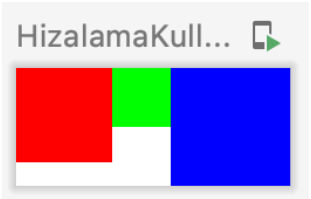
Alignment özelliği

- Row veya Column içinde Row ise dikeyde , column ise yatayda hizalama yapılır.

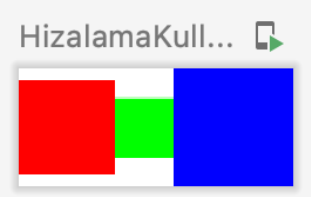
```
@Composable
fun HizalamaKullanimi(){
    Row(verticalAlignment = Alignment.CenterVertically){
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

verticalAlignment - Row

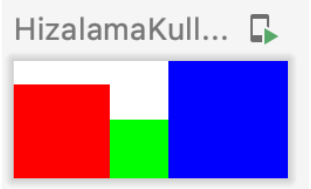
```
@Composable
fun HizalamaKullanimi(){
    Row(verticalAlignment = Alignment.CenterVertically){
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```



Top
varsayılan



CenterVertically



Bottom

horizontalAlignment - Column


```
@Composable
fun HizalamaKullanimi(){
    Column(horizontalAlignment = Alignment.CenterHorizontally){
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

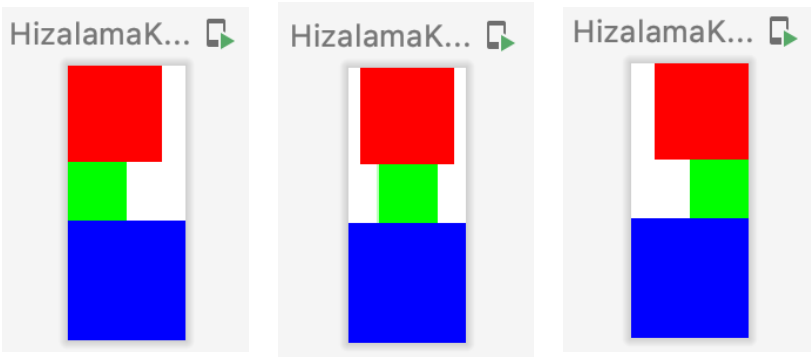
horizontalAlignment - Column

```
@Composable
fun HizalamaKullanimi(){
    Column(horizontalAlignment = Alignment.CenterHorizontally){
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Start varsayılan

CenterHorizontally

End

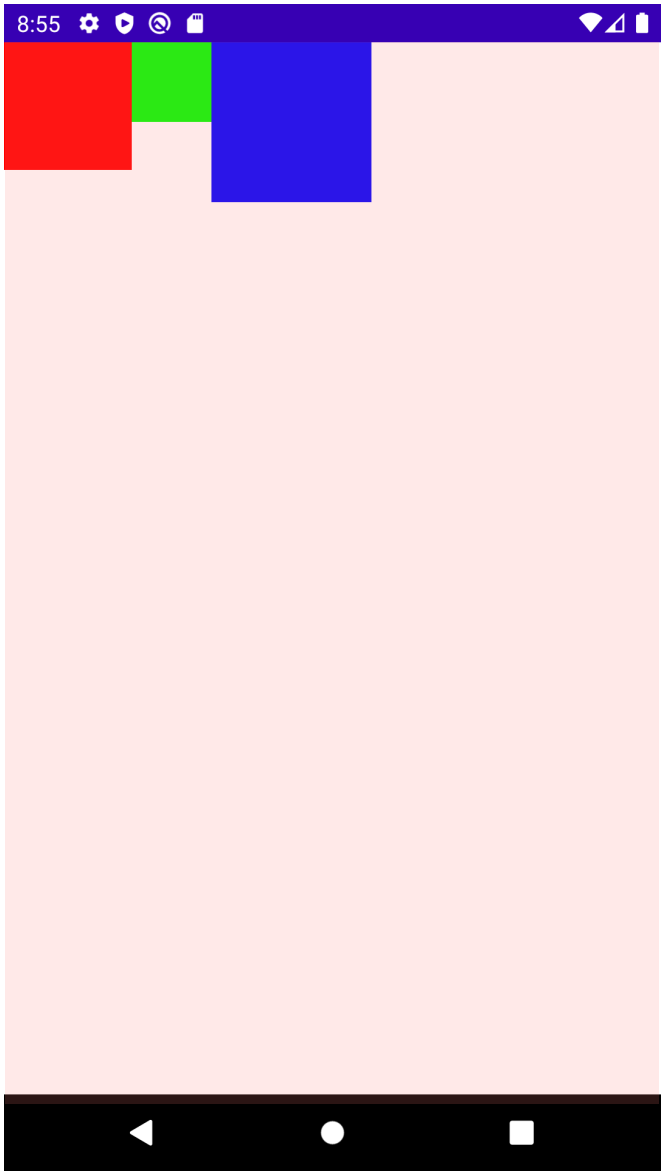


FillMax

fillMax

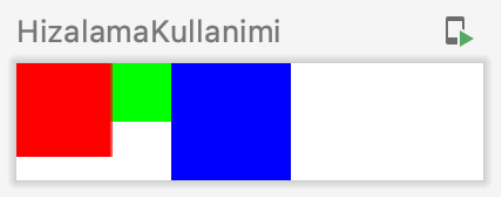
- Row veya Column yapısını bulunduğu alana göre yayar.
- Match parent özelliğidir.
- fillMaxSize() yatay ve dikeyde yayılma yapar.
- fillMaxWidth() yatayda yayılma yapar.
- fillMaxHeight() dikeyde yayılma yapar.

```
@Composable
fun HizalamaKullanimi(){
    Row(modifier = Modifier.fillMaxSize()){ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```



fillMax - Row

fillMaxWidth()



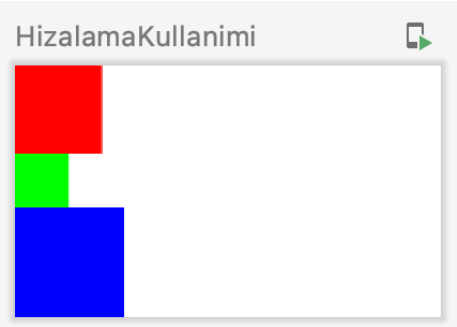
fillMaxHeight()

fillMaxSize()


```
@Composable
fun HizalamaKullanimi(){
    Row(modifier = Modifier.fillMaxSize()){ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

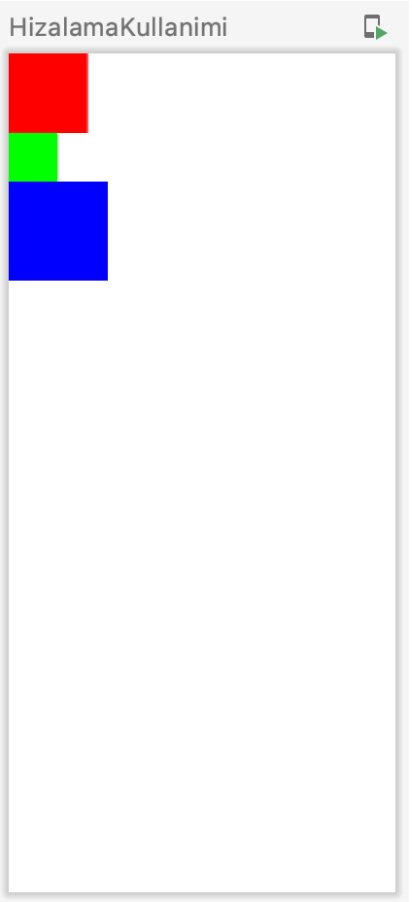
fillMax - Column

fillMaxWidth()



fillMaxHeight()

fillMaxSize()

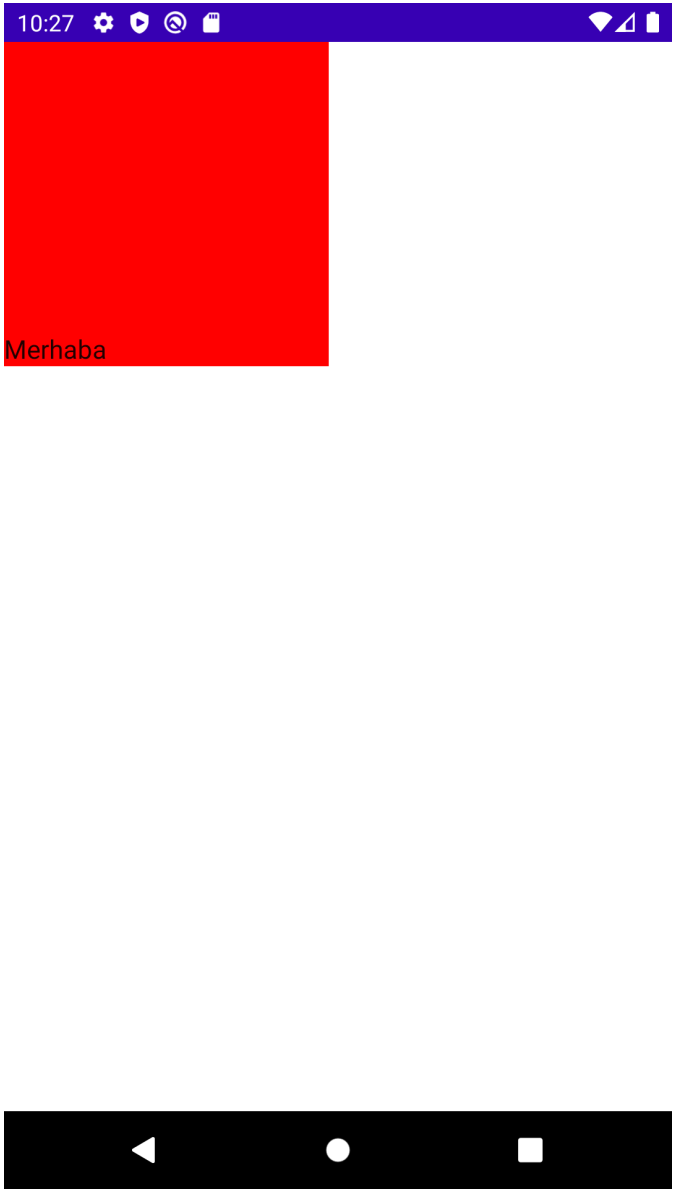



```
@Composable
fun HizalamaKullanimi(){
    Column(modifier = Modifier.fillMaxSize()){ this: ColumnS
        Box(Modifier.size(80.dp).background(Color.Red))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Align

Align

- İçinde bulunduğu alana göre kendini hizalama işlemi yapar.
- Eğer görsel nesne Box içindeyse 9 farklı noktaya hizalanabilir.
- Row ve Column ise 3 noktaya hizalanır.




