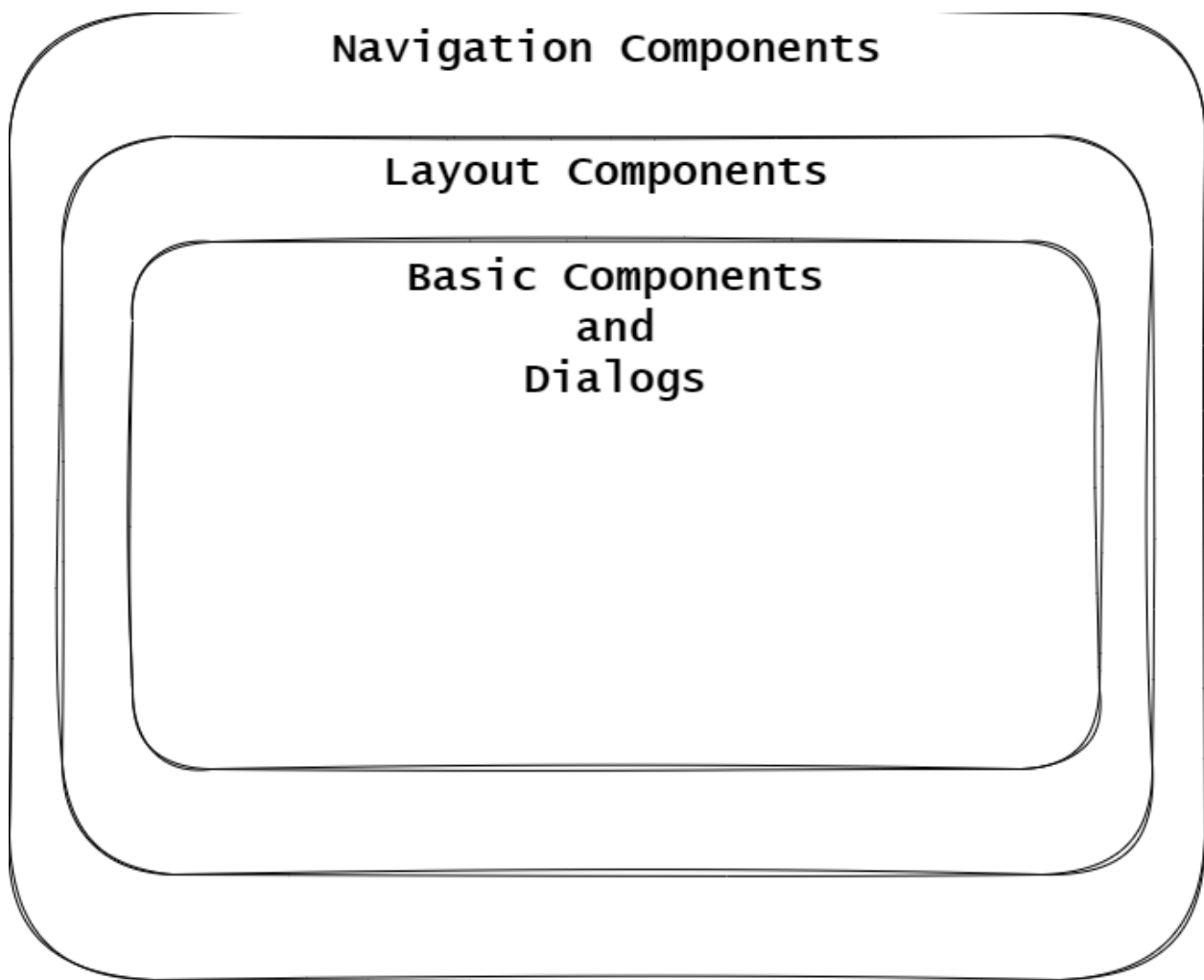


# BAB VI

## KOMPONEN UI



*Gambar 6.1 Lapisan Komponen UI di Nativescript*

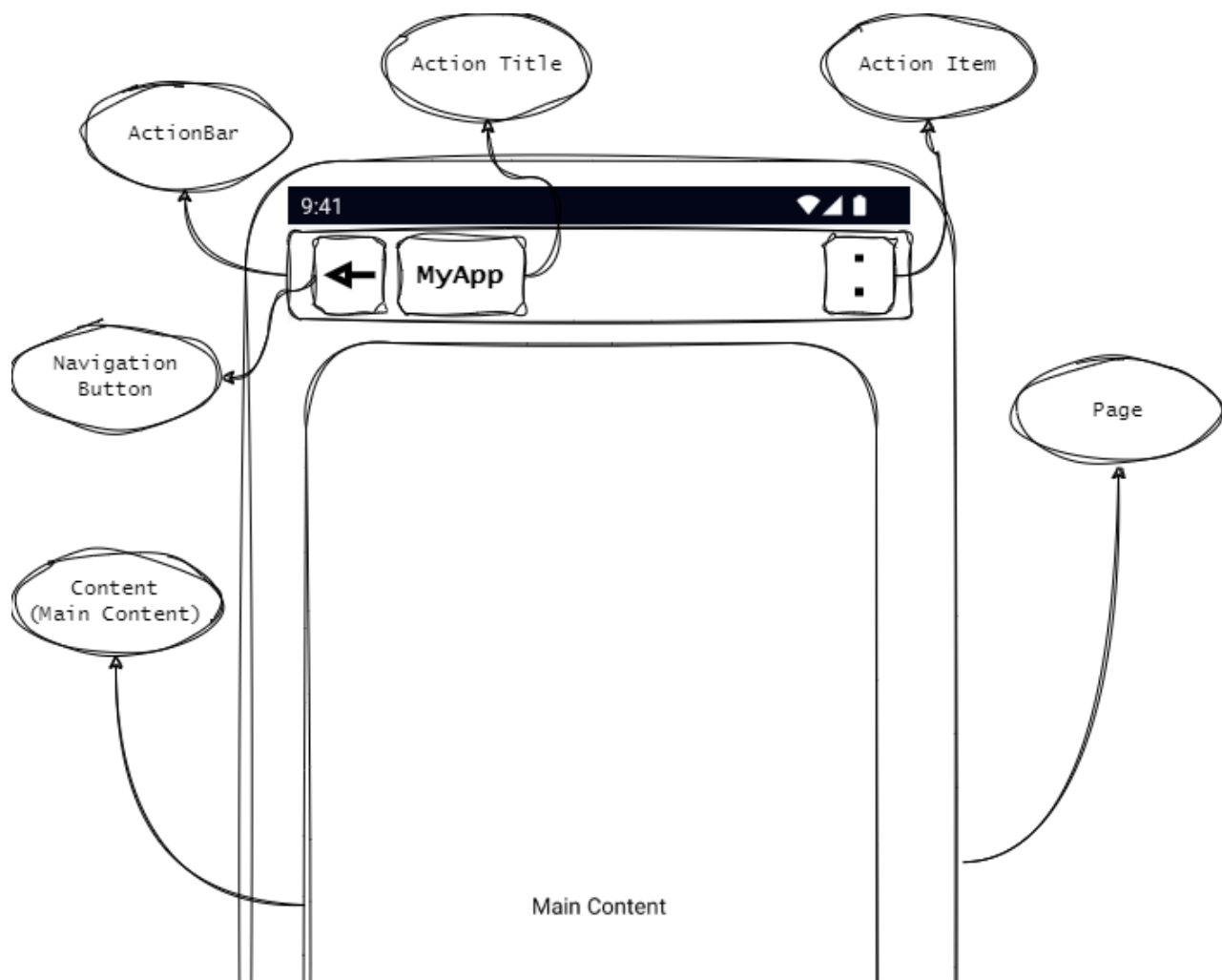
Komponen UI (User Interface) merupakan sebuah elemen visual yang digunakan dalam antarmuka pengguna, memungkinkan adanya interaksi antara pengguna dan aplikasi. Komponen UI dapat membantu pengguna dalam mengakses dan mengendalikan fungsi dan fitur aplikasi dengan mudah. Bagi pengembang sendiri Komponen UI ini sangat

dibutuhkan untuk membantu memudahkan pengembang dalam memvisualisasikan aplikasi yang mereka akan buat.

Cukup banyak komponen UI yang tersedia di Nativescript, mau itu komponen bawaan atau pun komponen yang berasal dari Plugin. Komponen UI di Nativescript dapat dibagi menjadi 3, yaitu :

- Navigation Components
- Layout Components
- Basic Components
- Dialogs

## 6.1 Navigation Components



**Gambar 6.2 Navigation Components**

Navigation Components adalah elemen visual yang menyediakan cara yang intuitif dan efisien untuk berpindah antara halaman, layar, atau bagian-bagian dari aplikasi.

### 6.1.1 Page

Komponen ini mewakili layar aplikasi yang dapat dinavigasi melalui Frame. Halaman itu sendiri hanya dapat berisi satu tampilan dengan pengecualian ActionBar yang memiliki "slot" khusus sendiri.

```
<Page>
  <ActionBar title="MyApp" />

  <!-- konten utama harus berupa satu tampilan -->
  <GridLayout>
    <Label text="Main Content" />
  </GridLayout>
</Page>
```

Kode di atas adalah contoh yang benar. Yang di maksud “satu tampilan” adalah, semua konten dalam sebuah Page harus di bungkus oleh 1 pembungkus, dalam contoh di atas adalah GridLayout yang menjadi pembungkusnya.

```
<Page>
  <ActionBar title="MyApp" />

  <GridLayout>
    <Label text=" Main Content" />
  </GridLayout>
  <GridLayout>
    <Label text=" Secondary Content " />
  </GridLayout>
</Page>
```

Di atas adalah contoh kode yang salah. Alasannya karena kode di atas memiliki 2 pembungkus yang sejajar, jika seperti itu maka akan di hitung sebagai 2 tampilan. Jika kode di atas coba di koreksi, maka kode yang benarnya akan menjadi seperti ini, Anda bisa bandingkan sendiri perbedaannya.

```
<Page>
  <ActionBar title="MyApp" />
```

```
<GridLayout>
  <Label text="Main Content" />
  <GridLayout>
    <Label text=" Secondary Content " />
  </GridLayout>
</GridLayout>
</Page>
```

### 6.1.2 Action Bar

Action Bar adalah elemen visual yang biasanya terletak di bagian atas layar dalam sebuah aplikasi. Fungsi utamanya adalah untuk menyediakan akses cepat ke fitur-fitur utama aplikasi, seperti navigasi, Action, dan menu. Di Nativescript, komponen Action Bar merupakan sebuah abstraksi dari komponen native, *ActionBar* jika di Android dan *NavigationBar* jika di IOS.

Action Bar sendiri terdiri dari beberapa bagian, di antaranya ada *Navigation Button*, *Action Title* dan *Action Item*, untuk ilustrasi Anda dapat lihat pada gambar 6.2.

#### 1. Kode dasar dari Action Bar

```
<ActionBar title="MyApp" />
```

#### 2. Bisa menambahkan ikon gambar pada bagian title

```
<ActionBar>
  <GridLayout columns="auto, *" width="100%">
    <Image src="~/assets/icon.png" width="40" height="40" />
    <Label text="MyApp" fontSize="24" />
  </GridLayout>
</ActionBar>
```

#### 3. Bisa juga di kombinasikan dengan ikon dari *Fontawesome* menggunakan *Unicode*

```
<ActionBar>
  <GridLayout columns="50, *">
    <Label class="action-bar-title" text="MyApp" colSpan="2"/>
    <Label class="fas" text="&#xf053;" tap="onBackTap"/>
  </GridLayout>
</ActionBar>
```

```
</GridLayout>
</ActionBar>
```

4. Bisa menambahkan ikon bawaan dari Android di Navigation Button, untuk daftar ikonnya lihat di sini <https://developer.android.com/reference/android/R.drawable>

```
<ActionBar title="MyApp">
  <NavigationButton text="Go back" android.systemIcon="ic_menu_back" />
</ActionBar>
```

5. Anda bisa menambahkan ActionItem sebagai menu tambahan pada Action Bar. Untuk menambah kan sebuah Action pada ActionItem, Anda bisa menambahkan atribut “tap”.

```
<ActionBar title="MyApp">
  <ActionItem android.systemIcon="ic_menu_share" android.position="actionBar" />
  <ActionItem text="Feedback" tap="aboutPage" android.position="popup" />
  <ActionItem text="About" tap="aboutPage" android.position="popup" />
</ActionBar>
```

### 6.1.3 Frame

Frame merupakan sebuah komponen UI yang digunakan untuk menampilkan dan menavigasi antar Halaman. Anda dapat menghubungkan halaman-halaman yang ada pada aplikasi dengan komponen ini.

1. Menampilkan halaman, saat tab “Page One” di klik maka akan tampil halaman One, lalu ketika tab “Page Two” di klik akan tampil halaman Two, dan begitu seterusnya.

```
<TabView android.TabsPosition="bottom">
  <TabViewItem title="Page one" >
    <Frame defaultPage="one/one-page" />
  </TabViewItem>

  <TabViewItem title="Page Two" >
    <Frame defaultPage="two/two-page" />
  </TabViewItem>
```

```
<TabViewItem title="Page Three" >
  <Frame defaultPage="three/three-page" />
</TabViewItem>
</TabView>
```

2. Komponen Frame dapat di gunakan untuk menavigasikan halaman, seperti dari halaman One berpindah ke halaman Two.

```
import { Frame } from "@nativescript/core";

Frame.topmost().navigate({
  moduleName: "two/two-page"
});
```

3. Anda juga bisa menambahkan Transition agar perpindahan halaman lebih halus dan menarik dengan adanya animasi transisi. Berikut adalah transisi yang tersedia di Nativescript.

- curl (sama seperti curlUp) (hanya di IOS)
- curlUp (hanya di IOS)
- curlDown (hanya di IOS)
- explode (hanya di Android)
- fade
- flip (sama seperti flipRight)
- flipRight
- flipLeft
- slide (sama seperti slideLeft)
- slideLeft
- slideRight
- slideTop
- slideBottom

```
import { Frame } from "@nativescript/core";

Frame.topmost().navigate({
  moduleName: "two/two-page",
  transition: {
    name: "fade",
```

```
},  
});
```

4. Anda juga bisa saling berkirim parameter antar halaman dengan menggunakan Context.

```
import { Frame } from "@nativescript/core";  
  
Frame.topmost().navigate({  
  moduleName: "two/two-page",  
  transition: {  
    name: "fade",  
  },  
  context: {  
    fullName: "Kang Cahya",  
    blog: "https://kang-cahya.com",  
  },  
});
```

Kemudian Anda dapat menangkap parameter yang dikirim dari halaman sebelumnya dengan cara berikut.

```
export function onNavigatingTo(args) {  
  const page = args.object;  
  const navData = page.navigationContext;  
  
  console.log("fullName", navData.fullName);  
  console.log("blog", navData.blog);  
}
```

5. Anda juga dapat kembali ke halaman sebelumnya tanpa harus mendefinisikan halaman sebelumnya secara spesifik, yaitu dengan menggunakan method *goBack*. Method ini akan bekerja sesuai history halaman yang telah Anda lewati sebelumnya.

```
import { Frame } from "@nativescript/core";  
  
Frame.goBack()
```

6. Fungsi `goBack` tidak akan berfungsi jika Anda menghapus History-Nya. Anda dapat menghapus History perpindahan halaman saat akan berpindah halaman, caranya dengan menambahkan properti `clearHistory`.

```
import { Frame } from "@nativescript/core";

Frame.topmost().navigate({
  moduleName: "two/two-page",
  transition: {
    name: "fade",
  },
  clearHistory: true
});
```

## 6.2 Layout Components

Komponen Layout dapat membantu Anda mengatur tata letak tampilan aplikasi Anda menjadi lebih rapi dengan cara yang lebih mudah.