```
@Composable
fun AlignKullanimi(){
    Box(Modifier.size(200.dp).background(Color.Red)){ this: BoxScope
        Text(text="Merhaba",
            modifier = Modifier.align(Alignment.BottomStart))
    }
}
```

Align - Box Örnekler

TopCenter

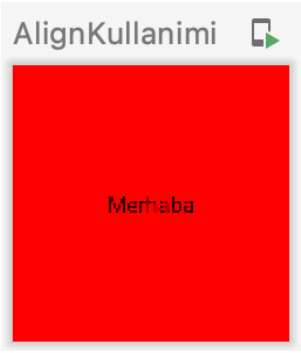
TopStart

TopEnd

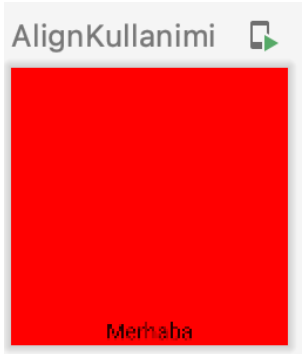
CenterEnd

CenterStart

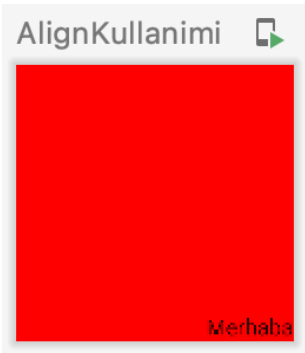
Center



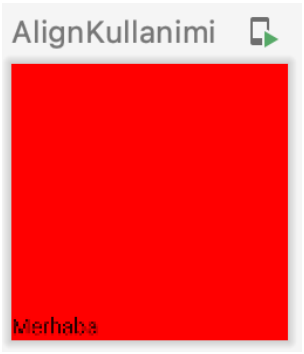
BottomCenter



BottomEnd



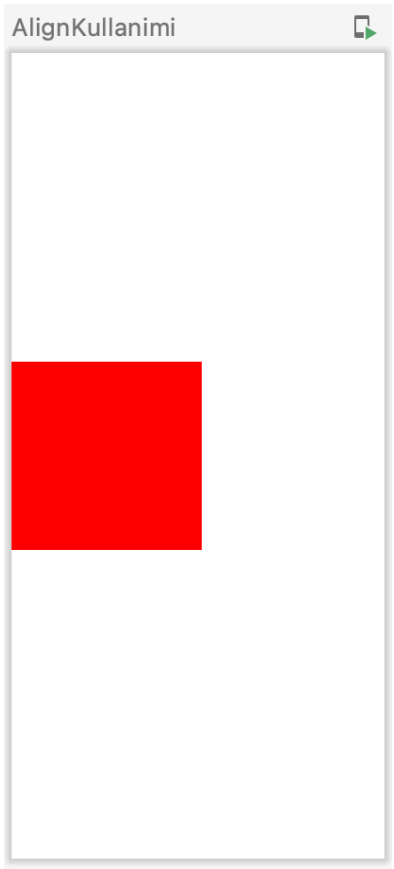
BottomStart



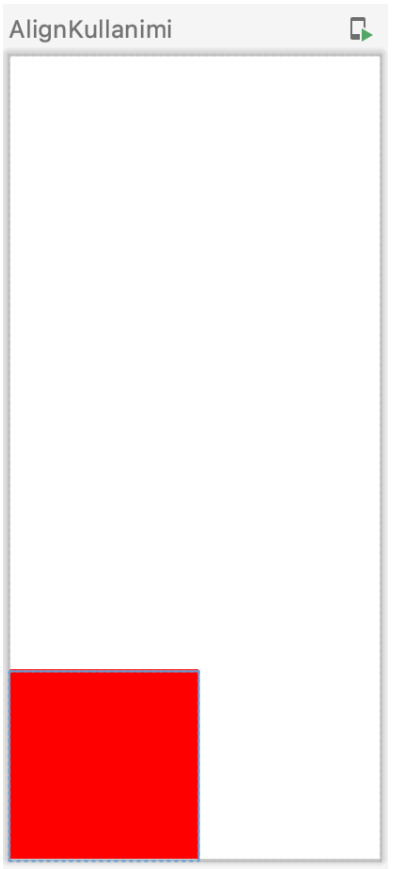
Align - Row Örnekler

```
@Composable
fun AlignKullanimi(){
    Row(modifier = Modifier.fillMaxSize()) {  this: Ro
        Box(Modifier
                .size(200.dp)
                .background(Color.Red)
                .align(Alignment.CenterVertically))
    }
}
```

CenterVertically



Bottom



Top

Align - Column Örnekler

```
@Composable
fun AlignKullanimi(){
    Column(modifier = Modifier.fillMaxSize()) {  this: C
        Box(Modifier
                .size(200.dp)
                .background(Color.Red)
                .align(Alignment.CenterHorizontally))
    }
}
```

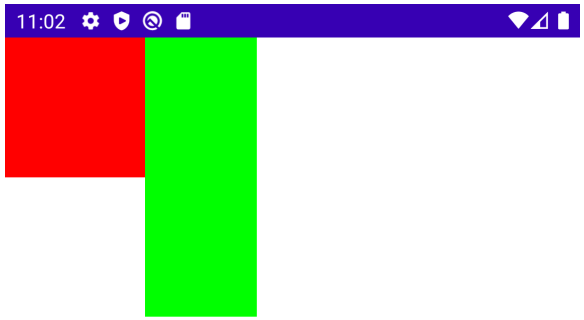
CenterHorizontally

Start

End

Size

Size : Boyutlandırma



- Herhangi bir görsel nesneyi boyutlandırabiliriz.

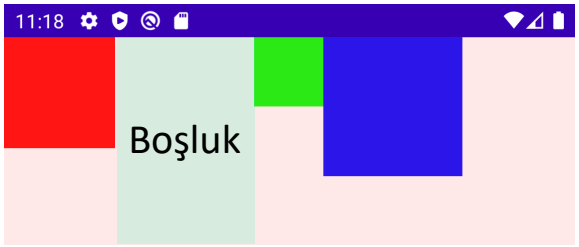
```
@Composable
fun SizeKullanimi(){
    Row{ this: RowScope
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(80.dp, 200.dp).background(Color.Green))
    }
}
```

Tek değer genişlik ve yükseklik için aynı kabul edilir.

Genişlik ve Yükseklik değerlerini ayrı ayrı verebiliriz.

Spacer

Spacer : Boyutlu Boşluk

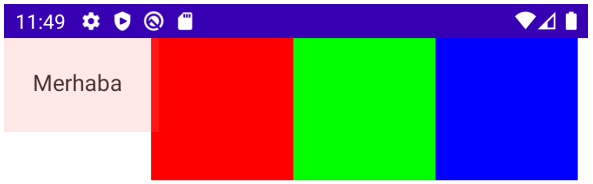


- Var olan boş alanı tam olarak kullanarak boşluk oluşturur.

```
@Composable
fun SpacerKullanimi(){
    Row(modifier = Modifier.fillMaxWidth()){ this: RowScope
        Box(Modifier.size(80.dp).background(Color.Red))
        Spacer(Modifier.size(100.dp,200.dp))
        Box(Modifier.size(50.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```


Padding

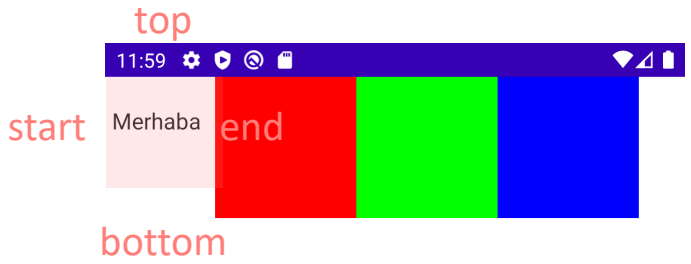
Padding



- Padding , widget etrafında boşluk oluşturmak için kullanılır.
- *all* özelliği bütün kenarlara aynı oranda boşluk verir.

```
@Composable
fun PaddingKullanimi(){
    Row{ this: RowScope
        Text(text = "Merhaba", modifier = Modifier.padding(all = 20.dp))
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(100.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Padding

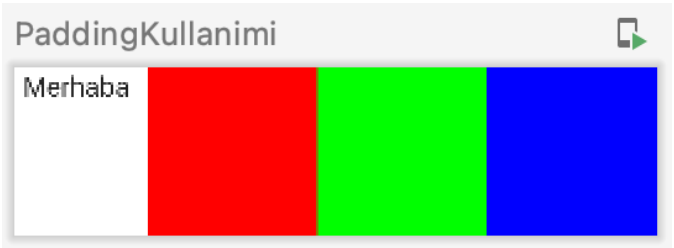


- start,end,top,bottom şeklinde etrafına farklı oranlarda boşluk verebiliriz.

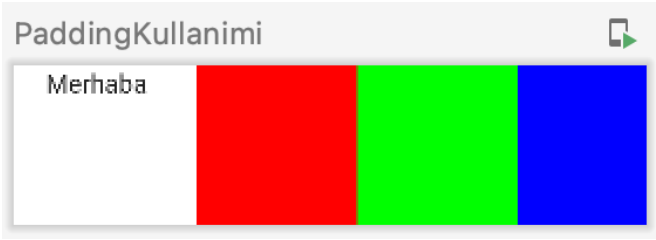
```
@Composable
fun PaddingKullanimi(){
    Row{ this: RowScope
        Text(text = "Merhaba", modifier = Modifier
            .padding(start = 5.dp,end = 10.dp,top = 20.dp,bottom = 30.dp))
        Box(Modifier.size(100.dp).background(Color.Red))
        Box(Modifier.size(100.dp).background(Color.Green))
        Box(Modifier.size(100.dp).background(Color.Blue))
    }
}
```

Padding

```
Row{  this: RowScope
    Text(text = "Merhaba", modifier = Modifier
        .padding(start = 5.dp,end = 10.dp))
    Box(Modifier.size(100.dp).background(Color.Red))
    Box(Modifier.size(100.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```

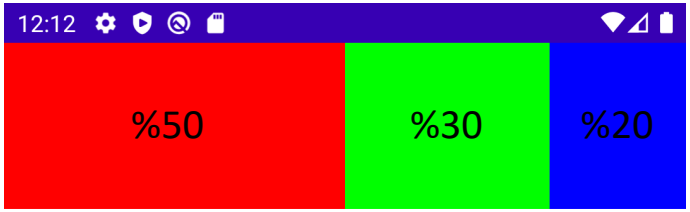



```
Row{  this: RowScope
    Text(text = "Merhaba", modifier = Modifier
        .padding(top = 20.dp,bottom = 30.dp))
    Box(Modifier.size(100.dp).background(Color.Red))
    Box(Modifier.size(100.dp).background(Color.Green))
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```



Weight

Weight



- Görsel nesne bulunduğu alana göre oranlı şekilde yayılabilir.

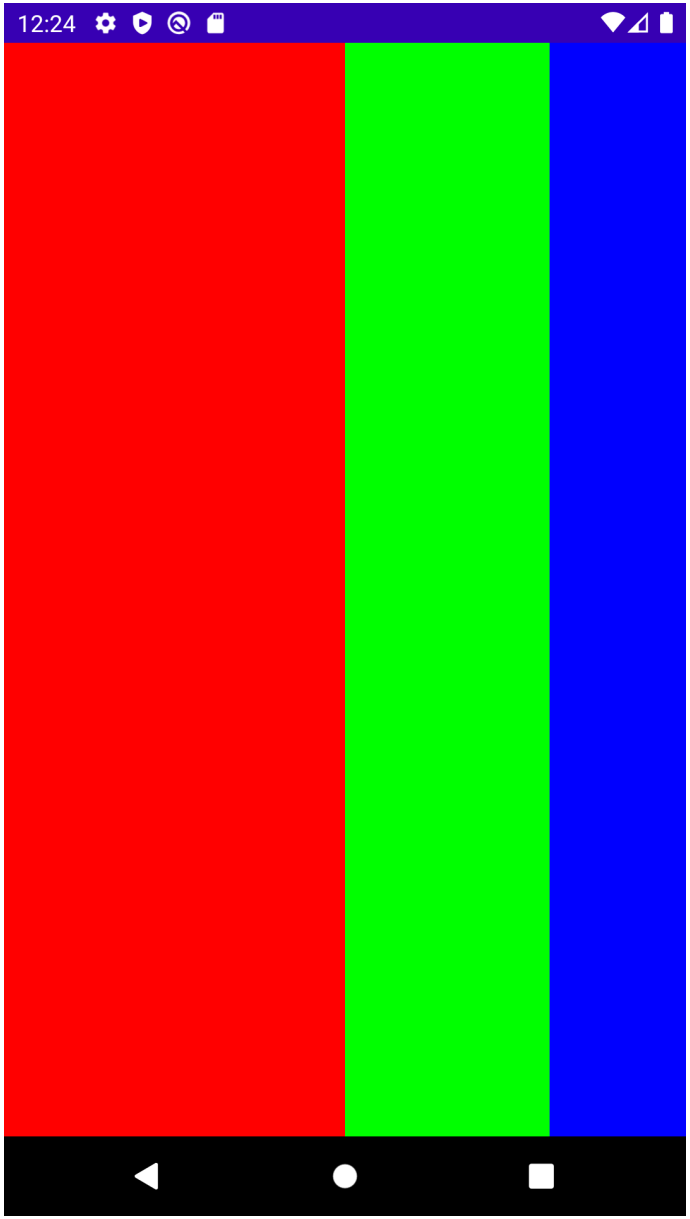
```
@Composable
fun OranlamaKullanimi(){
    Row(modifier = Modifier.fillMaxWidth()){
        Box(Modifier.size(100.dp)
                .background(Color.Red)
                .weight( weight: 50f))
        Box(Modifier.size(100.dp)
                .background(Color.Green)
                .weight( weight: 30f))
        Box(Modifier.size(100.dp)
                .background(Color.Blue)
                .weight( weight: 20f))
    }
}
```

Satır genişliğini bulunduğu alana göre yayılmasını sağladık. weight özelliği ile bu alanın tamamı 100 olarak temsil edilip oranladık.

Weight - Örnek

- Görsel nesne bulunduğu alana göre oranlı şekilde yayılabilir.

```
@Composable
fun OranlamaKullanimi(){
    Row(modifier = Modifier.fillMaxSize()){
        Box(Modifier.fillMaxHeight()
                .background(Color.Red)
                .weight( weight: 50f))
        Box(Modifier.fillMaxHeight()
                .background(Color.Green)
                .weight( weight: 30f))
        Box(Modifier.fillMaxHeight()
                .background(Color.Blue)
                .weight( weight: 20f))
    }
}
```



Özelleştirilmiş Widget Oluşturma

Özelleştirilmiş Widget Oluşturma

- Sık kullandığınız tasarımları modüler olarak kullanmak için ayrı bir widget gibi oluşturup ismi ile erişim sağlayabiliriz.

```
@Composable
fun MaviKare(){
    Box(Modifier.size(100.dp).background(Color.Blue))
}

@Composable
fun KirmiziKare(){
    Box(Modifier.size(100.dp).background(Color.Red))
}

@Composable
fun Yazi(icerik:String,yaziBoyutu:Int){
    Text(text = icerik,fontSize = yaziBoyutu.sp)
}
```

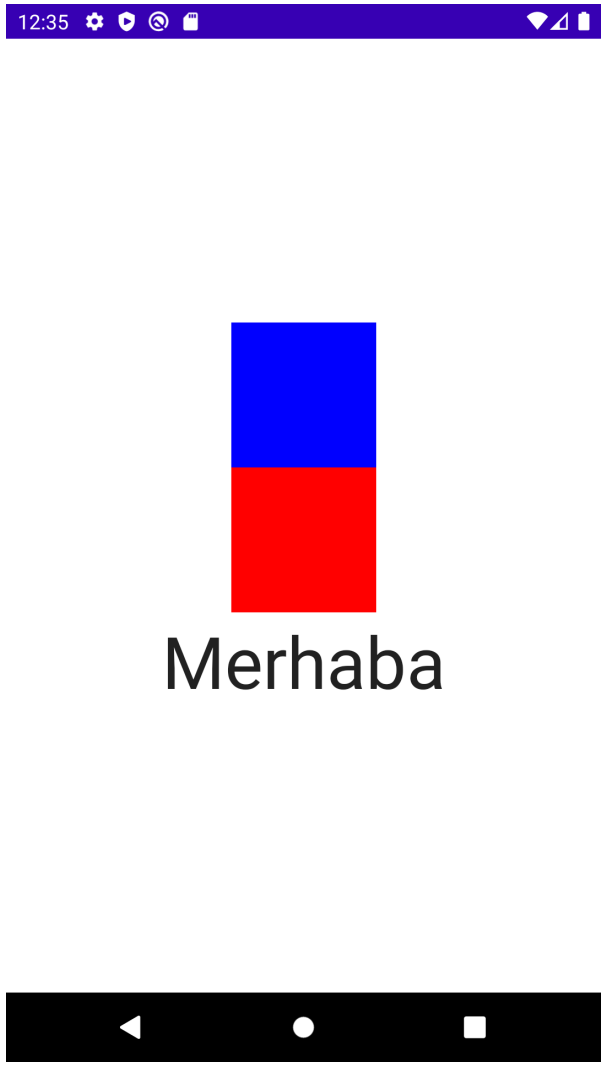
Özelleştirilmiş Widget Oluşturma

```
@Composable
fun OzellestirilmisWidgetKullanimi(){
    Column(verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()){ this: ColumnScope
        MaviKare()
        KirmiziKare()
        Yazi(icerik = "Merhaba", yaziBoyutu = 50)
    }
}
```

```
@Composable
fun MaviKare(){
    Box(Modifier.size(100.dp).background(Color.Blue))
}
```