### 6.2.1 Absolute Layout

Absolute Layout berfungsi untuk mengatur posisi elemen berdasarkan koordinat atas dan kiri.

1. Membuat Layout Grid dengan Absolute Layout

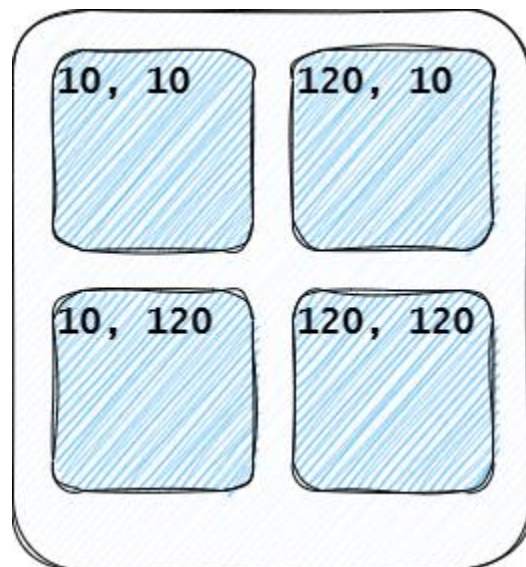
```
<AbsoluteLayout backgroundColor="#eef7ff">
  <Label
    text="10, 10"
    left="10"
    top="10"
    width="100"
    height="100"
    backgroundColor="#43b2f4"
  />
  <Label
    text="120, 10"
    left="120"
    top="10"
    width="100"
    height="100"
```



```

        backgroundColor="#43b2f4"
    />
    <Label
        text="10, 120"
        left="10"
        top="120"
        width="100"
        height="100"
        backgroundColor="#43b2f4"
    />
    <Label
        text="120, 120"
        left="120"
        top="120"
        width="100"
        height="100"
        backgroundColor="#43b2f4"
    />
</AbsoluteLayout>

```

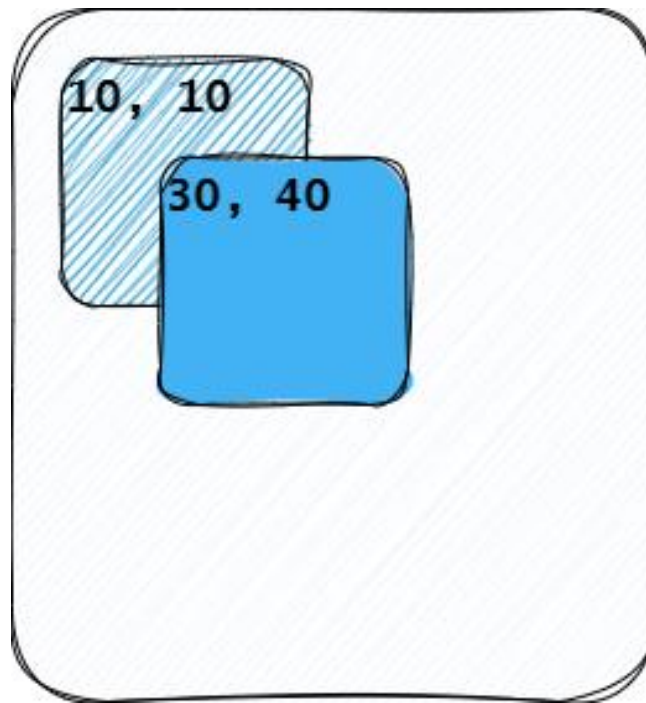


**Gambar 6.3 Membuat bentuk Grid dengan Absolute Layout**

Kode di atas hanya sebuah contoh untuk mengilustrasikan cara kerja dari AbsoluteLayout. Jika Anda ingin membuat sebuah tampilan Grid, akan lebih baik jika Anda menggunakan *Grid Layout*.

2. Selain itu, dengan komponen ini Anda dapat membuat tampilan yang bertumpuk (tumpang tindih) seperti sebuah Layer.

```
<AbsoluteLayout backgroundColor="#eef7ff">  
  <Label  
    text="10, 10"  
    left="10"  
    top="10"  
    width="100"  
    height="100"  
    backgroundColor="#0089ca"  
  />  
  <Label  
    text="30, 40"  
    left="30"  
    top="40"  
    width="100"  
    height="100"  
    backgroundColor="#43b2f4"  
  />  
</AbsoluteLayout>
```



**Gambar 6.4 Membuat tumpukan dengan Absolute Layout**

Properti Anak	Tipe	Value	Keterangan
<b>top</b>	number	0-9	Mendapatkan atau menyetel jarak tepi atas (satuan pixel)
<b>left</b>	number	0-9	Mendapatkan atau menyetel jarak tepi kiri (satuan pixel)

---

*Properti Anak adalah, dia tidak menempel pada komponen utama (dalam hal ini Absolute Layout), properti anak akan menempel pada komponen yang berada di dalam komponen utama (Absolute Layout).*

---

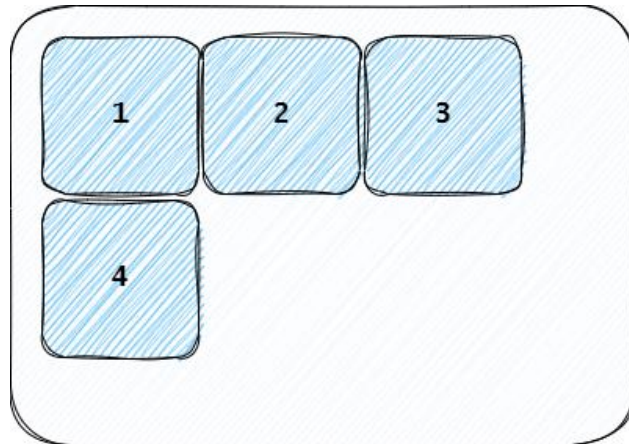
seperti yang dapat Anda lihat pada contoh di atas. Layout ini sama seperti *Position Absolute* jika dalam CSS.

## 6.2.2 Wrap Layout

Layout ini merupakan Layout pembungkus, yang mana pembungkusnya dapat Anda atur sesuka hati. Pembungkus yang ada akan di letakan sejajar secara vertical atau horizontal, dan jika pembungkus sudah mentok di ujung layar, maka pembungkus selanjutnya akan otomatis bergeser ke bawah (jika orientasinya horizontal), dan bergeser ke samping kanan (jika orientasinya vertical).

### 1. Horizontal

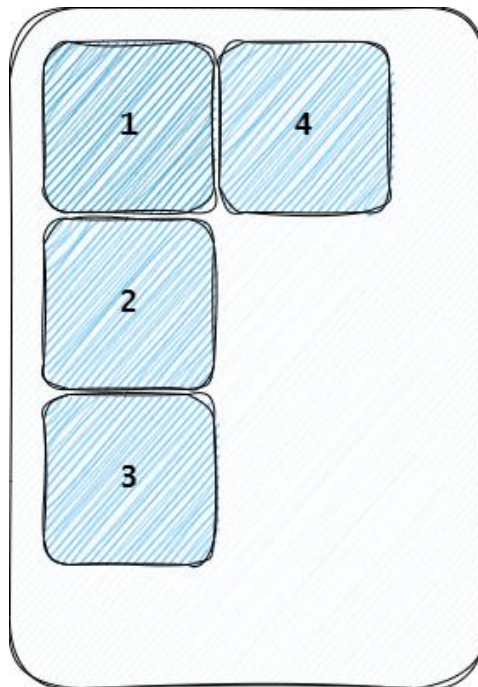
```
<WrapLayout backgroundColor="#eef7ff">
  <Label text="1" width="30%" height="30%" backgroundColor="#43B3F4" />
  <Label text="2" width="30%" height="30%" backgroundColor="#075B88" />
  <Label text="3" width="30%" height="30%" backgroundColor="#1089CA" />
  <Label text="4" width="30%" height="30%" backgroundColor="#1089CA" />
</WrapLayout>
```



**Gambar 6.5. Wrap Layout dengan Orientasi Horizontal**

## 2. Vertical

```
<WrapLayout orientation="vertical" backgroundColor="#eef7ff">  
  <Label text="1" width="30%" height="30%" backgroundColor="#43B3F4" />  
  <Label text="2" width="30%" height="30%" backgroundColor="#075B88" />  
  <Label text="3" width="30%" height="30%" backgroundColor="#1089CA" />  
  <Label text="4" width="30%" height="30%" backgroundColor="#1089CA" />  
</WrapLayout>
```



**Gambar 6.6. Wrap Layout dengan Orientasi Vertical**

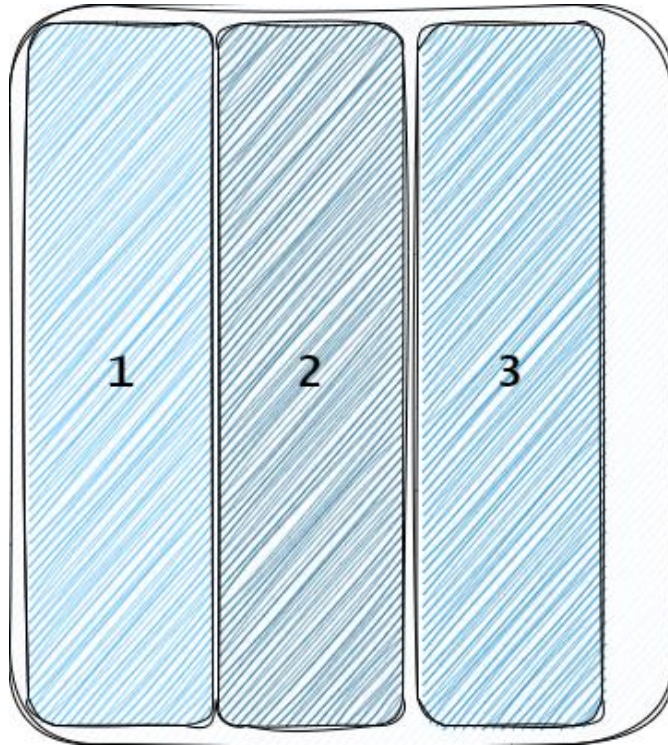
Properti	Tipe	Value	Keterangan
<b>orientation</b>	enum	horizontal	Mengatur arah tumpukan ke samping, dari kiri ke kanan
		vertical	Mengatur arah tumpukan ke bawah, dari atas ke bawah
<b>itemWidth</b>	number	0-9	Mengatur lebar setiap komponen anak, jika tidak disetel lebarnya akan otomatis disetel berdasarkan konten
<b>itemHeight</b>	number	0-9	Mengatur tinggi setiap komponen anak, jika tidak disetel tingginya akan otomatis disetel berdasarkan konten

### 6.2.3 Flexbox Layout

Layout ini konsepnya hampir sama seperti Flexbox yang ada di CSS. Layout ini memungkinkan Anda mengatur tampilan secara horizontal dan vertikal.

1. Membuat 3 baris item dengan tinggi mengikuti ukuran layar

```
<FlexboxLayout backgroundColor="#eef7ff">
  <Label text="1" width="70" backgroundColor="#43B3F4" />
  <Label text="2" width="70" backgroundColor="#075B88" />
  <Label text="3" width="70" backgroundColor="#1089CA" />
</FlexboxLayout>
```

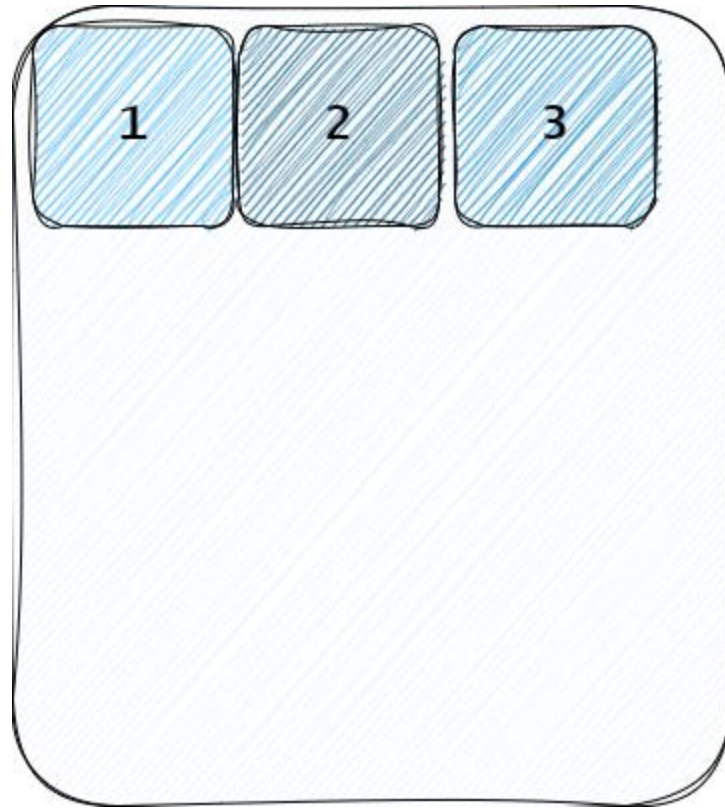


**Gambar 6.7 Tiga baris item di dalam Flexbox Layout dengan tinggi layar penuh**

2. Membuat 3 baris item yang disejajarkan dengan bagian atas kontainer FlexboxLayout (start-fleksibel).

```
<FlexboxLayout alignItems="flex-start" backgroundColor="#eef7ff">  
  <Label text="1" width="70" height="70" backgroundColor="#43B3F4" />  
  <Label text="2" width="70" height="70" backgroundColor="#075B88" />  
  <Label text="3" width="70" height="70" backgroundColor="#1089CA" />  
</FlexboxLayout>
```



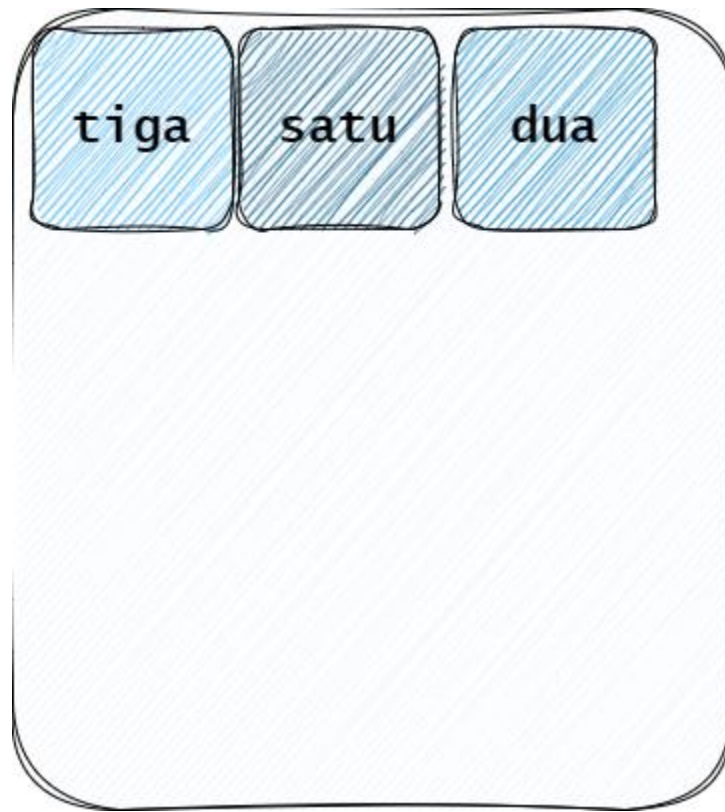


**Gambar 6.8 Tiga baris item di dalam Flexbox Layout bagian atas sejajar dengan kontainer**

3. Anda juga dapat mengurutkan baris item dengan menambahkan properti anak *"order"*.

```
<FlexboxLayout backgroundColor="#eef7ff">
  <Label
    text="satu"
    order="2"
    width="70"
    height="70"
    backgroundColor="#43B3F4"
  />
  <Label
    text="dua"
    order="3"
    width="70"
    height="70"
    backgroundColor="#075B88"
  />
  <Label
    text="tiga"
```

```
order="1"  
width="70"  
height="70"  
backgroundColor="#1089CA"  
/>  
</FlexboxLayout>
```