```
@Composable
fun KirmiziKare(){
    Box(Modifier.size(100.dp).background(Color.Red))
}
```

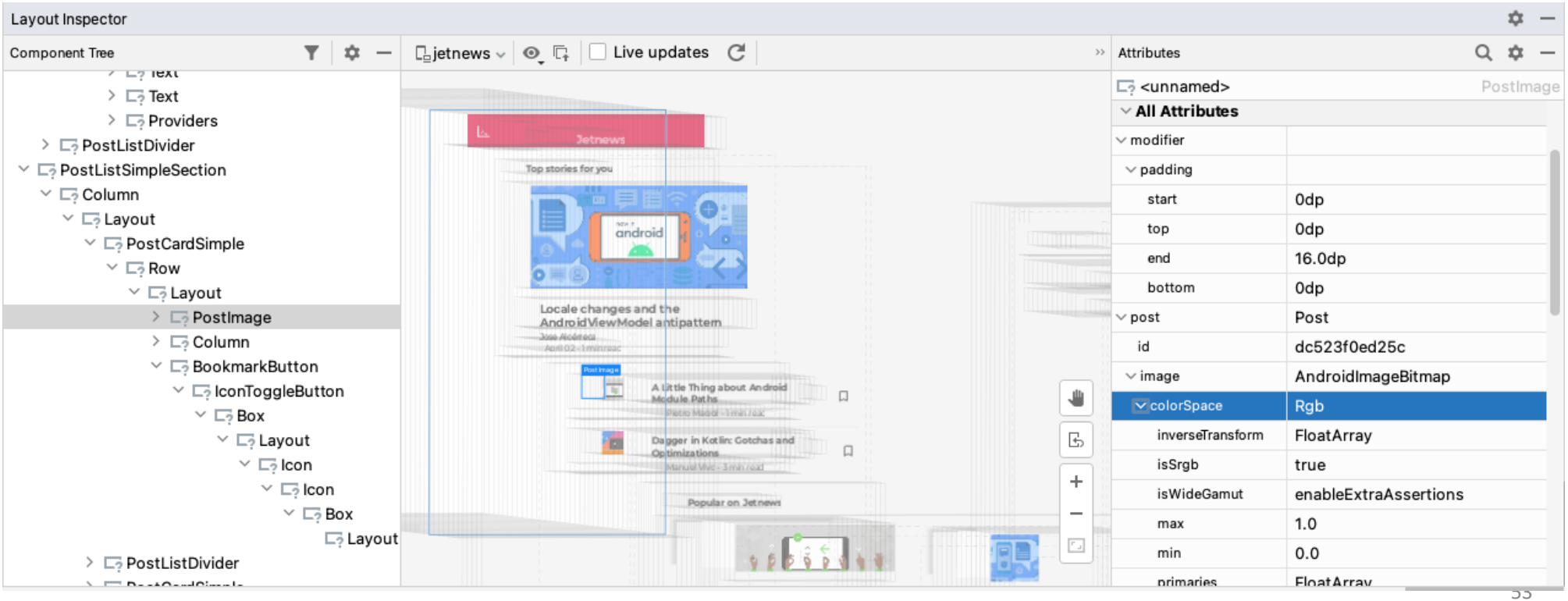
```
@Composable
fun Yazi(icerik:String,yaziBoyutu:Int){
    Text(text = icerik,fontSize = yaziBoyutu.sp)
}
```



Layout Inspector

Layout Inspector

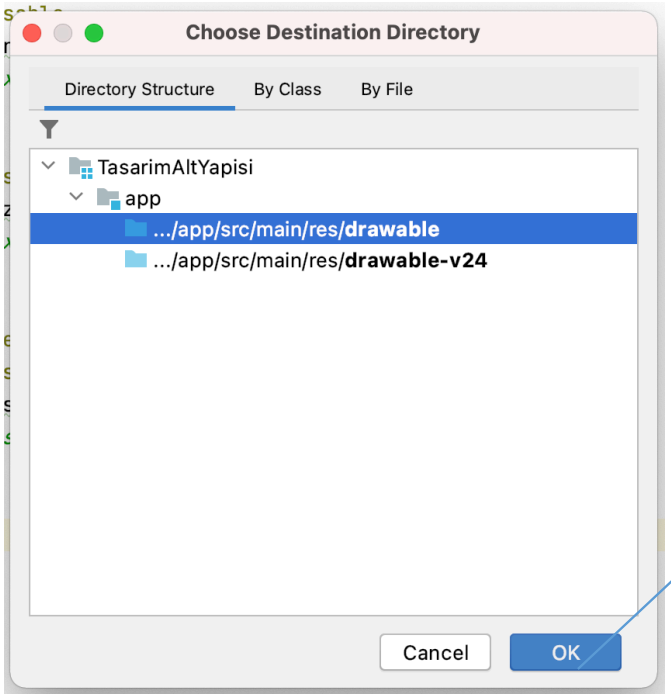
- Görsel olarak kullanılan widgetları ve katmanları görebiliyoruz

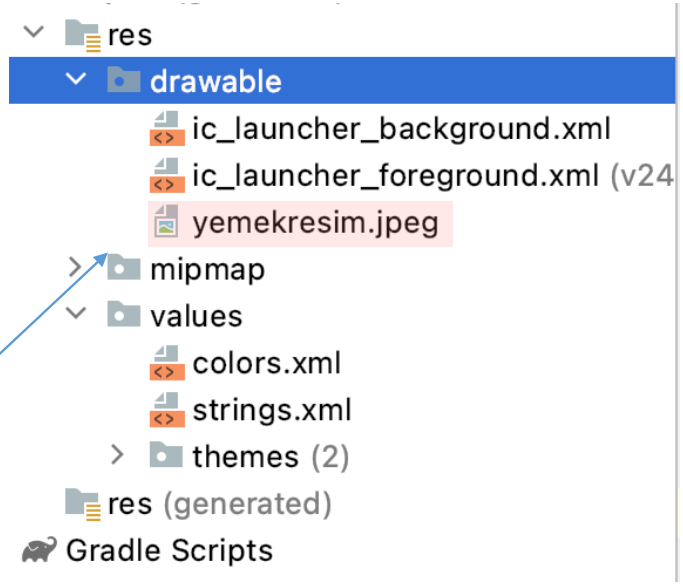


Projeye Resim Ekleme

Dışardan Resim Ekleme

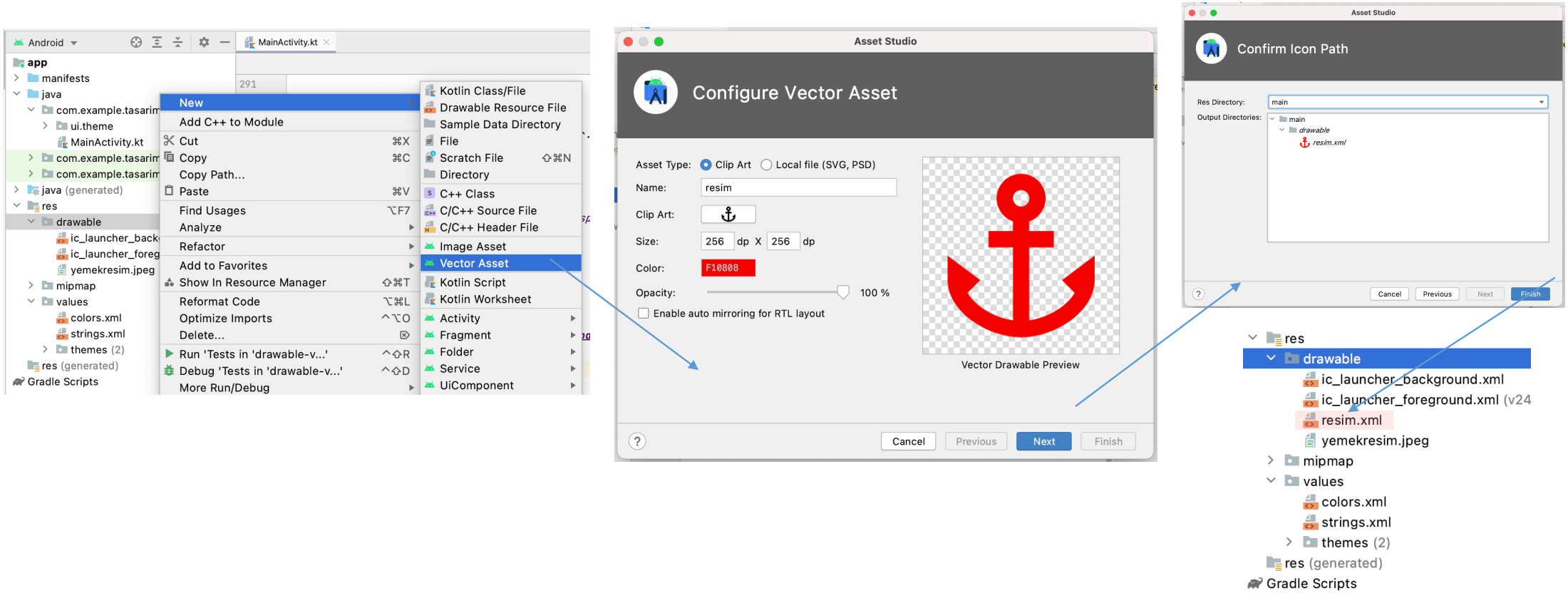
- Android Studio'ya eklemek istediğiniz resmi kopyalayıp drawable klasörüne yapıştırabilirsiniz.





Android Studio Resimlerini Kullanma

- Google’ın bizlere sunduğu lisanslı resimleri kullanabiliriz.



Resim kullanımı

```
@Composable
fun ResimKullanimi(){
    Column{ this: ColumnScope
        Image(painter = painterResource(id = R.drawable.yemekresim)
            , contentDescription: "açıklama : yemek resim")
        Image(painter = painterResource(id = R.drawable.resim)
            , contentDescription: "açıklama : vector resim")
    }
}
```

Çoklu Dil Desteği

Çoklu dil desteği

- Oluşturacağımız uygulamalar uluslar arası olabilir ve birçok ülke için dil desteği sunmalıyız.
- Android otomatik olarak cihaz hangi dilde ise ona göre bu dosyaları çalıştırmaktadır.
- Genelde varsayılan dil ingilizce yapılır.
- Çoklu dil desteği sağlamak için String dosyasının farklı diller için olan türlerini oluşturmalıyız.
- Arayüzde kullanılan metinlerin String dosyasından çekilmesi gereklidir.

```
res/
    values/
        strings.xml
    values-es/
        strings.xml
    values-fr/
        strings.xml
```

English (default locale), `/values/strings.xml`:

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">My Application</string>
    <string name="hello_world">Hello World!</string>
</resources>
```

Spanish, `/values-es/strings.xml`:

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">Mi Aplicación</string>
    <string name="hello_world">Hola Mundo!</string>
</resources>
```

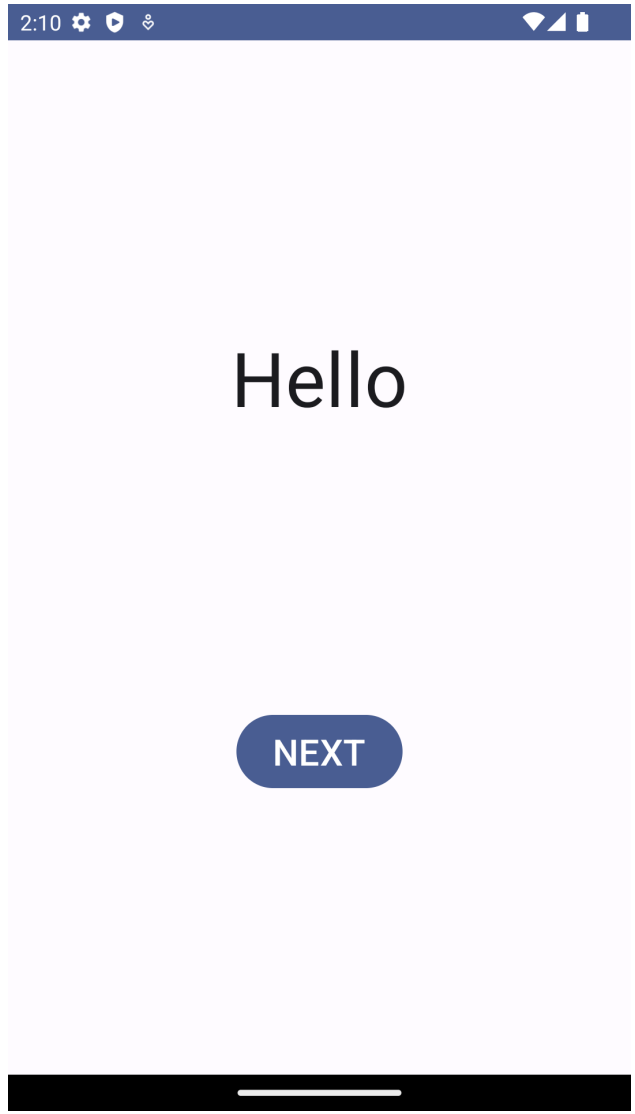
French, `/values-fr/strings.xml`:

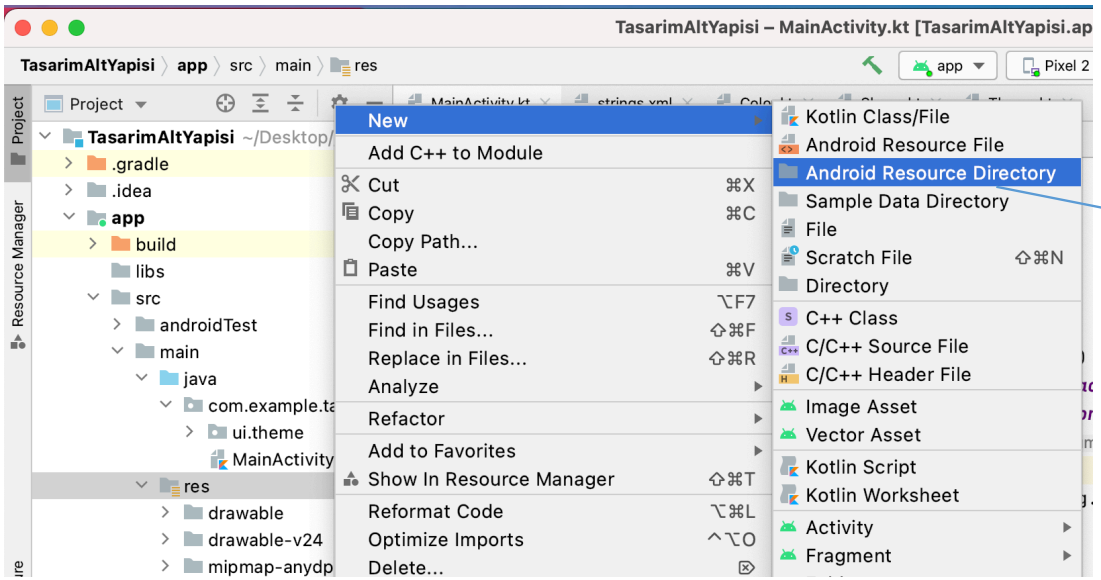
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">Mon Application</string>
    <string name="hello_world">Bonjour le monde !</string>
</resources>
```