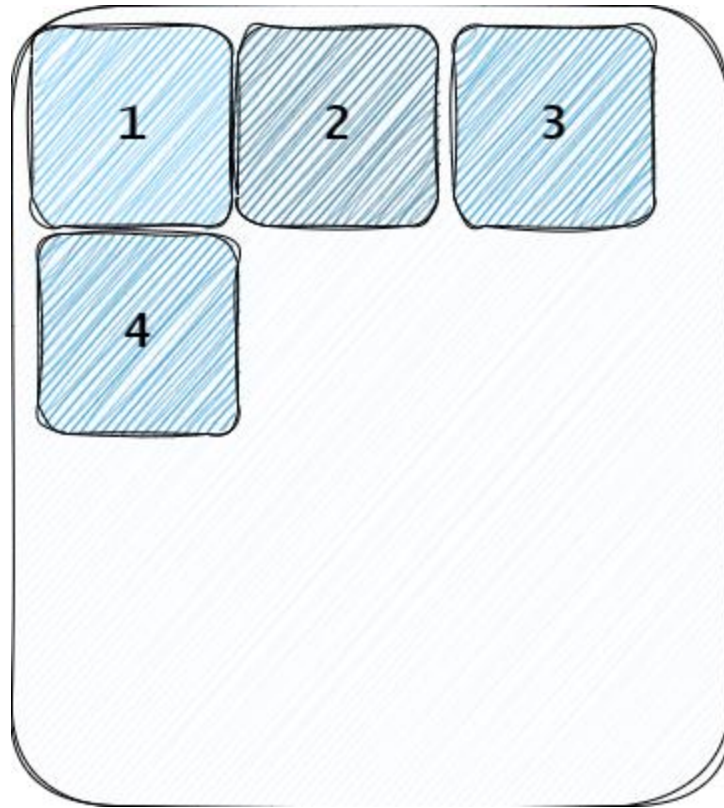


**Gambar 6.9 Mengurutkan tiga baris item di dalam Flexbox Layout**

4. Anda juga dapat melakukan Wrapping dengan menambahkan properti flexWrap

```
<FlexboxLayout flexWrap="wrap" backgroundColor="#eef7ff">  
  <Label text="1" width="30%" backgroundColor="#43B3F4" />  
  <Label text="2" width="30%" backgroundColor="#075B88" />  
  <Label text="3" width="30%" backgroundColor="#1089CA" />  
  <Label text="4" width="30%" backgroundColor="#1089CA" />  
</FlexboxLayout>
```

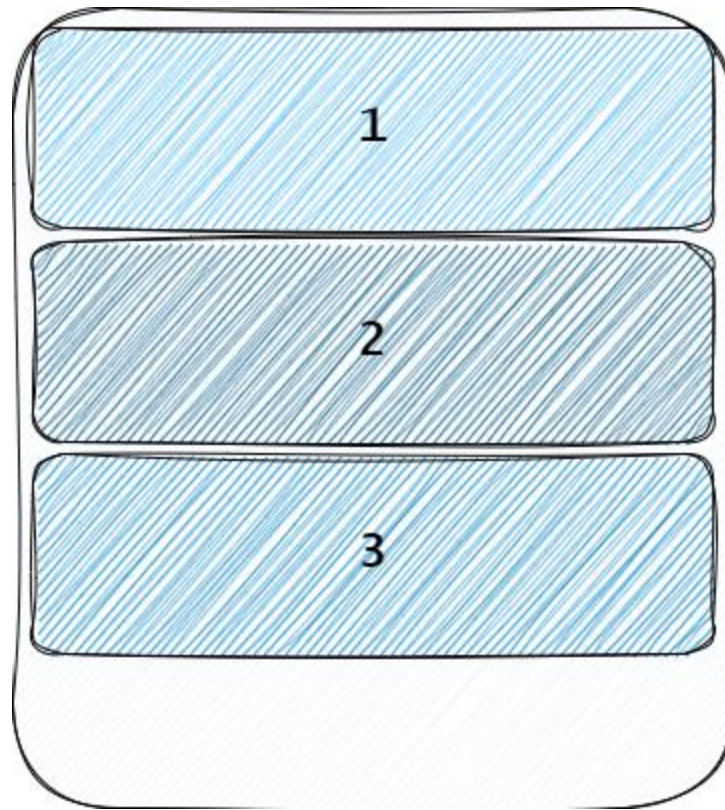




**Gambar 6.10 Wrapping di Flexbox Layout**

5. Membuat 3 baris item dengan lebar mengikuti ukuran layar

```
<FlexboxLayout flexDirection="column" backgroundColor="#eef7ff">  
  <Label text="1" height="70" backgroundColor="#43B3F4" />  
  <Label text="2" height="70" backgroundColor="#075B88" />  
  <Label text="3" height="70" backgroundColor="#1089CA" />  
</FlexboxLayout>
```



**Gambar 6.11 Tiga baris item di dalam Flexbox Layout dengan lebar selayar penuh**

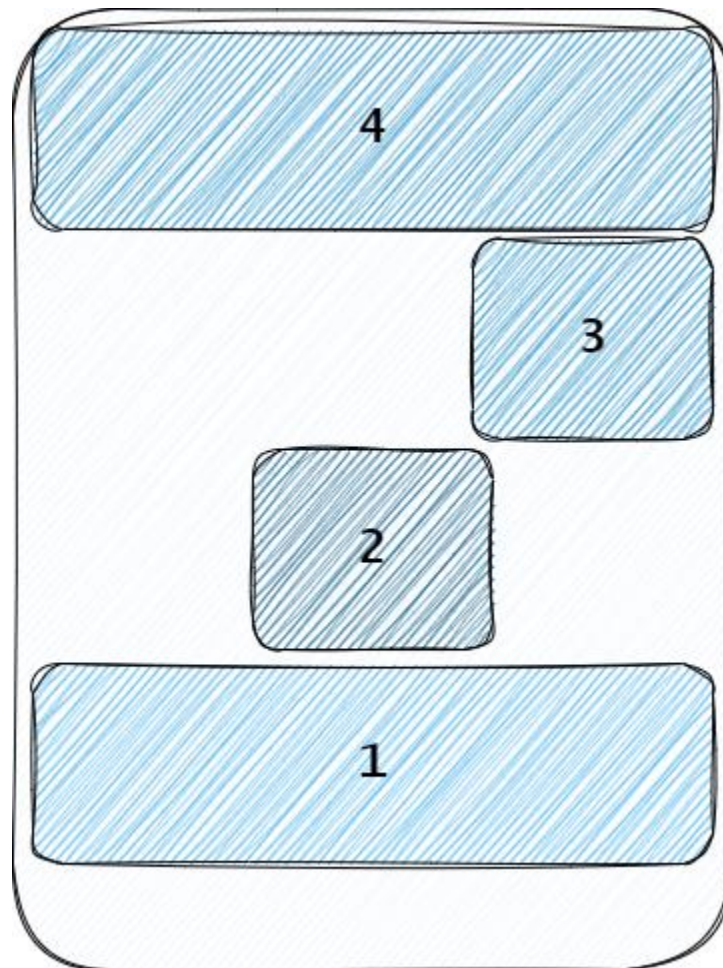
6. Reverse item dengan ukuran item yang berbeda-beda

```
<FlexboxLayout
  flexDirection="column-reverse"
  justifyContent="space-around"
  backgroundColor="#eef7ff"
>
  <Label text="1" height="70" backgroundColor="#43B3F4" />
  <Label
    text="2"
    alignSelf="center"
    width="70"
    height="70"
    backgroundColor="#075B88"
  />
  <Label
    text="3 "
    alignSelf="flex-end"
    width="70"
    height="70"
```

```

    backgroundColor="#1089CA"
  />
  <Label text="4" height="70" backgroundColor="#1089CA" />
</FlexboxLayout>

```



**Gambar 6.12 Reverse item di Flexbox Layout dengan ukuran yang berbeda-beda**

Flexbox memiliki properti sebagai berikut;

Properti	Type	Value	Keterangan
<b>flexDirection</b>	enum	row ( <i>default</i> )	Mengatur arah tumpukan secara horizontal dari kiri ke kanan
		row-reverse	Mengatur arah tumpukan secara horizontal dari kanan ke kiri
		column	Mengatur arah tumpukan secara vertical dari atas ke bawah

		column-reverse	Mengatur arah tumpukan secara vertical dari bawah ke atas
<b>flexWrap</b>	enum	nowrap ( <i>default</i> )	Memaksa konten berada dalam satu baris
		wrap	Ketika tempat sudah penuh sampai ke ujung, dia akan turun ke bawah. arahnya ditentukan oleh flexDirection
		wrap-reverse	Mirip seperti wrap, namun ini kebalikannya
<b>justifyContent</b>	enum	flex-start ( <i>default</i> )	Konten akan di mulai dari kiri atas dan di tata secara horizontal
		flex-end	Konten akan di mulai dari kanan atas dan di tata secara horizontal
		center	Konten akan di mulai dari tengah atas dan di tata secara horizontal
		space-between	Setiap konten akan mendapat ruang yang sama besarnya. Konten yang berada di awal akan di tempatkan di kiri atas, konten yang berada di tengah akan di tempatkan di tengah atas dan konten yang berada di akhir akan di tempatkan di kanan atas (akan di tata secara horizontal)
		space-around	Setiap konten akan mendapat ruang yang sama besarnya. Dan setiap konten akan mempunyai margin yang seimbang di kanan dan kirinya
<b>alignItems</b>	enum	flex-start	Item akan di mulai dari kiri atas dan di tata secara vertical
		flex-end	Item akan di mulai dari kiri bawah dan di tata secara vertical
		center	Item akan di mulai dari kiri tengah dan di tata secara vertical
		baseline	menyelaraskan item berdasarkan garis dasar teks mereka. Ini berarti teks dalam setiap item akan sejajar pada garis yang sama.
		stretch ( <i>default</i> )	semua item akan direntangkan untuk mengisi ruang yang tersedia
<b>alignContent</b>	enum	flex-start	Align Content hampir serupa dengan Justify Content, yang membedakan hanya direction-Nya saja. Jika Justify Content berfungsi untuk mengatur distribusi ruang secara Horizontal, maka Align Content berfungsi untuk mengatur distribusi ruang secara Vertical.
		flex-end	
		center	
		space-between	
		space-around	
		stretch ( <i>default</i> )	

Selain itu, Flexbox Layout juga memiliki properti anak sebagai berikut;

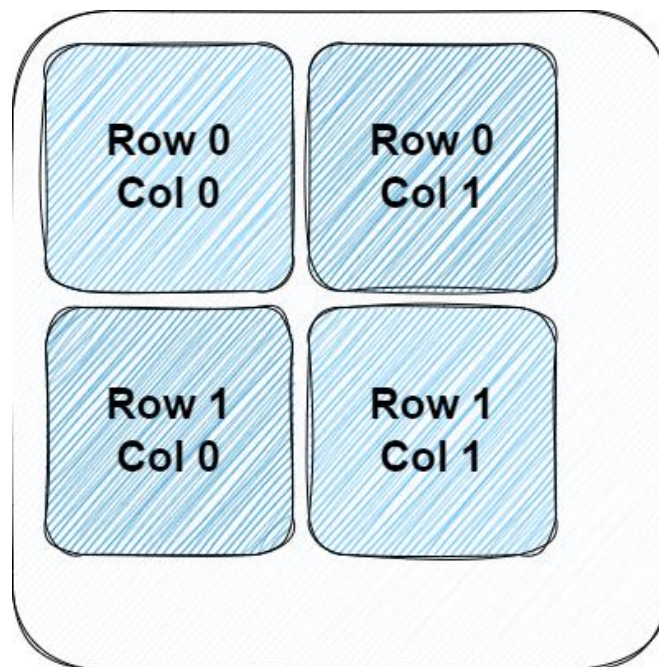
Properti Anak	Tipe	Value	Keterangan
<b>order</b>	number	0-9	Menentukan urutan item anak
<b>alignSelf</b>	enum	flex-start flex-end center baseline stretch (default)	Berfungsi untuk mengatur letak item anak

## 6.2.4 Grid Layout

Grid Layout memungkinkan Anda melakukan penataan konten dengan konsep Grid seperti Table di HTML.

1. Grid dengan ukuran kolom dan baris bernilai tetap. Pada contoh di bawah Grid berukuran 115 Pixel.

```
<GridLayout columns="115, 115" rows="115, 115">
  <Label text="Row 0, Col 0" row="0" col="0" backgroundColor="#43B3F4" />
  <Label text="Row 0, Col 1" row="0" col="1" backgroundColor="#1089CA" />
  <Label text="Row 1, Col 0" row="1" col="0" backgroundColor="#1089CA" />
  <Label text="Row 1, Col 1" row="1" col="1" backgroundColor="#43B3F4" />
</GridLayout>
```

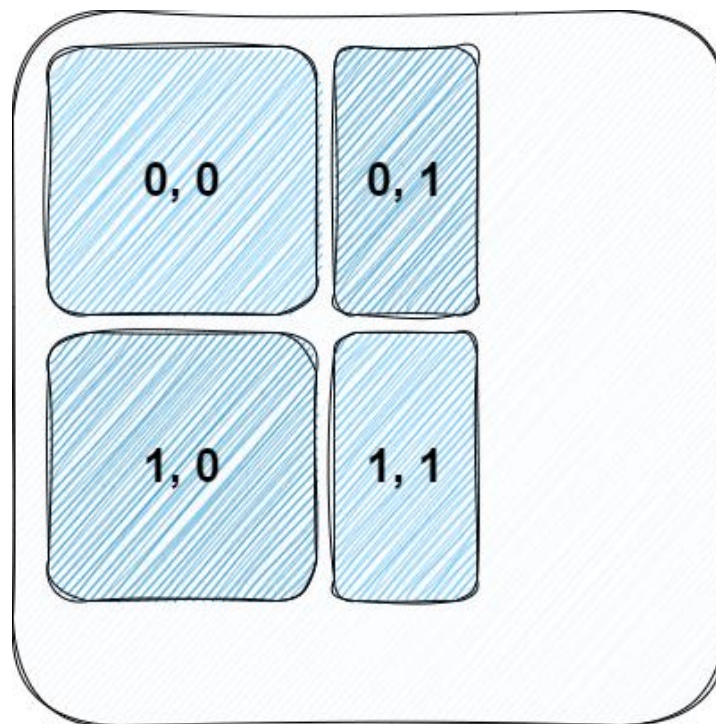


**Gambar 6.13 Grid Layout dengan ukuran Baris dan Kolom tetap**



2. Grid dengan ukuran tetap dan otomatis. Ketika di set otomatis maka ukuran Grid akan mengikuti ukuran kontennya.

```
<GridLayout columns="80, auto" rows="80, 80" backgroundColor="#eef7ff">  
  <Label text="0,0" row="0" col="0" backgroundColor="#43B3F4" />  
  <Label text="0,1" row="0" col="1" backgroundColor="#1089CA" />  
  <Label text="1,0" row="1" col="0" backgroundColor="#1089CA" />  
  <Label text="1,1" row="1" col="1" backgroundColor="#43B3F4" />  
</GridLayout>
```