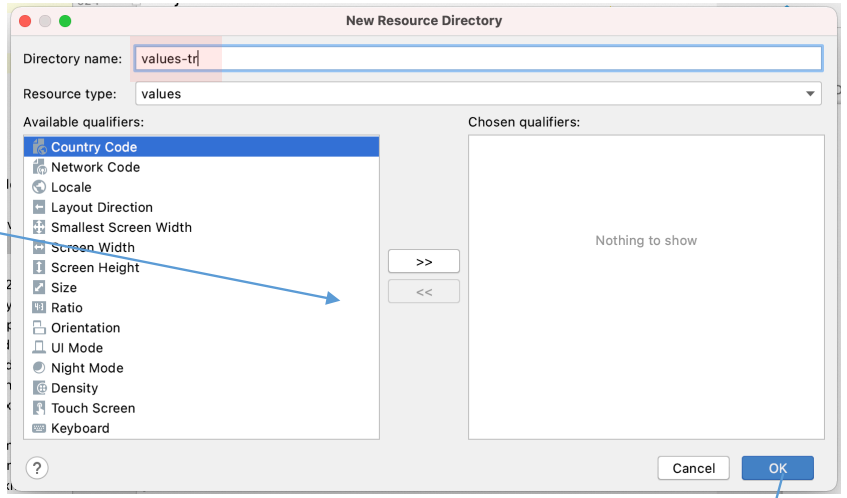
```
@Composable
fun CokluDilDestegiKullanimi(){
    Column(verticalArrangement = Arrangement.SpaceEvenly,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()){ this: ColumnScope

        Text(text = stringResource(id = R.string.textYazi)
            ,fontSize = 50.sp)

        Button(onClick = {}) { this: RowScope
            Text(text = stringResource(id = R.string.buttonYazi)
                ,fontSize = 24.sp)
        }
    }
}
```

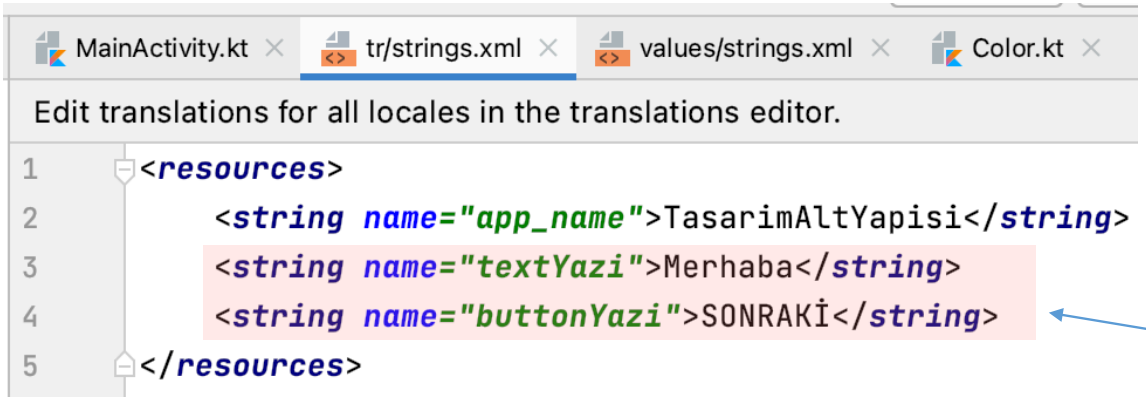
strings.xml

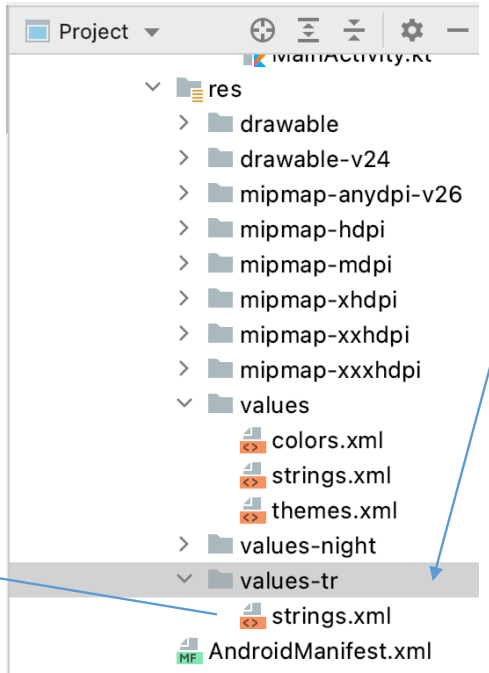
```
<resources>
    <string name="app_name">TasarimAltYapisi</string>
    <string name="textYazi">Hello</string>
    <string name="buttonYazi">NEXT</string>
</resources>
```




TasarimAltYapisi – MainActivity.kt [TasarimAltYapisi.ap
TasarimAltYapisi 〉 app 〉 src 〉 main 〉 res
app
Pixel 2
Project
TasarimAltYapisi ~/Desktop/
.gradle
.idea
app
build
libs
src
androidTest
main
java
com.example.ta
ui.theme
MainActivity
res
drawable
drawable-v24
mipmap-anydp
Resource Manager
New
Add C++ to Module
Cut ⌘X
Copy ⌘C
Copy Path...
Paste ⌘V
Find Usages ⌥F7
Find in Files... ⇧⌘F
Replace in Files... ⇧⌘R
Analyze
Refactor
Add to Favorites
Show In Resource Manager ⇧⌘T
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Delete...
Kotlin Class/File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File ⇧⌘N
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment


New Resource Directory
Directory name: values-tr
Resource type: values
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Density
Touch Screen
Keyboard
>>
<<
Chosen qualifiers:
Nothing to show
Cancel
OK

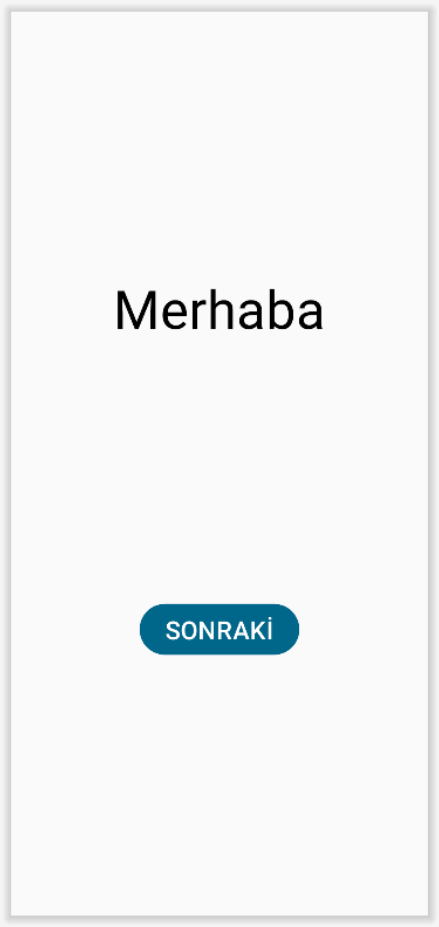

MainActivity.kt
tr/strings.xml
values/strings.xml
Color.kt
Edit translations for all locales in the translations editor.
1 <resources>
2 <string name="app_name">TasarimAltYapisi</string>
3 <string name="textYazi">Merhaba</string>
4 <string name="buttonYazi">SONRAKİ</string>
5 </resources>


Project
res
drawable
drawable-v24
mipmap-anydpi-v26
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
values
colors.xml
strings.xml
themes.xml
values-night
values-tr
strings.xml
AndroidManifest.xml

Test

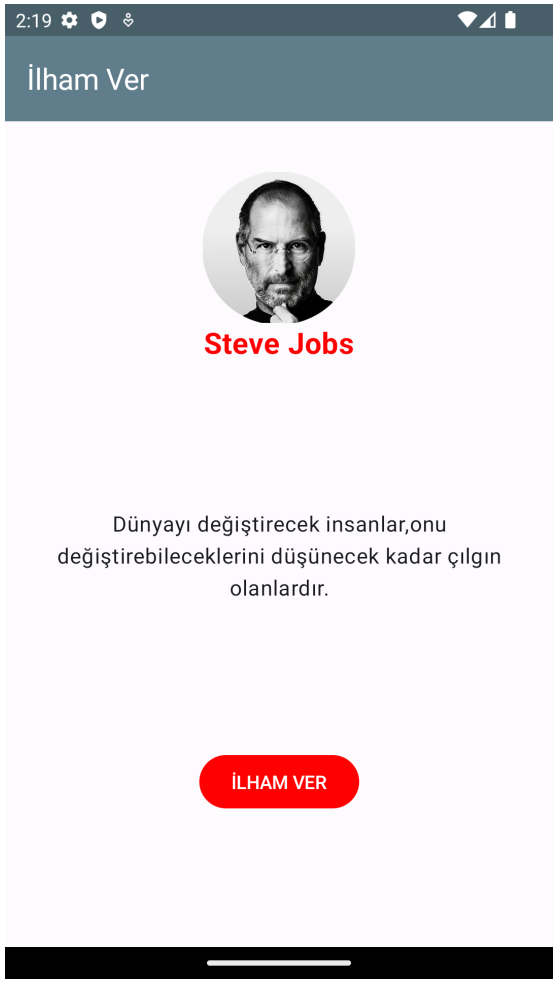
- Preview locale özelliği ile hangi dil arayüzünü görmek istiyorsak yazıp test edebiliriz.

```
@Preview(showBackground = true,locale = "tr")
@Composable
fun DefaultPreview() {
    TasarimAltYapisiTheme {
        CokluDilDestegiKullanimi()
    }
}
```



Tasarım Uygulamaları

UYGULAMA – İLHAM VER



Status bar rengi değişimi

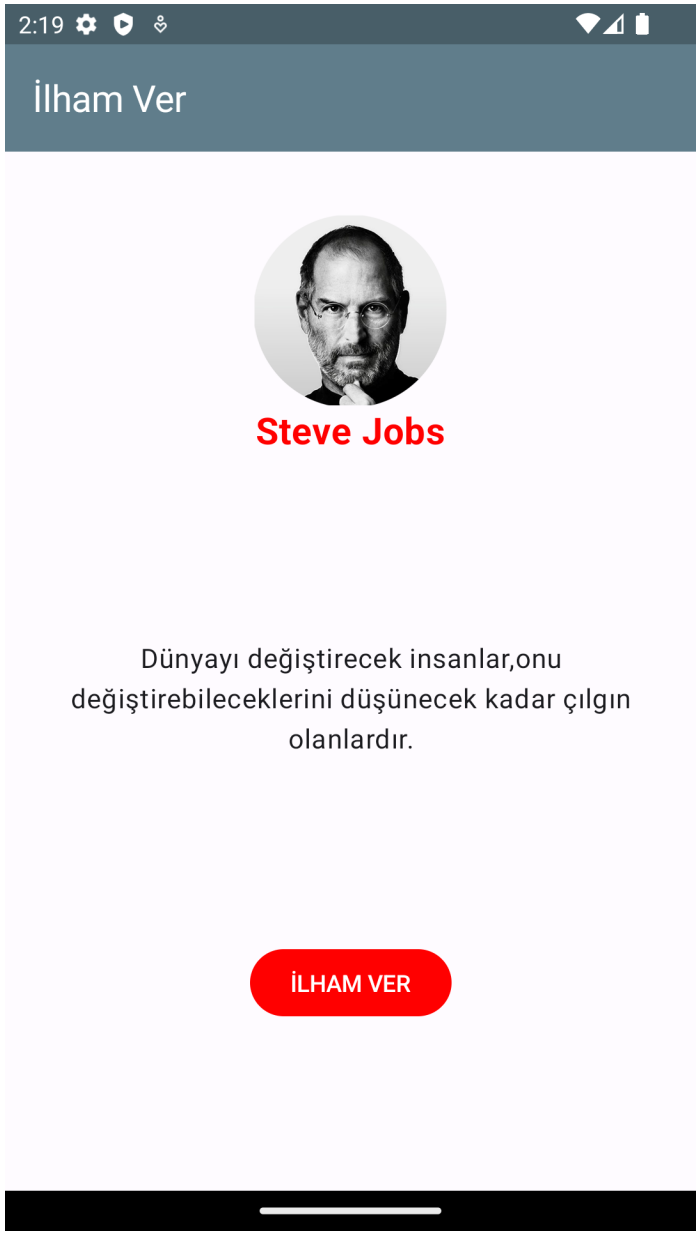
- Uygulama durum barının rengini değiştirmeliyiz.