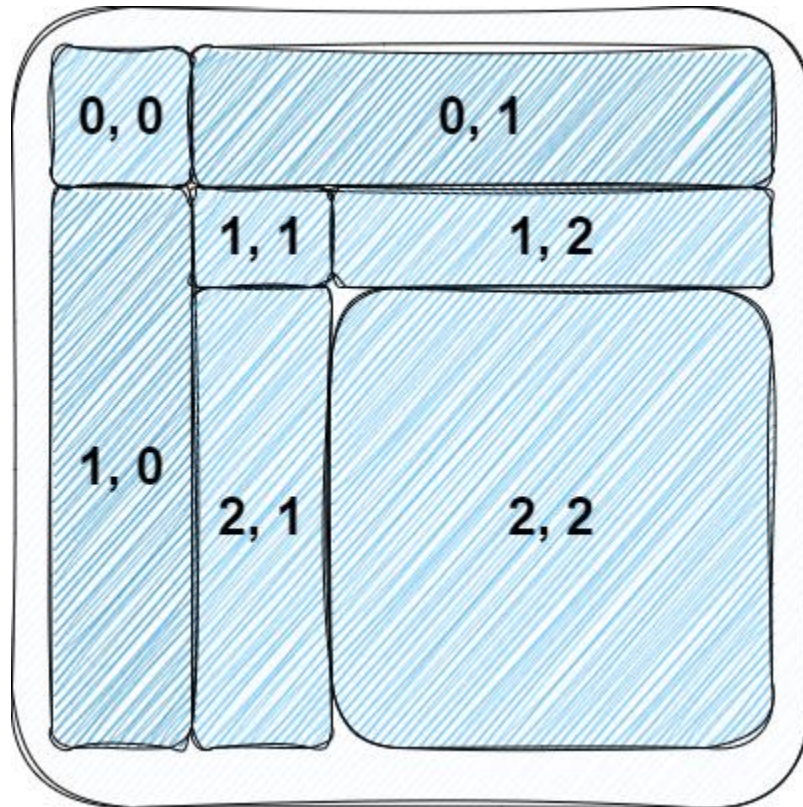


**Gambar 6.14 Grid Layout dengan ukuran tetap dan otomatis**

3. Grid dengan ukuran campuran dan gabungan sel

```
<GridLayout columns="40, auto, *" rows="40, auto, *" backgroundColor="#eef7ff">  
  <Label text="0,0" row="0" col="0" backgroundColor="#43B3F4" />  
  <Label text="0,1" row="0" col="1" colSpan="2" backgroundColor="#075B88" />  
  <Label text="1,0" row="1" col="0" rowSpan="2" backgroundColor="#1089CA" />  
  <Label text="1,1" row="1" col="1" backgroundColor="#43B3F4" />  
  <Label text="1,2" row="1" col="2" backgroundColor="#1089CA" />  
  <Label text="2,1" row="2" col="1" backgroundColor="#075B88" />
```

```
<Label text="2,2" row="2" col="2" backgroundColor="#43B3F4" />
</GridLayout>
```



**Gambar 6.15** Grid Layout dengan ukuran campuran dan gabungan sel

Properti	Tipe	Value	Keterangan
<b>columns</b>	string	*	Ukuran akan di set otomatis dengan cara mengisi ruang yang tersedia
		auto	Ukuran akan di set otomatis dengan cara mengikuti ukuran konten di dalamnya
		0-9 (number)	Ukuran akan di set dengan nilai tetap berdasarkan angka. Setiap angka yang terdefinisi akan memiliki satuan Pixel, saat menuliskan nilai, Anda tidak perlu menyebutkan kembali nilai satuannya, cukup angkanya saja.
<b>rows</b>	String	*, auto, 0-9 (number)	Kurang lebih keterangannya sama seperti yang ada pada columns.

Properti Anak	Tipe	Value	Keterangan
---------------	------	-------	------------

<b>row</b>	number	0-9	Elemen baris
<b>col</b>	number	0-9	Elemen kolom
<b>rowSpan</b>	number	0-9	Menggabungkan sel baris, properti ini mirip seperti yang ada pada Table di HTML
<b>colSpan</b>	number	0-9	Menggabungkan sel kolom, properti ini mirip seperti yang ada pada Table di HTML

## 6.2.5 Stack Layout

Stack Layout bekerja menyusun tampilan dengan cara menumpuk tiap komponen anak yang ada di dalamnya, tumpukan yang di lakukan bisa secara horizontal (kiri ke kanan) atau secara vertical (atas ke bawah).

---

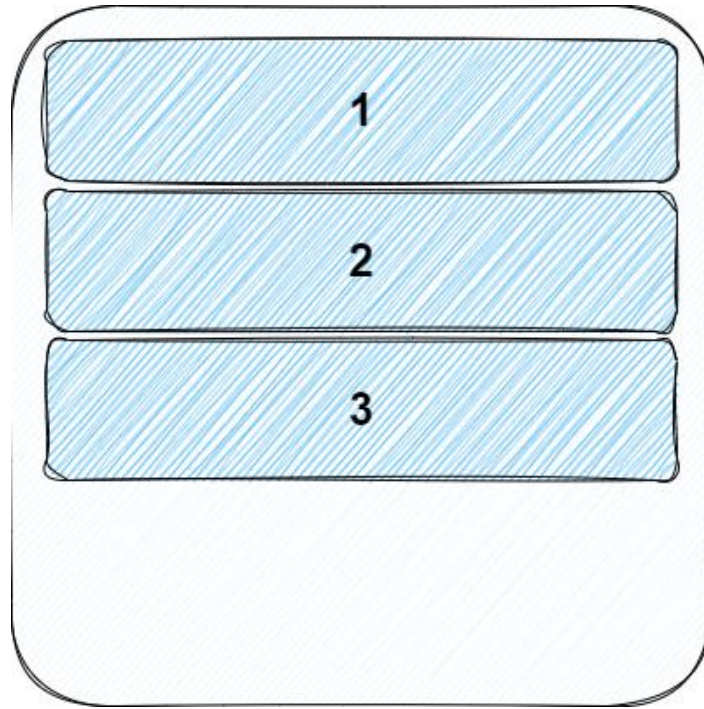
*Tidak di rekomendasikan membuat Layout yang kompleks menggunakan Stack Layout, seperti menggunakan Stack Layout di dalam Stack Layout sampai membentuk turunan yang banyak di dalamnya. Penggunaan Stack Layout secara berlebihan seperti itu akan mengakibatkan performa buruk dan rendering lambat. Jika Anda ingin membuat struktur tampilan yang kompleks, pertimbangkan untuk beralih ke GridLayout atau FlexboxLayout untuk kinerja yang lebih baik.*

---

1. Vertical Stack Layout, ini adalah orientasi bawaan jika Anda tidak mendefinisikan orientasinya secara jelas.

```
<StackLayout backgroundColor="#eef7ff">
  <Label text="1" height="70" backgroundColor="#43B3F4" />
  <Label text="2" height="70" backgroundColor="#1089CA" />
  <Label text="3" height="70" backgroundColor="#075B88" />
</StackLayout>
```





**Gambar 6.16 Vertical Stack Layout**

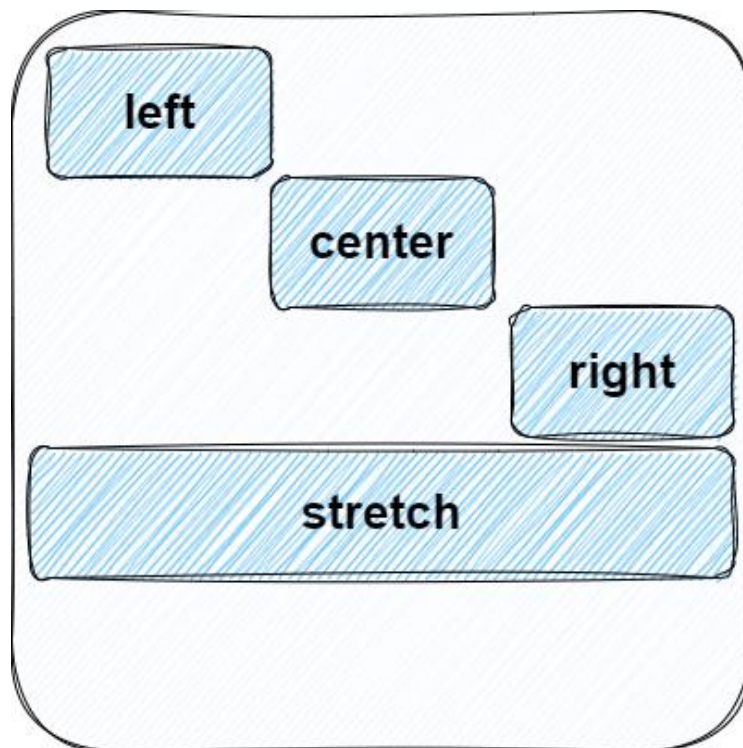
2. Mengatur tata letak item yang berada di dalam Stack Layout secara Vertical

```
<StackLayout backgroundColor="#eef7ff">
  <Label
    text="left"
    horizontalAlignment="left"
    width="33%"
    height="70"
    backgroundColor="#43B3F4"
  />
  <Label
    text="center"
    horizontalAlignment="center"
    width="33%"
    height="70"
    backgroundColor="#1089CA"
  />
  <Label
    text="right"
    horizontalAlignment="right"
    width="33%"
    height="70"
  />
```

```

        backgroundColor="#075B88"
    />
    <Label
        text="stretch"
        horizontalAlignment="stretch"
        height="70"
        backgroundColor="#43B3F4"
    />
</StackLayout>

```



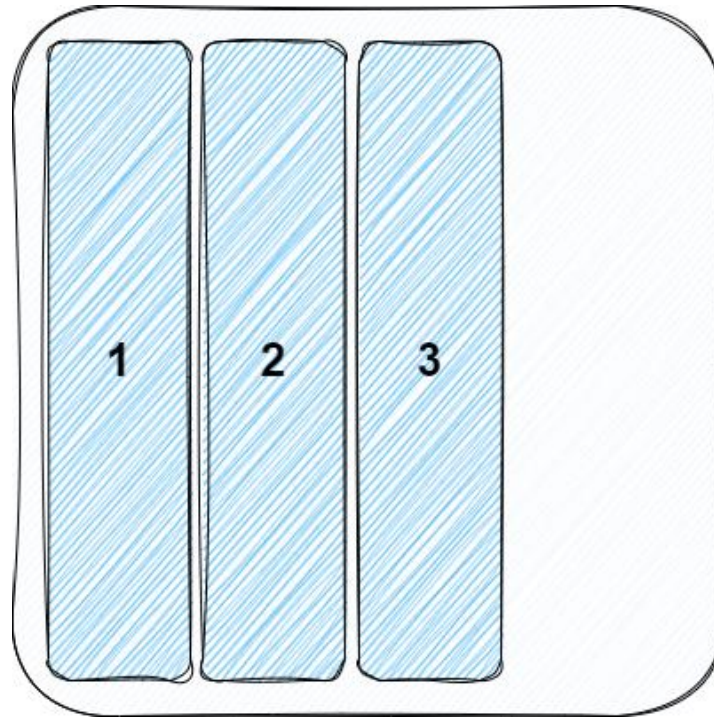
**Gambar 6.17 Mengatur letak item di dalam Stack Layout secara Vertical**

### 3. Horizontal Layout

```

<StackLayout orientation="horizontal" backgroundColor="#eef7ff">
    <Label text="1" width="70" backgroundColor="#43B3F4" />
    <Label text="2" width="70" backgroundColor="#1089CA" />
    <Label text="3" width="70" backgroundColor="#075B88" />
</StackLayout>

```



**Gambar 6.18 Horizontal Stack Layout**

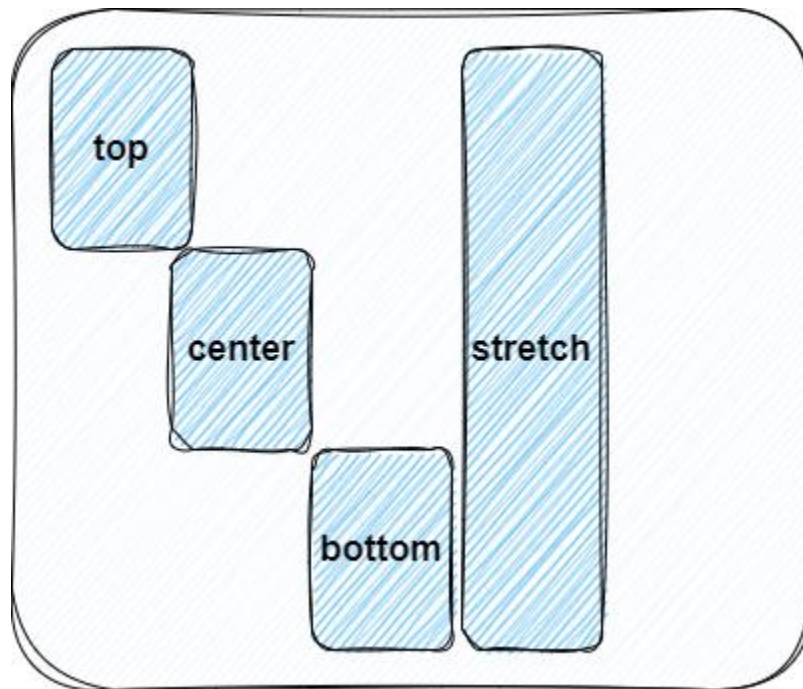
4. Mengatur tata letak item yang berada di dalam Stack Layout secara Horizontal

```
<StackLayout orientation="horizontal" backgroundColor="#eef7ff">
  <Label
    text="top"
    verticalAlignment="top"
    width="70"
    height="33%"
    backgroundColor="#43B3F4"
  />
  <Label
    text="center"
    verticalAlignment="center"
    width="70"
    height="33%"
    backgroundColor="#1089CA"
  />
  <Label
    text="bottom"
    verticalAlignment="bottom"
    width="70"
    height="33%"
  />
```

```

        backgroundColor="#075B88"
    />
    <Label
        text="stretch"
        verticalAlignment="stretch"
        width="70"
        backgroundColor="#43B3F4"
    />
</StackLayout>

```



**Gambar 6.19 Mengatur letak item di dalam Stack Layout secara Horizontal**

Properti	Tipe	Value	Keterangan
<b>orientation</b>	enum	vertical ( <i>default</i> )	Item akan di tumpuk dari atas ke bawah
		horizontal	Item akan di tumpuk dari kiri ke kanan

---

*“verticalAlignment” dan “horizontalAlignment” bukanlah properti anak dari Stack Layout, itu adalah properti umum yang dapat Anda gunakan di dalam komponen Layout lainnya.*

---

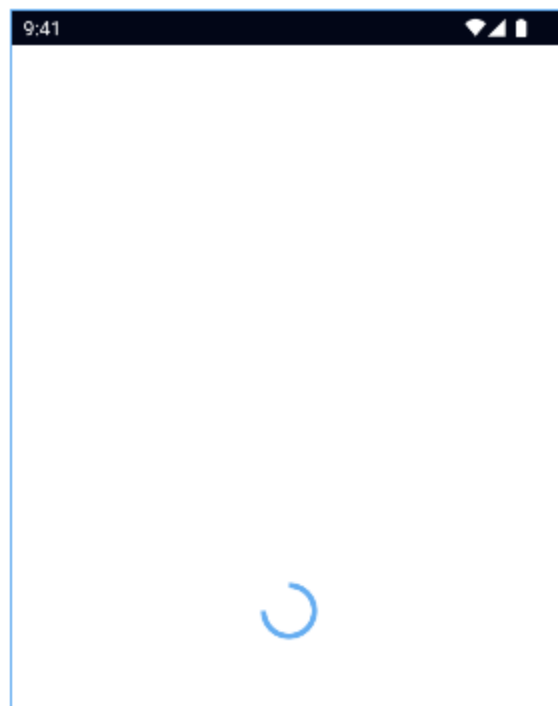
## 6.3 Basic Components

Ini adalah komponen dasar dalam pengembangan antarmuka (User Interface), yang menyediakan elemen-elemen visual dasar untuk membangun aplikasi. Komponen dasar ini adalah blok kecil yang digunakan untuk membuat antarmuka yang lebih kompleks. Contohnya seperti Label, Image, Button, dan elemen-elemen Input.

### 6.3.1 Activity Indicator

Activity Indicator adalah sebuah komponen animasi Loading berbentuk bulat. Komponen ini merupakan abstraksi dari Komponen Native *android.widget.ProgressBar* jika di Android dan *UIActivityIndicatorView* jika di IOS.

```
<ActivityIndicator busy="true" />
```



**Gambar 6.20** *Komponen Activity Indicator*

Properti	Tipe	Value	Keterangan
<b>busy</b>	boolean	true atau false	Jika False Loading akan mati, dan jika True Loading akan aktif

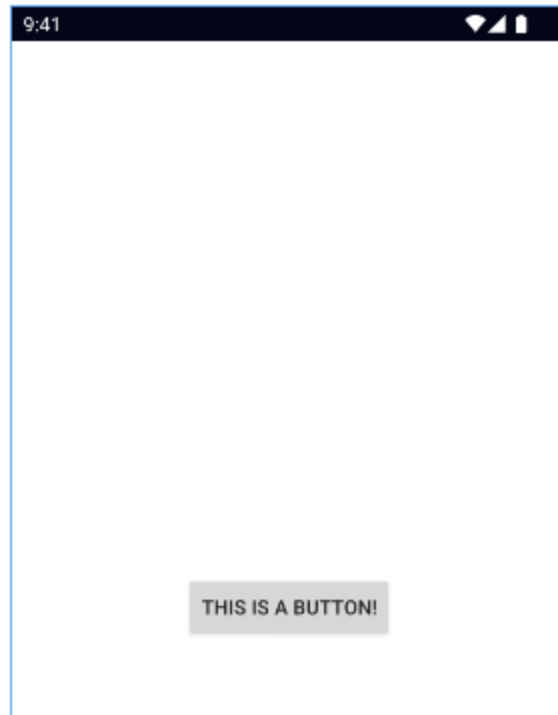
Komponen ini hanya mempunyai satu properti saja, Anda dapat mengganti ukuran dan warna Loading-Nya dengan menggunakan Inline CSS, seperti contoh berikut.

```
<ActivityIndicator busy="true" color="#000" height="40" width="40" />
```

### 6.3.2 Button

Komponen ini merupakan abstraksi komponen native *android.widget.Button* jika di Android dan *UIButton* jika di IOS.

```
<Button text="This is a button!" />
```



**Gambar 6.21 Komponen Button**

Anda juga bisa menata teks dengan style yang berbeda-beda pada Button secara lebih spesifik dengan menggunakan *FormattedString* dan *Span*, kurang lebih seperti ini contohnya.

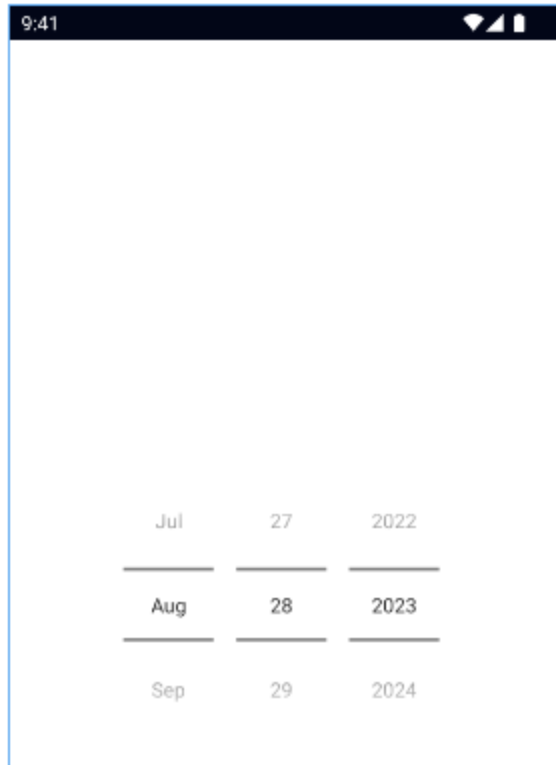
```
<Button textWrap="true">
  <FormattedString>
    <Span text="Kunjungi blog " fontWeight="bold" />
    <Span text="https://kang-cahya.com " fontStyle="italic" style="color: green" />
  </FormattedString>
</Button>
```

Properti	Tipe	Keterangan
<b>text</b>	string	Mengatur teks yang akan ditampilkan pada Button
<b>textWrap</b>	boolean	Jika di set True, saat teks terlalu panjang maka teks pada Button akan turun ke bawah. (default : false)
<b>isEnabled</b>	boolean	Untuk menonaktifkan Button (default : true)

### 6.3.3 Date dan Time Picker

Datepicker adalah komponen Input digunakan untuk memilih informasi tanggal. Komponen ini merupakan abstraksi dari komponen native *android.widget.DatePicker* jika di Android dan *UIDatePicker* jika di IOS.

```
<DatePicker year="2023" month="8" day="28" />
```



**Gambar 6.22 Komponen DatePicker**

Properti	Tipe	Keterangan
<b>date</b>	Date	Untuk melakukan Get atau Set data tanggal secara penuh (tanggal, bulan dan tahun)
<b>minDate</b>	Date	Untuk membatasi minimum tanggal yang dapat di pilih
<b>maxDate</b>	Date	Untuk membatasi maksimum tanggal yang dapat di pilih
<b>day</b>	Number	Untuk melakukan Get atau Set data tanggal saja (misal tanggal 28)
<b>month</b>	Number	Untuk melakukan Get atau Set data bulan saja (misal bulan 8)
<b>year</b>	number	Untuk melakukan Get atau Set data tahun saja (misal tahun 2023)

Ada juga komponen Timepicker, komponen ini terpisah dari DatePicker. Komponen Timepicker merupakan abstraksi dari komponen native *android.widget.TimePicker* jika di Android dan *UIDatePicker* jika di IOS.

```
<TimePicker hour="9" minute="41" />
```





**Gambar 6.23** *Komponen Timepicker*

Properti	Tipe	Keterangan
<b>hour</b>	Number	Untuk melakukan Get atau Set data Jam. Secara default akan terisi data waktu saat ini.
<b>minute</b>	Number	Untuk melakukan Get atau Set data Menit. Secara default akan terisi data waktu saat ini.
<b>time</b>	Date	Untuk melakukan Get atau Set data Waktu secara penuh (Jam dan Menit). Secara default akan terisi data waktu saat ini.
<b>minHour</b>	Number	Untuk membatasi minimum Jam yang dapat dipilih
<b>maxHour</b>	Number	Untuk membatasi maksimum Jam yang dapat dipilih
<b>minMinute</b>	Number	Untuk membatasi minimum Menit yang dapat dipilih
<b>maxMinute</b>	Number	Untuk membatasi maksimum Menit yang dapat dipilih

Untuk *minDate* dan *maxDate*, berisi rentang waktu minimum dan maksimum yang dapat di pilih, misalkan *minDate* di set dengan tanggal 1 Mei 2024 dan *maxDate* di set dengan tanggal 17 Agustus 2024, maka tanggal yang dapat di pilih adalah antara rentang waktu 1 Mei sampai 17 Agustus saja. Timepicker juga begitu, di sana terdapat properti *minHour*, *maxHour*, *minMinute* dan *maxMinute*.

Pengimplementasian komponen DatePicker dan Timepicker di Nativescript menurut saya agak sedikit ribet. Walau begitu Anda tetap masih bisa menggunakannya, Cuma pada praktiknya saya jarang sekali menggunakan komponen ini, saya lebih suka menggunakan komponen yang dalam bentuk plugin, bisa lihat di sini : <https://docs.nativescript.org/plugins/datetimetypepicker>

Untuk memasangnya Anda bisa menjalankan perintah berikut.

```
ns plugin install @nativescript/datetimetypepicker
```

Kemudian jangan lupa untuk menambahkan Namespace berikut pada Tag <Page />

```
xmlns:datetime="@nativescript/datetimetypepicker"
```

Contoh kode lengkapnya kurang lebih seperti ini

```
<Page
  xmlns="http://schemas.nativescript.org/tns.xsd"
  xmlns:datetime="@nativescript/datetimetypepicker"
>
  <datetime:DatePickerField hint="select date"></datetime:DatePickerField>
  <datetime:TimePickerField hint="select time"></datetime:TimePickerField>
  <datetime:DateTimePickerFields
    hintDate="select date"
    hintTime="select time"
  ></datetime:DateTimePickerFields>
</Page>
```

### 6.3.4 Html View

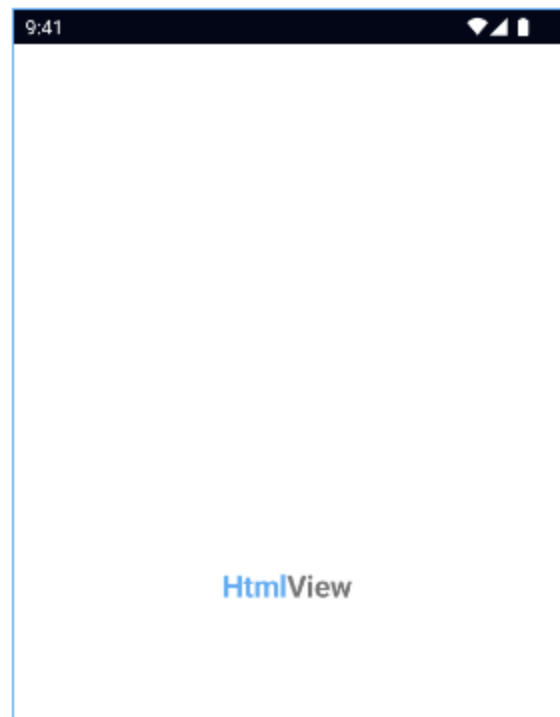
Komponen ini berfungsi untuk menampilkan konten HTML statis yang sederhana. Komponen ini merupakan abstraksi dari komponen native *android.widget.TextView* jika di Android dan *UITextView* jika di IOS.

```
<!-- Kode XML-nya -->

<HtmlView html="{{ htmlString }}" />
```



```
/** Kode JS-nya */  
  
const htmlString = `  
<h1 style="color: black; font-weight: bold;">  
  <span style="color: #65adf1;">Html</span>View  
</h1>  
`;  
;
```



*Gambar 6.24 Komponen HtmlView*

Properti	Tipe	Keterangan
html	string	Tempat HTML konten yang akan di tampilkan
selectable	boolean	Apakah konten HTML dapat di pilih (blok atau select) atau tidak

### 6.3.5 Image

Komponen ini berfungsi untuk menampilkan konten gambar. Komponen Image merupakan sebuah abstraksi dari komponen native *android.widget.ImageView* jika di Android dan *UIImageView* jika di IOS.

1. Menampilkan gambar dari *App\_Resources*. Untuk menampilkan gambar dari lokasi spesifik dari tiap Platform, Untuk android lokasinya gambarnya ada di *App\_Resources/Android/drawable-XXX/*, lalu untuk IOS ada di *App\_Resources/iOS/*. Setiap ingin menampilkan gambar yang bersumber dari *App\_Resources* harus diawali dengan Prefix "*res://*", kemudian diiringi dengan nama file tanpa ekstensi.

```
<Image src="res://icon" />
```

2. Menampilkan gambar dari direktori *app/assets*. Anda dapat membuat folder dengan nama "assets" di dalam folder "app" jika memang belum ada. Setelah folder di buat, Anda dapat meletakkan aset gambar pada folder tersebut. Dan untuk menampilkan gambar tersebut, Anda bisa menggunakan Prefix "*~/assets/*".

```
<Image src="~/assets/nama_gambar.png" />
```

3. Menampilkan gambar dari URL

```
<Image src=https://example.com/image.png />
```

4. Menampilkan gambar dari base64-encoded

```
<Image src="data:image/png;base64,iVBORw..." />
```

5. Menampilkan font icon (ex : Font Awesome) sebagai gambar. Pastikan Anda menggunakan Prefix "*font://*". Pastikan juga Anda sudah meletakkan file font icon pada folder "*app/fonts*", kemudian membuat Class font icon pada file CSS.

```
.fab {  
  font-family: "Font Awesome 5 Brands", "fa-brands-400";
```

```

font-weight: 400;
}

.fas {
font-family: "Font Awesome 5 Free", "fa-solid-900";
font-weight: 900;
}

.far {
font-family: "Font Awesome 5 Free", "fa-regular-400";
font-weight: 400;
}

```

```
<Image src="font://&#xf004;" class="fas" />
```

6. Sejak Nativescript versi 8.8, untuk perangkat IOS versi 8.8 ke atas, Anda dapat menggunakan SF Symbols (<https://developer.apple.com/sf-symbols>) dari Apple. Untuk menampilkannya Anda harus menggunakan Prefix “sys://”.

```

<image
src="sys://photo.on.rectangle.angled"
tintColor="green"
iosSymbolEffect="bounce"
iosSymbolScale="small"
/>

```

Properti	Tipe	Value	Keterangan
<b>src</b>	string		String dapat berupa http://, https://, res://, font:// atau Path absolut seperti ~/assets/image.png.
<b>stretch</b>	enum	none, aspectFill, aspectFit, fill	Mendapatkan atau mengatur ukuran gambar dengan cara mengisi ruang yang dialokasikan
<b>loadMode</b>	enum	sync, async	Mengatur mode untuk bagaimana cara gambar di tampilkan
<b>iosSymbolEffect</b>	enum	<i>Di bawah</i>	[Hanya IOS] untuk mengatur animasi dari SF Symbol

<b>iosSymbolScale</b>	number	0-9	[Hanya IOS] untuk mengatur skala ukuran dari SF Symbol
<b>tintColor</b>	string	<i>Kode warna</i>	Untuk memberi warna pada gambar

Untuk *iosSymbolEffect* yang tersedia saat ini adalah sebagai berikut:

- appear
- appearUp
- appearDown
- bounce
- bounceUp
- bounceDown
- disappear
- disappearDown
- disappearUp
- pulse
- scale
- scaleDown
- scaleUp
- variableColor
- breathe
- breathePlain
- breathePulse
- rotate
- rotateClockwise
- rotateCounterClockwise
- wiggle
- wiggleBackward
- wiggleClockwise
- wiggleCounterClockwise
- wiggleDown
- wiggleForward
- wiggleUp
- wiggleLeft
- wiggleRight.