Renk Tanımlama *colors.xml*

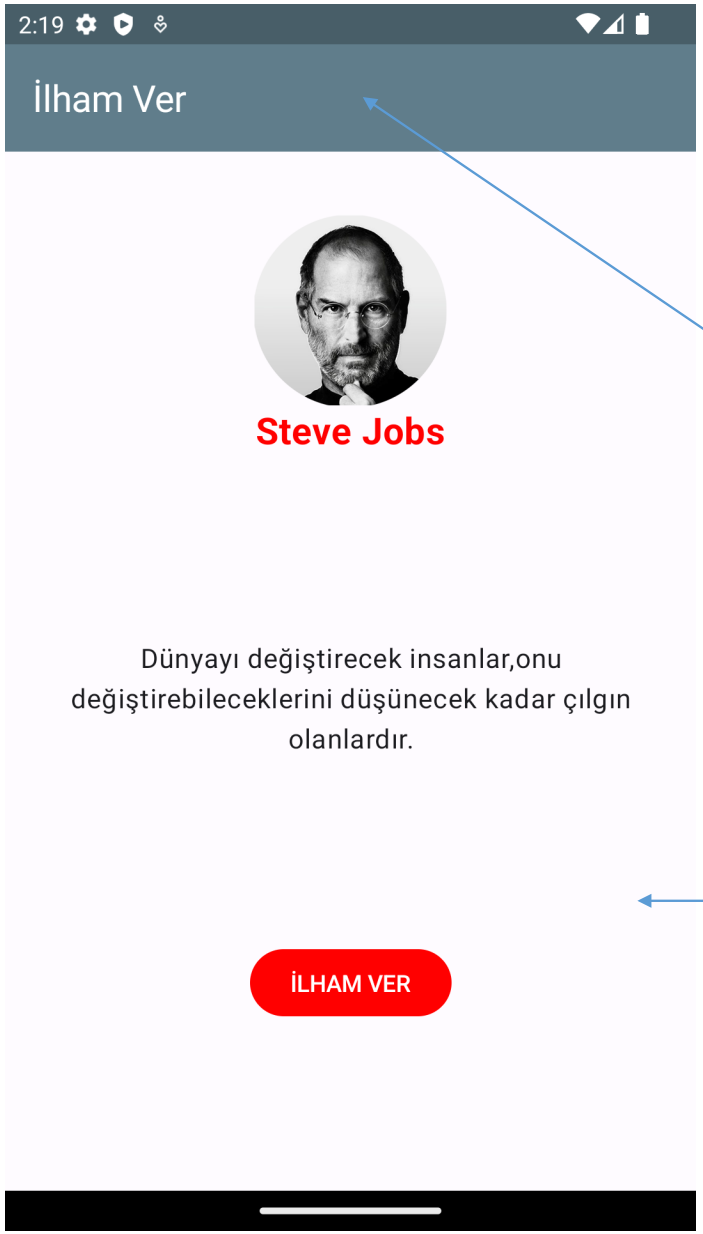
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="anaRenk">#607D8B</color>
    <color name="anaRenkKoyu">#485E68</color>
</resources>
```

Status Bar Renk Değişimi *Theme.kt*

```
val view = LocalView.current

val statusBarRenk = colorResource(id = R.color.anaRenkKoyu).toArgb()

if (!view.isInEditMode) {
    SideEffect {
        val window = (view.context as Activity).window
        window.statusBarColor = statusBarRenk
        WindowCompat.getInsetsController(window, view).isAppearanceLightStatusBars = darkTheme
    }
}
```

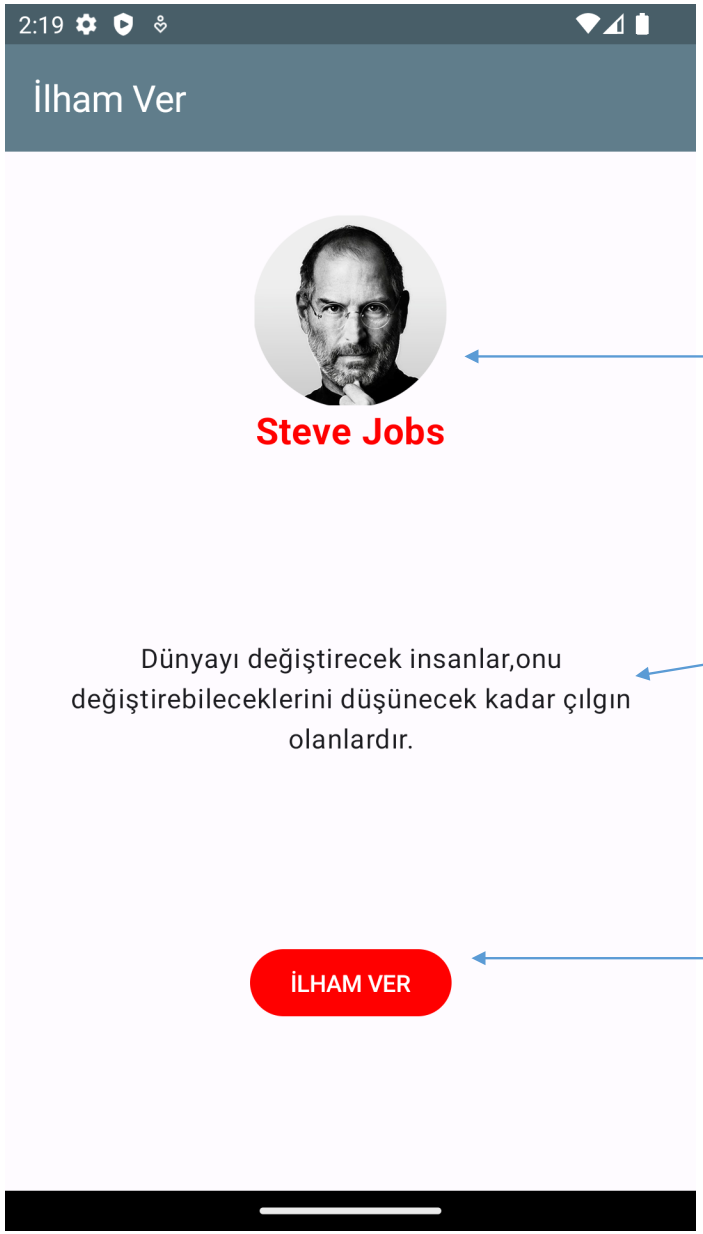



```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            IlhamVerTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Sayfa()
                }
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    IlhamVerTheme {
        Sayfa()
    }
}
```



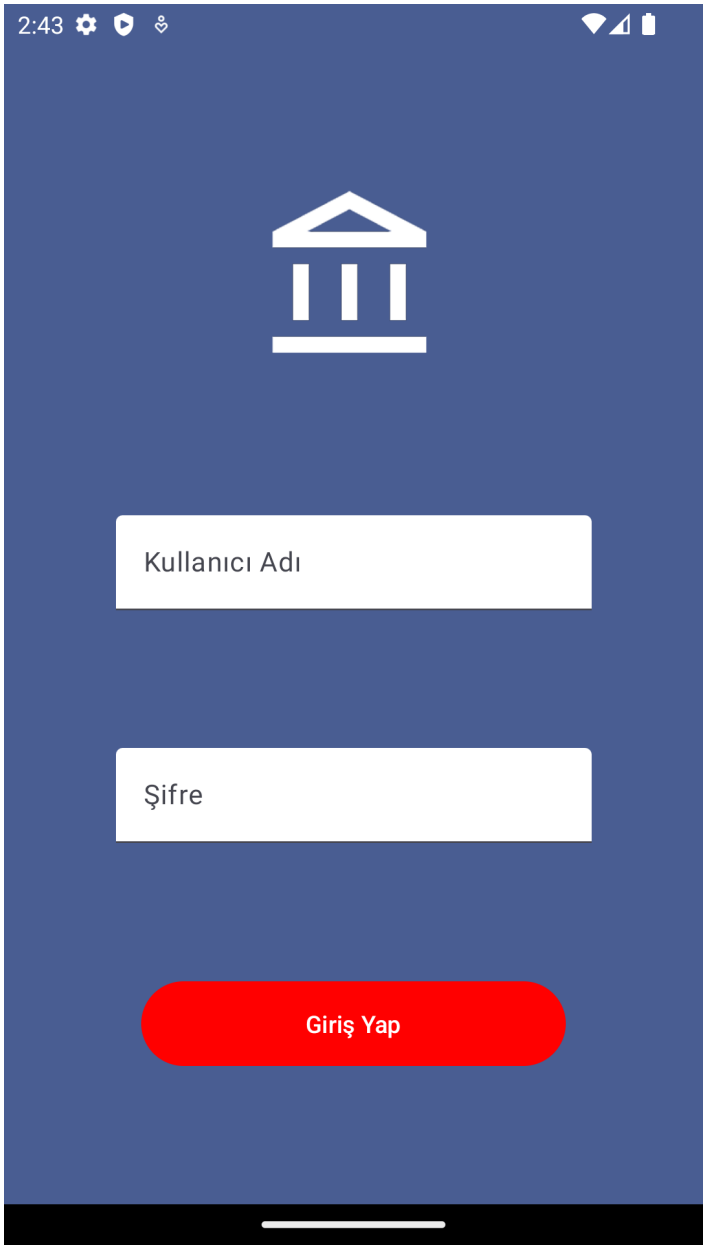

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text(text = "İlham Ver")},
                colors = TopAppBarDefaults.smallTopAppBarColors(
                    containerColor = colorResource(id = R.color.anaRenk),
                    titleContentColor = colorResource(id = R.color.white),
                )
            )
        },
        content = { it: PaddingValues
            Column(
                modifier = Modifier.fillMaxSize(),
                verticalArrangement = Arrangement.SpaceEvenly,
                horizontalAlignment = Alignment.CenterHorizontally
```