Jika aplikasi yang Anda buat melibatkan banyak gambar dalam satu halaman, hal itu bisa cepat menghabiskan banyak memori. Anda dapat melakukan optimasi dengan cara

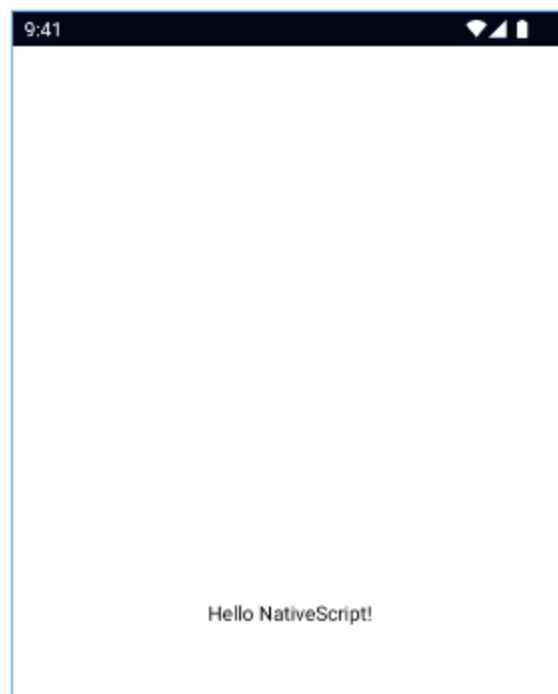
melakukan *caching* pada tiap gambar yang sudah pernah di muat sebelumnya. Untuk melakukan hal itu, Anda dapat menggunakan 2 plugin berikut :

- **@triniwiz/image-cache-it**, plugin ini berbasis *Glide* jika di Android dan *SDWebImage* jika di IOS.
- **@nativescript-community/ui-image**, plugin ini berbasis *Fresco* jika di Android dan *SDWebImage* jika di IOS.

### 6.3.6 Label

Komponen ini berfungsi untuk menampilkan teks biasa. `<Label>` di Nativescript berbeda dengan `<label>` yang ada di HTML, sekilas tampak serupa namun sebenarnya berbeda. Penulisan di Nativescript menggunakan huruf kapital di awal, sedangkan di HTML menggunakan huruf kecil semua. Lumayan banyak yang terkecoh dengan ini sehingga saat Nativescript mulai di jalankan, akan menimbulkan Error. Komponen ini merupakan abstraksi dari *android.widget.TextView* jika di Android dan *UILabel* jika di IOS.

```
<Label text="Hello NativeScript!" />
```



**Gambar 6.26 Komponen Label**

Anda juga bisa menata teks dengan style yang berbeda-beda pada Label secara lebih spesifik dengan menggunakan FormattedString dan Span, kurang lebih seperti ini contohnya.

```
<Label textWrap="true">
  <FormattedString>
    <Span text="Kunjungi blog " fontWeight="bold" />
    <Span text="https://kang-cahya.com " fontStyle="italic" style="color: green" />
  </FormattedString>
</Label>
```

Properti	Tipe	Keterangan
<b>text</b>	string	Mengatur teks yang akan di tampilkan
<b>textAlignment</b>	enum	Mengatur letak teks. <i>Initial, left (default), center, right</i>
<b>textWrap</b>	boolean	Jika di set True, saat teks terlalu panjang maka teks pada Button akan turun ke bawah. (default : false)

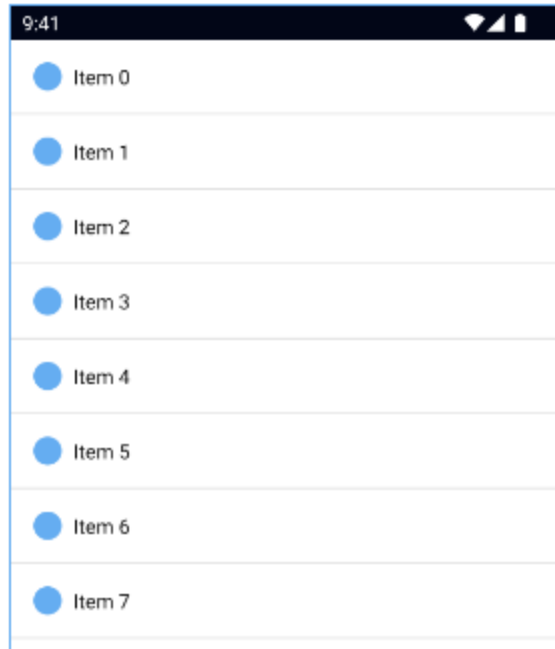
### 6.3.7 ListView

ListView berfungsi untuk menampilkan data Array dalam bentuk List. Komponen ini akan merender Items secara vertikal dengan Scroll.

```
<ListView items="{{ items }}">
  <ListView.itemTemplate>
    <!-- itemTemplate hanya dapat memiliki satu elemen root -->
    <GridLayout padding="16" columns="20, *">
      <ContentView width="20" height="20" borderRadius="20" />
      <Label text="{{ title }}" col="1" textWrap="true" marginLeft="8" />
    </GridLayout>
  </ListView.itemTemplate>
</ListView>
```

```
const items = Array.from({ length: 100 }).map((_, i) => ({
  title: `Item ${i}`,
}))
```





**Gambar 6.27 Komponen ListView**

Properti	Tipe	Keterangan
<b>items</b>	Array ObservableArray	Berisi item dari List
<b>separatorColor</b>	String (kode warna)	Untuk mengatur warna garis pembatas antara Item satu dengan yang lainnya
<b>rowHeight</b>	number	Berfungsi untuk mengatur tinggi dari Item di dalam List
<b>itemTap</b>	event	Mendapatkan nilai dari item yang di Tap

Method	Keterangan
<b>refresh()</b>	Untuk merefresh semua item
<b>scrollToIndex(index: number)</b>	Melakukan scroll menuju item berdasarkan indeks tertentu
<b>scrollToIndexAnimated(index: number)</b>	Melakukan scroll menuju item berdasarkan indeks tertentu dengan animasi

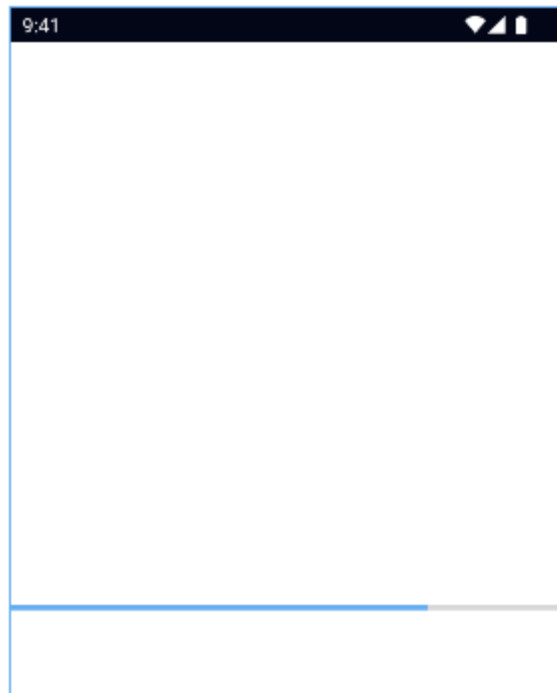
Komponen ListView ini hanya bisa menangani List berbentuk vertical saja, jika Anda mempunyai kebutuhan bentuk List yang lebih Kompleks, seperti List dengan bentuk Grid atau List dengan orientasi Horizontal dan yang lainnya, saya menyarankan Anda untuk menggunakan plugin [@nativescript-community/ui-collectionview](https://github.com/nativescript-community/ui-collectionview).

### 6.3.8 Progress

Progress bukan komponen Loading, melainkan komponen untuk menampilkan Progress dari sebuah proses. Tampilannya berbentuk Bar, dia akan menampilkan Progress pengerjaan yang di akumulasi dalam satuan persen. Misalnya Anda mempunyai total 1000 Task dan yang sudah di kerjakan adalah 750 Task, maka komponen ini akan menampilkan Progress sebesar 75% dalam bentuk Bar.

Komponen ini merupakan abstraksi dari *android.widget.ProgressBar* jika di Android dan *UIProgressView* jika di IOS.

```
<Progress  
  value="750"  
  max="1000"  
  backgroundColor="#fff"  
  color="#000"  
>
```



**Gambar 6.28 Komponen Progress**

Anda dapat memodifikasi Style-Nya menggunakan Inline CSS. Yang seperti Anda lihat pada contoh di atas.

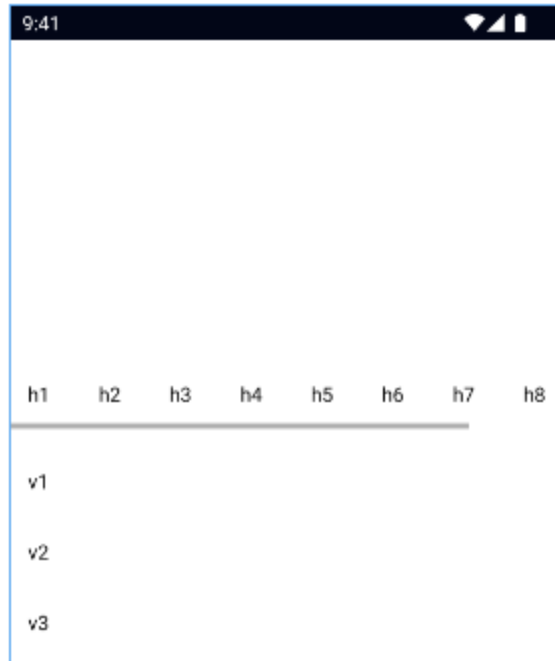
Properti	Tipe	Keterangan
<b>value</b>	number	Nilai saat ini dari Progress
<b>maxValue</b>	number	Nilai maksimal dari Progress ( <i>default : 100</i> )

### 6.3.9 Scroll View

Scroll View berfungsi merender konten agar dapat di Scroll. Khusus untuk komponen ListView dan sejenisnya tidak memerlukan komponen ini, karena di dalam ListView-Nya sendiri sudah Include Scroll. Anda dapat mengatur orientasi dari komponen ini, apakah scroll yang Anda mau berbentuk Vertical atau Horizontal.

Komponen ini merupakan abstraksi dari *android.view* jika di Android dan *UIScrollView* jika di IOS.

```
<ScrollView>
  <StackLayout padding="12">
    <Label text="v1" height="50" />
    <Label text="v2" height="50" />
    <Label text="v3" height="50" />
    <Label text="v4" height="50" />
    <Label text="v5" height="50" />
    <Label text="v6" height="50" />
    <Label text="v7" height="50" />
    <Label text="v8" height="50" />
    <Label text="v9" height="50" />
  </StackLayout>
</ScrollView>
```



**Gambar 6.29 Komponen ScrollView**

Properti	Tipe	Keterangan
<b>orientation</b>	enum	Untuk mengatur orientasi scroll, horizontal atau vertical ( <i>default : vertical</i> )
<b>scrollBarIndicatorVisible</b>	boolean	Untuk mengatur visibilitas ScrollBar ( <i>default : true</i> )
<b>isScrollEnabled</b>	boolean	Untuk Mengaktifkan atau menonaktifkan Scroll
<b>verticalOffset</b>	number	Mendapatkan offset vertikal dari konten yang di-scroll
<b>horizontalOffset</b>	number	Mendapatkan offset horizontal dari konten yang di-scroll.
<b>scrollableHeight</b>	number	Mendapatkan ketinggian maksimum yang dapat di-scroll, ini juga merupakan nilai maksimum untuk <i>verticalOffset</i> .
<b>scrollableWidth</b>	number	Mendapatkan lebar maksimum yang dapat di-scroll, ini juga merupakan nilai maksimum untuk <i>horizontalOffset</i> .
<b>scroll</b>	event	Mendapatkan nilai terkini dari scroll, Y adalah data dari Vertical Scroll dan X adalah data dari Horizontal Scroll.

### 6.3.10 Search Bar

Search Bar menyediakan tampilan untuk memasukkan Query pencarian. Komponen ini merupakan abstraksi dari *android.widget.SearchView* jika di Android dan *UISearchBar* jika di IOS.

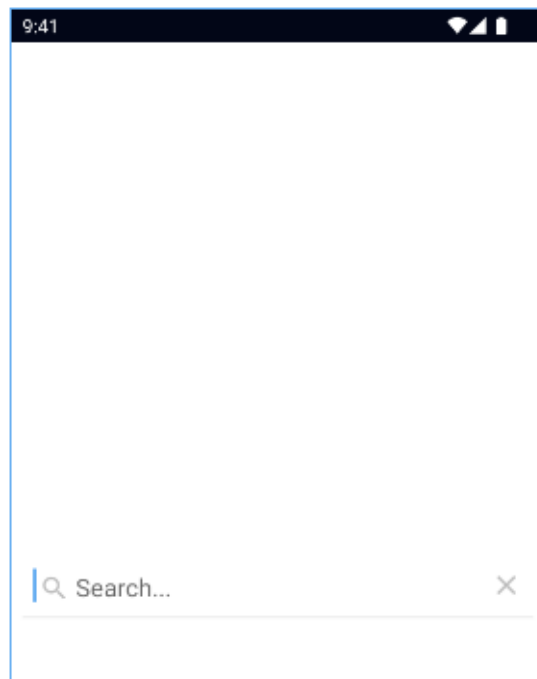
```
<!-- Kode XML-Nya -->
```

```
<SearchBar hint="Search..." submit="onSubmit" clear="onClear" />
```

```
/* Kode Javascript-Nya */
```

```
export function onSubmit(args) {  
  const queryText = args.object.text;  
  console.log(queryText);  
}
```

```
export function onClear() {  
  console.log('Clear qeuryText');  
}
```



**Gambar 6.30 Komponen SearchBar**

Properti	Tipe	Keterangan
<b>hint</b>	string	Placeholder teks
<b>text</b>	string	Search query teks
<b>textFieldBackgroundColor</b>	string	Mengatur warna latar belakang area input.
<b>textFieldHintColor</b>	string	Mengatur warna teks placeholder.
<b>submit</b>	event	Mendapatkan query teks setelah pengguna melakukan submit
<b>clear</b>	event	Menghapus query teks

### 6.3.11 Segmented Bar

Komponen ini berbentuk tombol namun membentuk satu kelompok tombol yang dapat di Select, fungsinya mirip seperti Komponen Radio di HTML, saat Anda men-select salah satu tombol maka tombol yang lainnya akan ter-unselect. Komponen ini merupakan abstraksi dari *android.widget.TabHost* jika di Android dan *UISegmentedControl* jika di IOS.