Sayfanın ana tasarım alanı içeriği

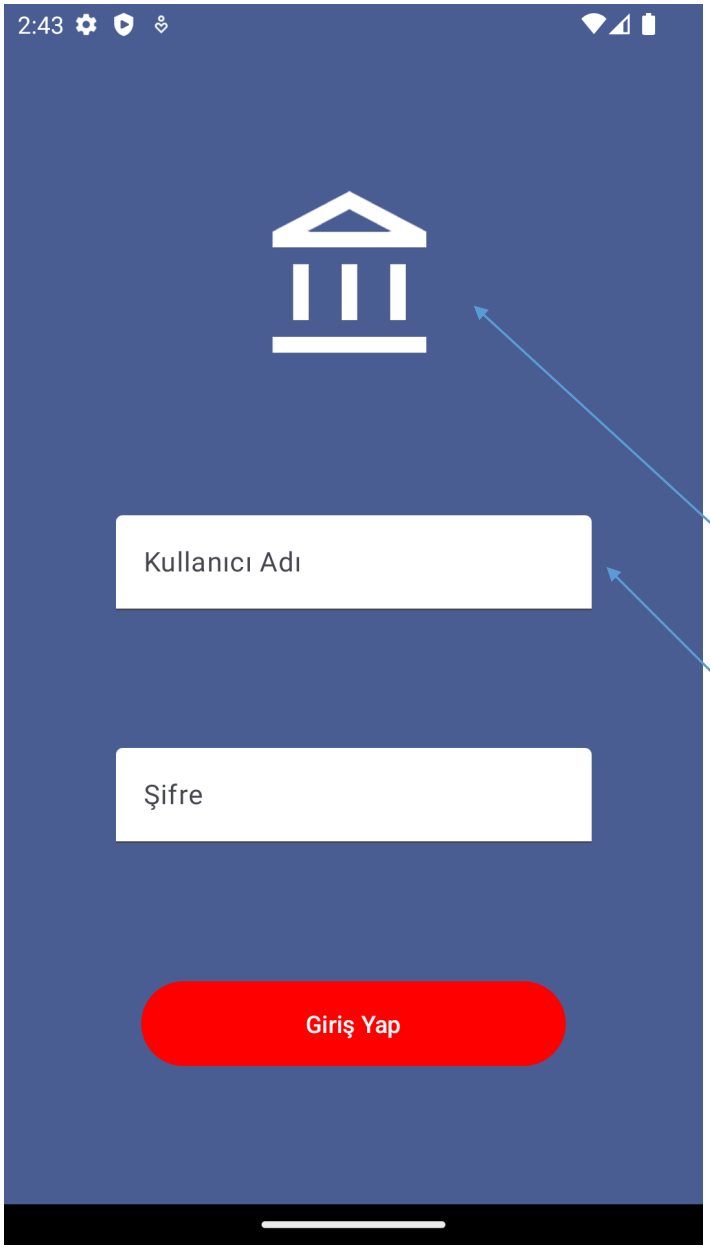



```
content = { it: PaddingValues
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.SpaceEvenly,
        horizontalAlignment = Alignment.CenterHorizontally
    ) { this: ColumnScope
        Column(horizontalAlignment = Alignment.CenterHorizontally) { this: ColumnScope
            Image(painter = painterResource(id = R.drawable.stevejobs),
                contentDescription = "")
            Text(text = "Steve Jobs", color = Color.Red, fontWeight = FontWeight.Bold,fontSize = 22.sp)
        }

        Text(
            text = "Dünyayı değiştirecek insanlar,onu değiştirebileceklerini düşünecek kadar çılgın olanlardır.",
            modifier = Modifier.padding(all = 10.dp),
            textAlign = TextAlign.Center
        )

        Button(onClick = { Log.e( tag: "Button", msg: "İlham Verildi") },
            colors = ButtonDefaults.buttonColors(
                containerColor = Color.Red,
                contentColor = Color.White
            )
        ) { this: RowScope
            Text(text = "İLHAM VER")
        }
    }
}
```

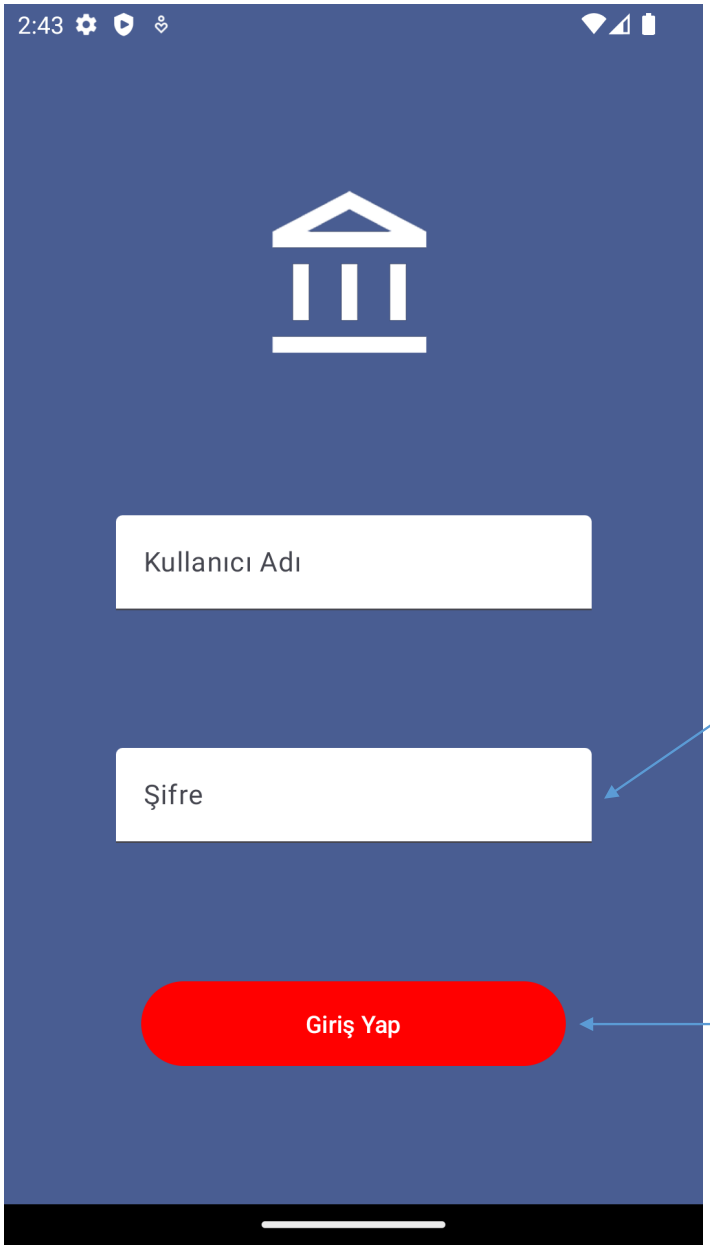
UYGULAMA – LOG IN




```kotlin
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            LoginUygulamaTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Sayfa()
                }
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    LoginUygulamaTheme {
        Sayfa()
    }
}
```




```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    //Alt surface sadece bu sayfada renk değişimi uygular.
    Surface(color = MaterialTheme.colorScheme.primary) {

        val tfKullaniciAdi = remember { mutableStateOf( value: "") }
        val tfSifre = remember { mutableStateOf( value: "") }

        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.SpaceEvenly,
            horizontalAlignment = Alignment.CenterHorizontally
        ) { this: ColumnScope
            Image(painter = painterResource(id = R.drawable.logo), contentDescription = "")
            TextField(
                value = tfKullaniciAdi.value,
                onValueChange = {tfKullaniciAdi.value = it},
                label = { Text(text = "Kullanıcı Adı")},
                colors = TextFieldDefaults.textFieldColors(
                    containerColor = Color.White,
                    textColor = Color.Black,
                    focusedIndicatorColor = Color.Red
                )
            )
```




```
TextField(
    value = tfSifre.value,
    onValueChange = {tfSifre.value = it},
    label = { Text(text = "Şifre")},
    colors = TextFieldDefaults.textFieldColors(
        containerColor = Color.White,
        textColor = Color.Black,
        focusedIndicatorColor = Color.Red
    )
)

Button(
    onClick = { Log.e( tag: "Button", msg: "Giriş Yapıldı") },
    colors = ButtonDefaults.buttonColors(
        containerColor = Color.Red,
        contentColor = Color.White
    ),
    modifier = Modifier.size(250.dp,50.dp)
) { this: RowScope
    Text(text = "Giriş Yap")
}
```

UYGULAMA – YEMEK TARİFİ

Status bar rengi değişimi

- Uygulama durum barının rengini değiştirmeliyiz.

Renk Tanımlama *colors.xml*

```xml
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="anaRenk">#FF9801</color>
    <color name="anaRenkKoyu">#BF7100</color>
    <color name="alternatifRenk">#FF6E40</color>
</resources>
```

Status Bar Renk Değişimi *Theme.kt*

```
val view = LocalView.current

val statusBarRenk = colorResource(id = R.color.anaRenkKoyu).toArgb()

if (!view.isInEditMode) {
    SideEffect {
        val window = (view.context as Activity).window
        window.statusBarColor = statusBarRenk
        WindowCompat.getInsetsController(window, view).isAppearanceLightStatusBars = darkTheme
    }
}
```

Özel Text Oluşturma

```
@Composable
fun Yazi(icerik:String){
    Text(text = icerik)
}
```


```kotlin
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            YemekTarifiTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Sayfa()
                }
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    YemekTarifiTheme {
        Sayfa()
    }
}
```


```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Sayfa() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Yazi(icerik = "Yemek Tarifi")},
                colors = TopAppBarDefaults.smallTopAppBarColors(
                    containerColor = colorResource(id = R.color.anaRenk),
                )
            )
        },
        content = { it: PaddingValues
            Column(
                horizontalAlignment = Alignment.CenterHorizontally,
                modifier = Modifier.fillMaxSize()
            ) { this: ColumnScope
                Image(painter = painterResource(id = R.drawable.yemekresim),
                    contentDescription = "")
```