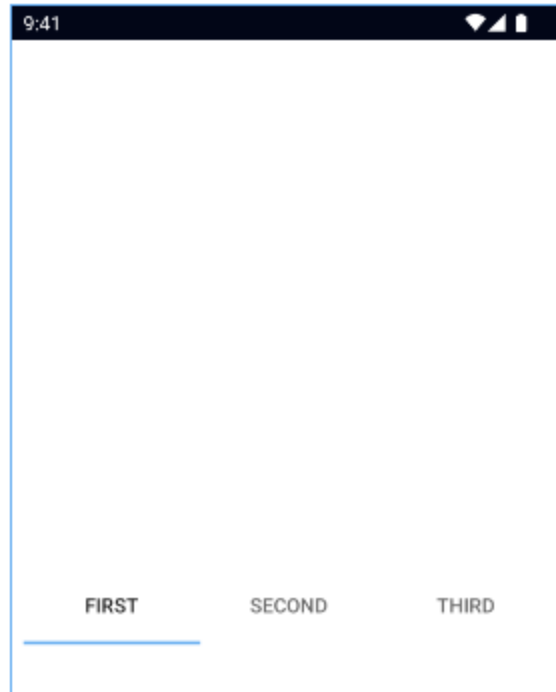
*<!-- Kode XML-Nya -->*

```
<SegmentedBar
  selectedIndex="2"
  selectedBackgroundColor="#C8E6C9"
  selectedIndexChanged="SegmentedBarChanged">
  <SegmentedBarItem title="First" />
  <SegmentedBarItem title="Second" />
  <SegmentedBarItem title="Third" />
</SegmentedBar>
```

*/\* Kode Javascript-Nya \*/*

```
export function SegmentedBarChanged(args) {
  console.log("SegmentedBarChanged", args.object.selectedIndex);
}
```

Method *selectedIndexChanged* akan mengembalikan informasi Segmented Item mana yang sedang dipilih dalam bentuk Indeks yang berupa angka.



**Gambar 6.31** *Komponen SegmentedBar*

Berikut adalah properti dari *SegmentedBar*.

Properti	Tipe	Keterangan
<b>selectedIndex</b>	number	Mendapatkan atau menetapkan indeks item yang dipilih.
<b>selectedBackgroundColor</b>	string	Berisi nama atau kode warna untuk mengatur warna dari latar belakang Segmented Item yang dipilih
<b>selectedIndexChanged</b>	event	Berfungsi untuk mengembalikan nilai indeks dari Segmented item yang sedang dipilih

Berikut adalah properti dari *SegmentedBarItem*.

Properti	Tipe	Keterangan
<b>title</b>	string	Mengatur nama dari Segmented Bar Item

### 6.3.12 Slider

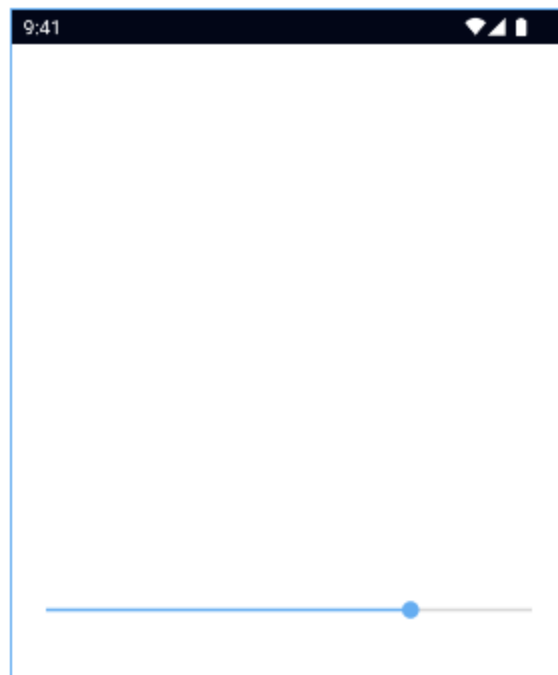
Komponen Slider berfungsi untuk memilih angka dari rentang angka tertentu. Untuk memilih suatu angka, pengguna harus menggeser Slider Kontrol-Nya ke kanan atau ke kiri. Komponen ini merupakan abstraksi dari *android.widget.SeekBar* jika di Android dan *UISlider* jika di IOS.

```
<!-- Kode XML-Nya -->
```

```
<Slider value="75" minValue="0" maxValue="100" valueChange="sliderChanged" />
```

```
/* Kode Javascript-Nya */
```

```
export function sliderChanged(args) {  
  console.log('Slider value changed to', args.value);  
}
```



**Gambar 6.32 Komponen Slider**

Properti	Tipe	Keterangan
<b>value</b>	number	Mendapatkan atau menetapkan nilai penggeser yang dipilih saat ini.
<b>minValue</b>	number	Mendapatkan atau menetapkan nilai minimum penggeser.
<b>maxValue</b>	number	Mendapatkan atau menetapkan nilai maksimum penggeser.

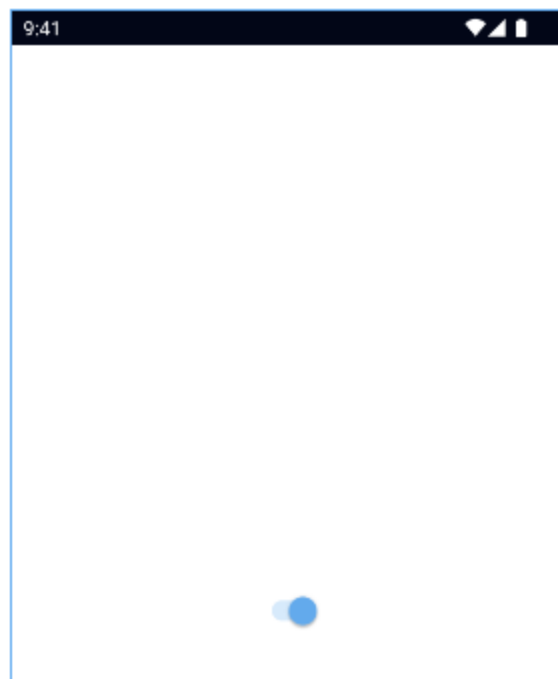


<b>valueChange</b>	event	Mendapatkan nilai terkini saat Slider mengalami perubahan
--------------------	-------	---

### 6.3.13 Switch

Komponen ini mirip seperti Checkbox di HTML, dia hanya memiliki 2 tingkalakku yaitu Checked dan Unchecked, namun pada komponen Switch ini tingkalakku-Nya yaitu Aktif atau Tidak Aktif, seperti saklar lampu. Komponen ini merupakan abstraksi dari *android.widget.Switch* jika di Android dan *UISwitch* jika di IOS.

```
<Switch checked="true"
  color="#BFCDAC"
  backgroundColor="green"
  offBackgroundColor="#DC493D"
/>
```



**Gambar 6.33** *Komponen Switch*

Properti	Tipe	Keterangan
<b>checked</b>	boolean	Mengatur apakah Switch-nya aktif atau tidak
<b>offBackgroundColor</b>	string	Untuk mengatur latar belakang Switch saat posisi Off

<b>checkedChange</b>	event	Mendapatkan nilai terkini saat Switch mengalami perubahan
----------------------	-------	---

### 6.3.14 Tab View

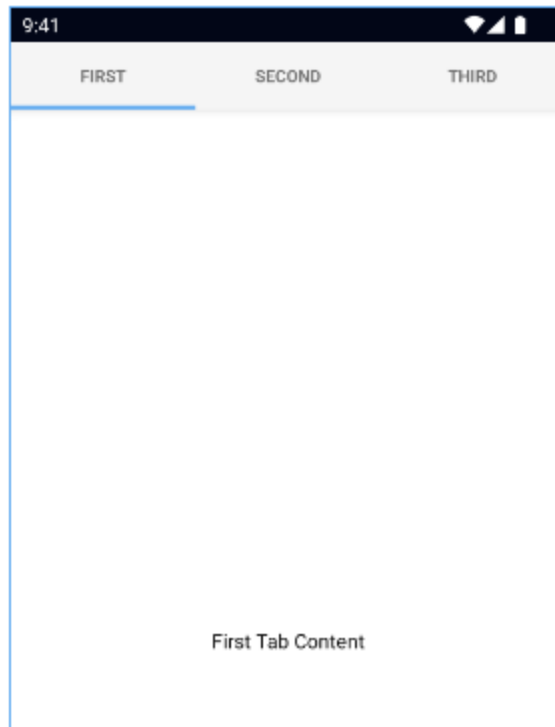
Tab View adalah komponen yang menampilkan konten yang sudah dikelompokkan dalam bentuk Tab, memungkinkan pengguna beralih antara tab-tab tersebut. Komponen ini merupakan abstraksi dari *androidx.viewpager.widget.ViewPager* jika di Android dan *UITabBarController* jika di IOS.

*<!-- Kode XML-Nya -->*

```
<TabView selectedIndexChanged="tabChanged">
  <TabViewItem title="First">
    <Label
      text="First Tab Content"
      textAlignment="center"
      verticalAlignment="center" />
  </TabViewItem>
  <TabViewItem title="Second">
    <Label
      text="Second Tab Content"
      textAlignment="center"
      verticalAlignment="center" />
  </TabViewItem>
  <TabViewItem title="Third">
    <Label
      text="Third Tab Content"
      textAlignment="center"
      verticalAlignment="center" />
  </TabViewItem>
</TabView>
```

*/\* Kode Javascript-Nya \*/*

```
export function tabChanged(args){
  const tabView = args.object
  console.log('TabView selectedIndex:', tabView.selectedIndex)
}
```



**Gambar 6.34** *Komponen TabView*

Berikut adalah properti dari *TabView*.

Properti	Tipe	Keterangan
<b>selectedIndex</b>	number	Mendapatkan atau menyetel tab yang dipilih saat ini.
<b>tabTextColor</b>	string	Mengatur warna teks pada judul tab.
<b>tabTextFontSize</b>	number	Mengatur ukuran huruf pada judul tab
<b>tabBackgroundColor</b>	string	Mengatur latar belakang dari tab
<b>androidSelectedTabHighlightColor</b>	string	[Android Only] Mengatur warna Highlight pada tab
<b>androidTabsPosition</b>	enum	[Android Only] Mengatur posisi tab. top (default) atau bottom.
<b>iosIconRenderingMode</b>	enum	[IOS Only] Mengatur mode rendering icon. automatic (default), alwaysOriginal atau alwaysTemplate.
<b>selectedIndexChanged</b>	event	Mendapatkan informasi terkini dari posisi Tab saat Tab mengalami perpindahan.

Berikut adalah properti dari *TabViewItem*.

Properti	Tipe	Keterangan
<b>title</b>	string	Mengatur nama dari item tab
<b>textTransform</b>	enum	initial (default), none, capitalize, uppercase atau lowercase
<b>iconSource</b>	string	Mengatur source icon. Mendukung path res://, font:// atau path absolut (contoh: ~/assets/image.png), untuk path-nya sama persis seperti yang ada di komponen image.

### 6.3.15 Text Field

Text Field adalah komponen Input satu baris, mirip seperti *Input Text* yang ada di HTML. Komponen ini merupakan abstraksi dari *android.widget.EditText* jika di Android dan *UITextField* jika di IOS.

```
<TextField hint="Placeholder..." />
```



**Gambar 6.35** *Komponen TextField*

Anda juga bisa menata teks dengan style yang berbeda-beda pada TextField secara lebih spesifik dengan menggunakan FormattedString dan Span, kurang lebih seperti ini contohnya.

```
<TextField>
  <FormattedString>
    <Span text="Kunjungi blog " fontWeight="bold" />
    <Span text="https://kang-cahya.com " fontStyle="italic" style="color: green" />
  </FormattedString>
</TextField>
```

Properti	Tipe	Keterangan
<b>text</b>	string	Mendapatkan atau menyetel teks pada TextField
<b>hint</b>	string	Mengatur placeholder
<b>editable</b>	boolean	Mengatur mode TextField apakah read-only atau dapat di edit (default: true)
<b>keyboardType</b>	enum	Mengatur tipe keyboard untuk TextField. <i>datetime, email, integer, number, phone atau url</i>
<b>returnKeyType</b>	enum	Mengatur return key keyboard pada TextField. <i>done, go, next, search atau send</i>
<b>isEnabled</b>	boolean	Mengatur apakah TextField aktif atau tidak. (default: true)
<b>maxLength</b>	number	Mengatur batas karakter yang dapat di Input pada TextField
<b>secure</b>	boolean	Digunakan untuk Input Password, set True jika inputan berupa password. (default: false)
<b>secureWithoutAutofill</b>	boolean	[iOS Only] digunakan untuk input password tanpa autofill. (default: false)
<b>autocapitalizationType</b>	enum	Huruf otomatis kapital saat teks di inputkan. <i>allCharacters, none, sentences atau words</i>
<b>textChange</b>	event	Mendapatkan nilai terkini saat teks yang di inputkan mengalami perubahan
<b>returnPress</b>	event	Mendapatkan nilai terkini saat return key pada keyboard di tekan.
<b>focus</b>	event	Mendeteksi apakah TextField sudah fokus. Pada posisi ini biasanya pengguna sudah siap mengetik pada TextField.

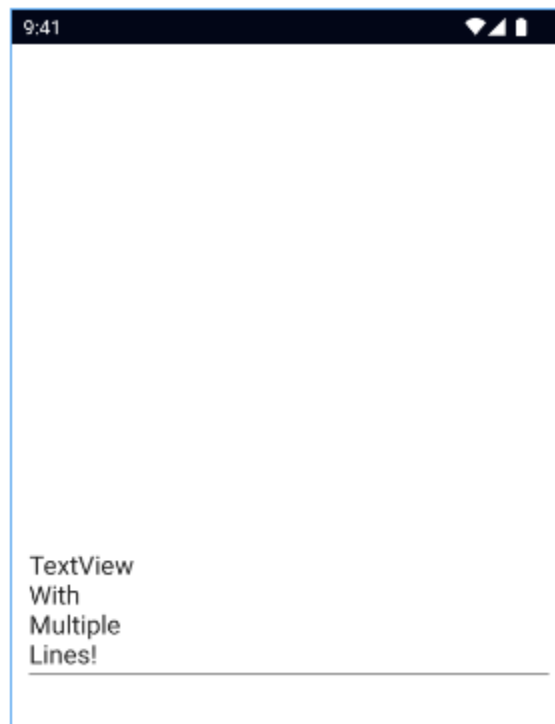
<b>blur</b>	event	Mendeteksi apakah TextField sudah tidak fokus. Pada posisi ini biasanya pengguna sudah meninggalkan TextField.
-------------	-------	--

### 6.3.16 Text View

Text View adalah komponen input multi-line, mirip seperti Textarea yang ada di HTML. Komponen ini hampir sama seperti TextField, perbedaannya terletak pada jumlah baris yang dapat di tampung. Komponen ini merupakan abstraksi dari *android.widget.EditText* jika di Android dan *UITextView* jika di IOS.

```
<TextView text="{{ text }}" />
```

```
const text = `TextView\nWith\nMultiple\nLines!`
```



**Gambar 6.36** *Komponen TextView*

Anda juga bisa menata teks dengan style yang berbeda-beda pada TextView secara lebih spesifik dengan menggunakan FormattedString dan Span, kurang lebih seperti ini contohnya.

```
<TextView>
  <FormattedString>
    <Span text="Kunjungi blog " fontWeight="bold" />
    <Span text="https://kang-cahya.com " fontStyle="italic" style="color: green" />
  </FormattedString>
</TextView>
```

Properti	Tipe	Keterangan
<b>text</b>	string	Mendapatkan atau menyetel teks pada TextView
<b>hint</b>	string	Mengatur placeholder
<b>editable</b>	boolean	Mengatur mode TextView apakah read-only atau dapat di edit (default: true)
<b>keyboardType</b>	enum	Mengatur tipe keyboard untuk TextView. <i>datetime, email, integer, number, phone atau url</i>
<b>returnKeyType</b>	enum	Mengatur return key keyboard pada TextView. <i>done, go, next, search atau send</i>
<b>isEnabled</b>	boolean	Mengatur apakah TextView aktif atau tidak. (default: true)
<b>maxLines</b>	number	Mengatur batas baris yang dapat di Input pada TextView
<b>autocorrect</b>	boolean	Mengatur apakah koreksi otomatis aktif atau tidak
<b>textChange</b>	event	Mendapatkan nilai terkini saat teks yang di inputkan mengalami perubahan
<b>returnPress</b>	event	Mendapatkan nilai terkini saat return key pada keyboard di tekan.
<b>focus</b>	event	Mendeteksi apakah TextView sudah fokus. Pada posisi ini biasanya pengguna sudah siap mengetik pada TextView.
<b>blur</b>	event	Mendeteksi apakah TextView sudah tidak fokus. Pada posisi ini biasanya pengguna sudah meninggalkan TextView.

### 6.3.17 Web View

Web View adalah komponen yang digunakan untuk merender konten web. Kontennya bisa berupa URL Web, File HTML lokal atau String HTML. Komponen ini merupakan abstraksi dari *android.webkit.WebView* jika di Android dan *WKWebView* jika di IOS.

1. Konten berasal dari URL Web

```
<WebView src="https://www.kang-cahya.com/" />
```

---

*Untuk konten URL Web direkomendasikan URL-Nya menggunakan https, jika menggunakan http ada kemungkinan akan terblokir*

---

2. Konten berasal dari File HTML Lokal, pada contoh di bawah file HTML terletak di *app/assets/html/indeks.html*

```
<WebView src="~/assets/html/index.html" />
```

3. Konten berupa String HTML langsung yang di tampung pada sebuah variabel

```
<WebView src="{{ stringHtml }}" />
```

```
const stringHtml = `<div><h1>Kode HTML Statis</h1></div>`;
```

Properti	Tipe	Keterangan
<b>src</b>	string	Tempat mendefinisikan sumber yang akan dimuat di WebView. Nilainya bisa berupa URL HTTP yang valid, Path ke file HTML lokal, atau string HTML Statis.



## 6.4 Dialogs

Komponen ini sangat menarik perhatian pengguna, karena dapat tiba-tiba muncul di sebuah halaman dalam bentuk Popup berbentuk kotak. Komponen ini abstraksi dari *android.app.AlertDialog.Builder* jika di Android dan *UIAlertController* jika di IOS.

NativeScript menawarkan berbagai Dialog, yang dapat Anda gunakan dan tersedia melalui impor Dialog dari *@nativescript/core*.

```
import { Dialogs } from '@nativescript/core'

Dialogs.alert(options)
Dialogs.action(options)
Dialogs.confirm(options)
Dialogs.prompt(options)
Dialogs.login(options)
```

Selain itu Anda juga dapat mengaksesnya secara global seperti ini.

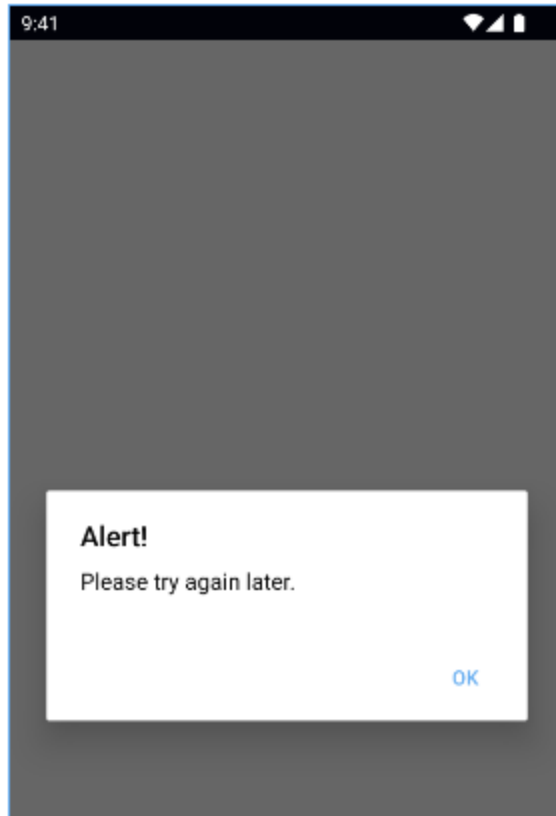
```
alert(options)
action(options)
confirm(options)
prompt(options)
login(options)
```

### 6.4.1 Alert

Dialog ini biasanya digunakan untuk memperingatkan pengguna.

```
import { Dialogs } from "@nativescript/core";

Dialogs.alert({
  title: "Alert!",
  message: "Please try again later.",
  okButtonText: "OK",
  cancelable: true,
}).then(() => {
  console.log("Dialog alert closed.");
});
```



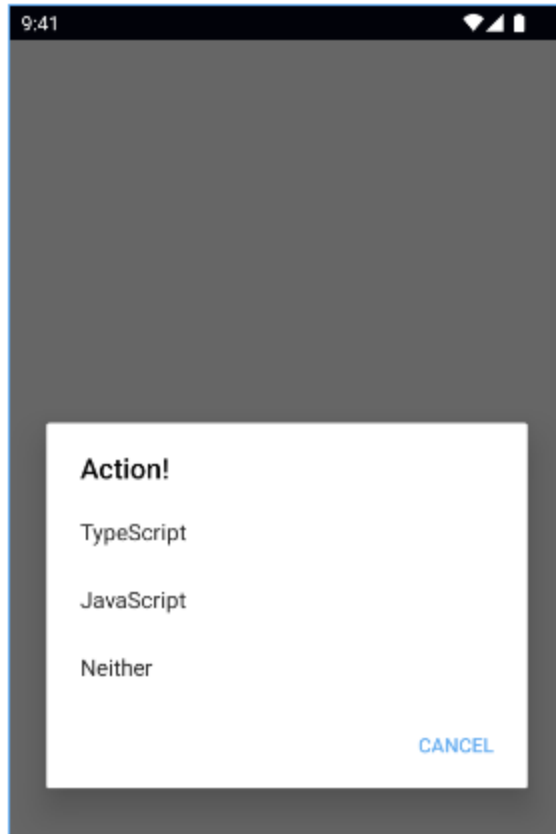
**Gambar 6.37 Dialog Alert**

### 6.4.2 Action

Dialog ini di gunakan untuk meminta pengguna memilih. Pada tiap pilihan Anda dapat menyisipkan fungsi untuk melakukan sebuah Action seperti menghapus data, mengedit atau yang lainnya.

```
import { Dialogs } from "@nativescript/core";

Dialogs.action({
  title: "Action!",
  message: "Choose your language:",
  cancelButtonText: "Cancel",
  actions: ["TypeScript", "JavaScript", "Neither"],
  cancelable: true,
}).then((result) => {
  console.log(result);
});
```



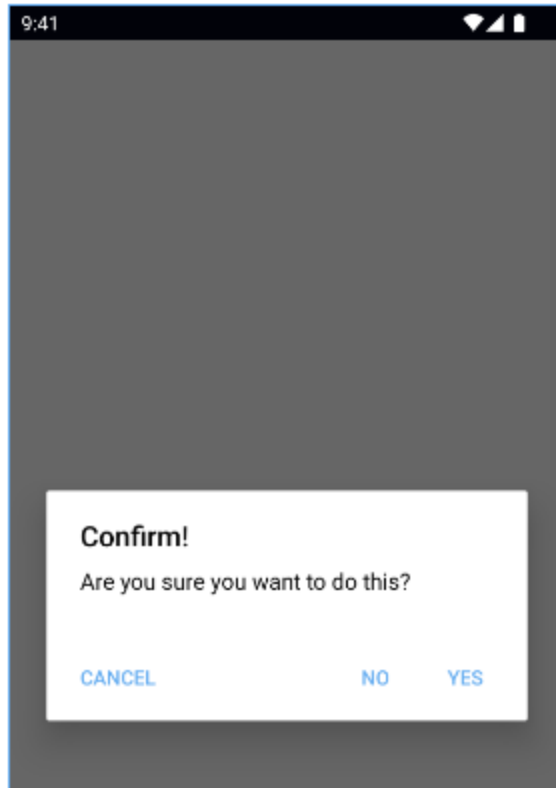
**Gambar 6.38 Dialog Action**

### 6.4.3 Confirm

Dialog ini biasanya digunakan untuk meminta konfirmasi pada pengguna. Bentuk konfirmasinya hanya sebatas Ya atau Tidak.

```
import { Dialogs } from "@nativescript/core";

Dialogs.confirm({
  title: "Confirm!",
  message: "Are you sure you want to do this?",
  okButtonText: "Yes",
  cancelButtonText: "No",
  neutralButtonText: "Cancel",
}).then((result) => {
  console.log(result);
});
```



**Gambar 6.39 Dialog Confirm**

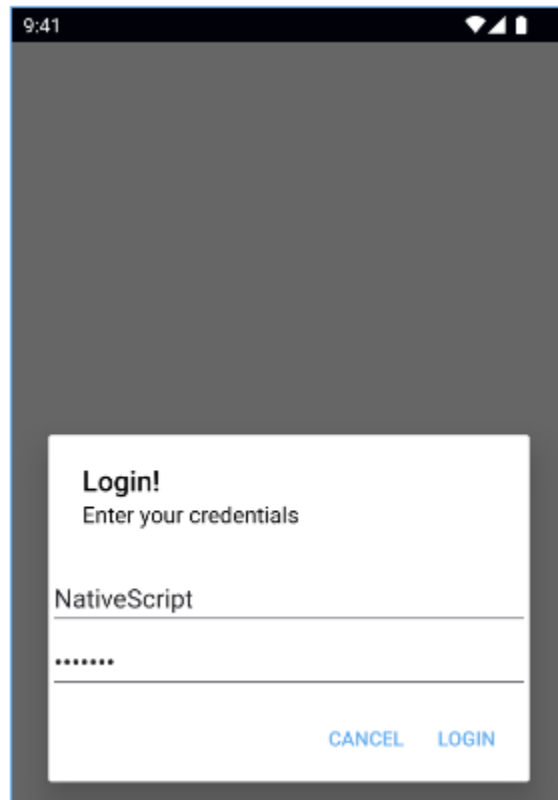
#### **6.4.4 Prompt**

Dialog ini digunakan untuk meminta pengguna memasukkan sebuah input. Tipe inputan-nya dapat Anda atur sesuai kebutuhan. Untuk pilihan tipe input yang ada saat ini adalah *email*, *decimal*, *phone*, *number*, *text*, *password*.

```
import { Dialogs } from "@nativescript/core";

Dialogs.prompt({
  title: "Prompt!",
  message: "Enter the name of this framework:",
  defaultText: "NativeScript",
  okButtonText: "OK",
  neutralButtonText: "Cancel",
  inputType: 'email', // email, decimal, phone, number, text, password
  cancelable: true,
  cancelButtonText: 'Cancel',
  capitalizationType: 'none', // none, allCharacters, sentences, or words
}).then((result) => {
  console.log(result);
});
```

```
});
```



**Gambar 6.40 Dialog Prompt**

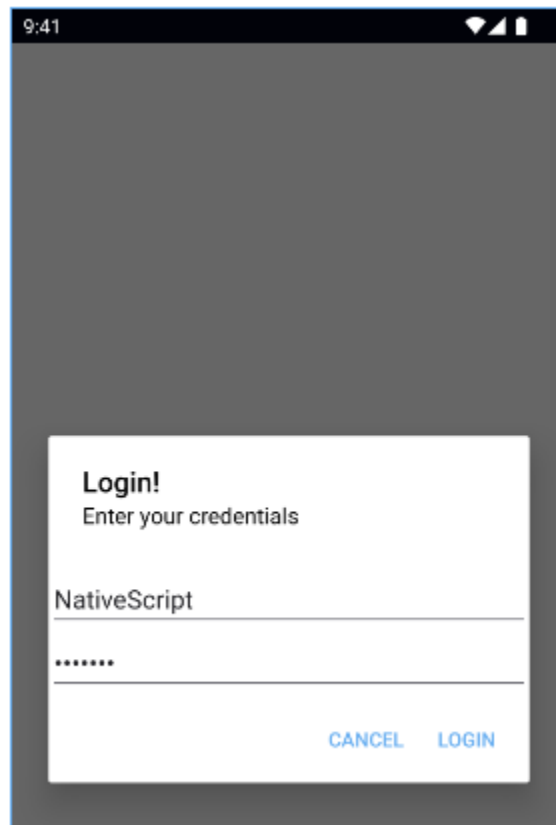
### 6.4.5 Login

Dialog ini biasanya digunakan untuk meminta kredensial pengguna.

```
import { Dialogs } from "@nativescript/core";

Dialogs.login({
  title: "Login!",
  message: "Enter your credentials",
  okButtonText: "Login",
  cancelButtonText: "Cancel",
  userName: "NativeScript",
  password: "hunter2",
  // neutralButtonText: 'Neutral',
  // cancelable: true,
  // passwordHint: 'Your password',
});
```

```
// userNameHint: 'Your username',
}).then((result) => {
  console.log(result);
});
```



**Gambar 6.41 Dialog Login**

---

*Anda dapat melakukan Styling pada komponen Dialog (Alert, Action, Confirm, Prompt, Login). Namun cara Styling-Nya tidak menggunakan CSS, karena komponen Dialog sangat berbeda dari komponen lainnya, ia tidak memiliki wujud di XML, hanya bisa di atur melalui Javascript.*

---

Untuk melakukan styling pada komponen Dialog, Anda bisa menggunakan styles.xml yang ada di `App_Resources\Android\src\main\res\values\styles.xml`

1. Tambahkan dahulu kode baru berikut ini pada styles.xml. Kode di bawah akan mencoba mengatur Style sederhana dengan mengganti warna Background menjadi Yellow dan Text-Nya berwarna hitam.

```
<style
  name="CustomAlertDialogTheme"
  parent="Theme.AppCompat.Light.Dialog.Alert">
  <!-- Example: Black text color -->
  <item name="android:textColorPrimary">#000000</item>
  <!-- Example: Yellow background -->
  <item name="android:background">#fdd835</item>
</style>
```

2. Kemudian pada Javascript tambahkan kode berikut ini, lalu panggil name style (*CustomAlertDialogTheme*) yang sebelumnya Anda sudah buat.

```
import { Application } from "@nativescript/core";

const applicationContext = Application.android.context;
const themeld = applicationContext
  .getResources()
  .getIdentifier(
    "CustomAlertDialogTheme",
    "style",
    applicationContext.getPackageName()
  );
```

3. Terakhir tambahkan properti *theme* pada komponen Dialog Anda. Di sini saya akan coba contohkan pada komponen *Dialog Alert*, kurang lebih kode lengkap Javascript-Nya seperti ini.

```
import { Application, Dialogs } from "@nativescript/core";

export function showAlert() {
  const applicationContext = Application.android.context;
  const themeld = applicationContext
    .getResources()
    .getIdentifier(
      "CustomAlertDialogTheme",
      "style",
```

```
        applicationContext.getPackageName()
    );

    Dialogs.alert({
        title: "Alert!",
        message: "Please try again later.",
        okButtonText: "OK",
        cancelable: true,
        theme: themeld,
    }).then(() => {
        console.log("Dialog alert closed.");
    });
}
```

Dan saat Anda coba jalankan, warna Background pada Dialog Alert akan berubah warna-Nya sesuai warna yang telah atur sebelumnya.

File *styles.xml* yang ada di dalam *App\_Resources\Android* saya menyebutnya dengan *R.style*, Untuk mengetahui style apa saja yang dapat Anda atur di *R.style*, Anda dapat baca lengkapnya di sini : <https://developer.android.com/reference/android/R.style>.