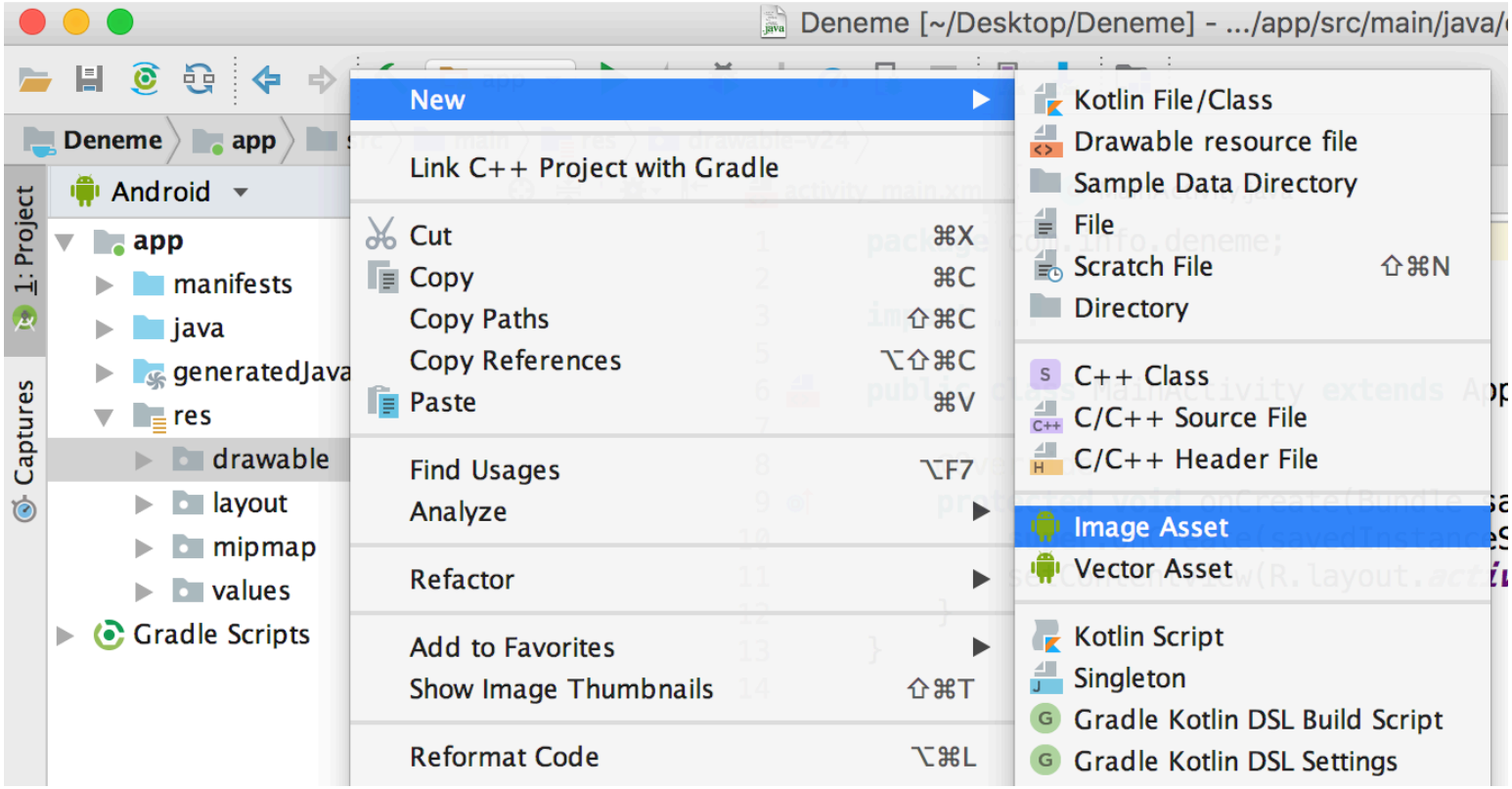

```
Row(modifier = Modifier.fillMaxWidth()) { this: RowScope
    Button(
        onClick = { Log.e( tag: "Button", msg: "Beğenildi")},
        colors = ButtonDefaults.buttonColors(
            containerColor = colorResource(id = R.color.anaRenk)
        ),
        modifier = Modifier.weight(50f)
    ) { this: RowScope
        Yazi(icerik = "Beğen")
    }

    Button(
        onClick = { Log.e( tag: "Button", msg: "Yorum yapıldı")},
        colors = ButtonDefaults.buttonColors(
            containerColor = colorResource(id = R.color.alternatifRenk)
        ),
        modifier = Modifier.weight(50f)
    ) { this: RowScope
        Yazi(icerik = "Yorum Yap")
    }
}
```


```
Column(modifier = Modifier
    .fillMaxWidth()
    .padding(all = 10.dp)) { this: ColumnScope
    Text(text = "Köfte",
        color = colorResource(id = R.color.alternatifRenk),
        fontSize = 18.sp)
    Row(
        modifier = Modifier.fillMaxWidth(),
        horizontalArrangement = Arrangement.SpaceBetween
    ) { this: RowScope
        Yazi(icerik = "Izgaraya Uygun")
        Yazi(icerik = "7 Haziran")
    }
}

Text(
    text = "Köfte harcını hazırlamak için, soğanları rendeleyin ve
    modifier = Modifier.padding(all = 10.dp),
    textAlign = TextAlign.Center
)
```

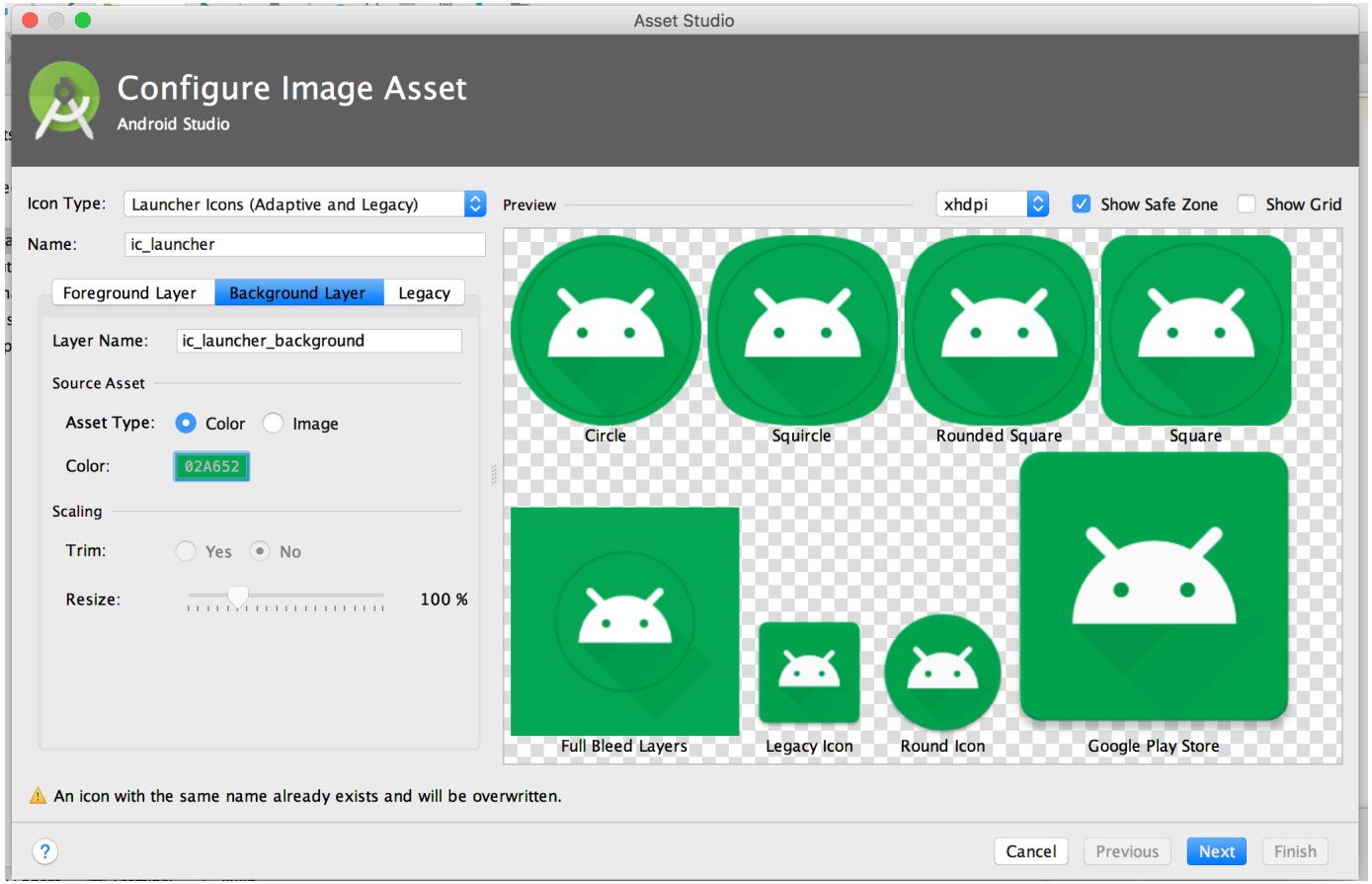
Köfte harcını hazırlamak için, soğanları rendeleyin ve maydanozları ince ince kıyın. İsterseniz, bir diş sarımsak da ekleyebilirsiniz.

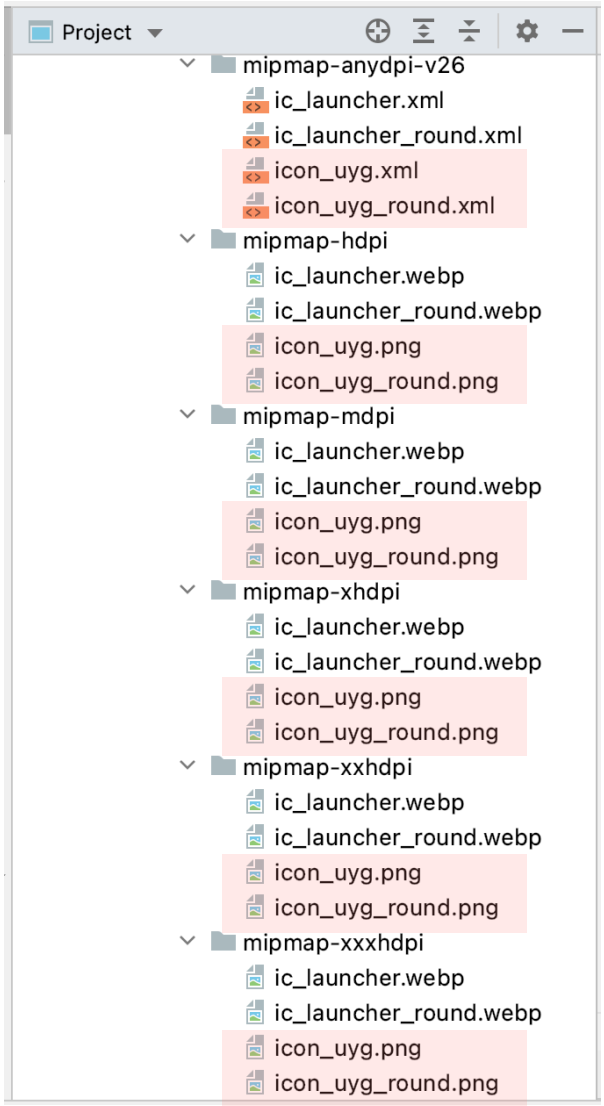
Icon Oluşturma

Icon Oluşturma



Icon Oluşturma





Manifest.xml

MainActivity.kt | AndroidManifest.xml | values/themes.xml | night/themes.xml | colors.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.yemektarifuygulamasi">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/icon_uyg"
8          android:label="YemekTarifUygulamasi"
9          android:roundIcon="@mipmap/icon_uyg_round"
10         android:supportsRtl="true"
11         android:theme="@style/Theme.YemekTarifUygulamasi">
12         <activity
13             android:name=".MainActivity"
14             android:exported="true"
15             android:label="YemekTarifUygulamasi"
16             android:theme="@style/Theme.YemekTarifUygulamasi.NoActionBar">
17             <intent-filter>
18                 <action android:name="android.intent.action.MAIN" />
19
20                 <category android:name="android.intent.category.LAUNCHER" />
21             </intent-filter>
22         </activity>
23     </application>
24
25 </manifest>
```

Teşekkürler...

kasım-adalan

kasimadalan@gmail.com

kasimadalan