

Total Tampilan Seluruh Kode Modul Library

File custom_row.dart pada folder model dari folder invoice

```
class CustomRow {
    final String itemName;
    final String itemPrice;
    final String amount;
    final String subTotalProduct;

    CustomRow(
        this.itemName,
        this.itemPrice,
        this.amount,
        this.subTotalProduct);
}
```

File product pada folder model dari folder invoice

```
class Product {
    final String name;
    final double price;
    final int ppnInPercent = 11;
    int amount = 0;

    Product(
        this.name,
        this.price);
}
```

File get_total_invoice.dart pada folder invoice

```
import 'package:gd8_library_xxxx/invoice/model/product.dart';

String getSubTotal(List<Product> products) {
    return products
        .fold(0.0,
            (double prev, element) => prev + (element.amount * element.price))
        .toStringAsFixed(2);
}

String getPPNTotal(List<Product> products) {
    return products
        .fold(
```

```

    0.0,
    (double prev, next) =>
        prev + ((next.price / 100 * next.ppnInPercent) * next.amount),
    )
    .toStringAsFixed(2);
}

```

File item_doc.dart di folder invoice

```

import 'package:gd8_library_xxxx/invoice/model/custom_row_invoice.dart';
import 'package:pdf/pdf.dart';
import 'package:pdf/widgets.dart' as pw;
import 'package:responsive_sizer/responsive_sizer.dart';

pw.Container itemColumn(List<CustomRow> elements) {
    return pw.Container(
        height: 180,
        child: pw.Column(
            children: [
                for (var element in elements)
                    pw.Row(
                        children: [
                            pw.Expanded(
                                child:
                                    pw.Text(element.itemName, textAlign:
pw.TextAlign.left)),
                            pw.Expanded(
                                child: pw.Text(element.itemPrice,
                                    textAlign: pw.TextAlign.right)),
                            pw.Expanded(
                                child:
                                    pw.Text(element.amount, textAlign: pw.TextAlign.right)),
                            pw.Expanded(
                                child: pw.Text(element.subTotalProduct, textAlign:
pw.TextAlign.right)),
                        ],
                    ),
            ],
        );
}

```

File home_page.dart

```
import 'package:flutter/foundation.dart';
```

```
import 'package:flutter/material.dart';

import 'package:gd8_library_xxxx/invoice/model/product.dart';

import 'dart:io';
import 'package:image_picker/image_picker.dart';
import 'package:flutter/services.dart';

import 'package:responsive_sizer/responsive_sizer.dart';
import 'package:uuid/uuid.dart';

import 'package:gd8_library_xxxx/pdf_view.dart';

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  final nameController = TextEditingController();
  final birthdayController = TextEditingController();
  final ageController = TextEditingController();
  final phoneController = TextEditingController();
  final addressController = TextEditingController();
  String id = const Uuid().v1(); // untuk membuat kode unik berdasarkan waktu
  File? image;
  List<Product> products = [
    Product("Membership", 50000),
    Product("Books", 30000),
    Product("Pen", 5000),
    Product("Snacks", 10000),
    Product("Hamburger", 25000),
  ];
  int number = 0;
  getTotal() => products
    .fold(0.0,
        (double prev, element) => prev + (element.price * element.amount))
    .toStringAsFixed(2);

  getPPN() => products
    .fold(
      0.0,
      (double prev, element) =>
        prev +
        (element.price / 100 * element.ppnInPercent * element.amount))
    .toStringAsFixed(2);
}
```

```
// untuk input pemilihan gambar dari galeri
Future pickImage() async {
    try {
        final image = await ImagePicker().pickImage(source:
ImageSource.gallery);

        if (image == null) return;

        final imageTemp = File(image.path);
        setState(() => this.image = imageTemp);
    } on PlatformException catch (e) {
        debugPrint(
            'Failed to pick image: $e'); //untuk cek saat di log dan sebaiknya
dihapus ketika deploy app, namun dapat ditambah dengan toast
    }
}

//untuk input pemilihan gambar dari kamera
Future pickImageC() async {
    try {
        final image = await ImagePicker().pickImage(source: ImageSource.camera);

        if (image == null) return;
        final imageTemp = File(image.path);
        setState(() => this.image = imageTemp);
    } on PlatformException catch (e) {
        debugPrint(
            'Failed to pick image: $e'); //untuk cek saat di log dan sebaiknya
dihapus ketika deploy app, namun dapat ditambah dengan toast
    }
}

//tampilan input berbentuk form yang terdiri dari:
// -button,
// -preview gambar yang dipilih,
// -textfield
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            centerTitle: true,
            backgroundColor: Theme.of(context).colorScheme.inversePrimary,
            title: Text(
                'Modul 8 Library',
                style: TextStyle(
                    fontFamily: 'Poppins',
                    fontSize: 17.sp,
```

```
        fontWeight: FontWeight.bold),
    ),
),
body: SingleChildScrollView(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
            //input identitas
            Container(
                margin: EdgeInsets.all(3.h),
                child: Column(children: [
                    Center(
                        child: Text(
                            'Fill This Form',
                            style: TextStyle(
                                fontFamily: 'Raleway',
                                fontWeight: FontWeight.bold,
                                fontSize: 15.sp,
                            ),
                        ),
                    ),
                ]),
            ),
            Column(
                mainAxisAlignment: MainAxisAlignment
                    .center, // Center the buttons horizontally
                children: [
                    Padding(
                        padding:
                            EdgeInsets.symmetric(horizontal: 2.h, vertical:
2.h),
                        child: ElevatedButton(
                            style: ElevatedButton.styleFrom(
                                backgroundColor: Colors.amber,
                                textStyle: TextStyle(
                                    fontWeight: FontWeight.bold,
                                    fontSize: 15.sp,
                                ),
                            ),
                            onPressed: () {
                                pickImageC();
                            },
                            child: const Text("Pick Image from Camera"),
                        ),
                    ),
                    Padding(
                        padding:
                            EdgeInsets.symmetric(horizontal: 2.h, vertical:
2.h),
                    ),
                ],
            ),
        ],
    ),
);
```

```
        child: ElevatedButton(
            style: ElevatedButton.styleFrom(
                padding: EdgeInsets.symmetric(horizontal: 2.h),
                backgroundColor: Colors.amber,
                textStyle: TextStyle(
                    fontWeight: FontWeight.bold,
                    fontSize: 15.sp,
                ),
            ),
            onPressed: () {
                pickImage();
            },
            child: const Text("Pick Image from Gallery"),
        ),
    ),
    SizedBox(
        width: 10.w), // Add some spacing between the buttons
],
),
SizedBox(
    height: 2.h,
),
if (image != null)
Column(
    children: [
        //jika pada web menggunakan Image.network, jika selain
        //web menggunakan Image.file
        kIsWeb ? Image.network(image!.path) :
Image.file(image!),
        Padding(
            padding: EdgeInsets.symmetric(horizontal: 2.h),
            child: IconButton(
                color: Colors.red,
                iconSize: 5.h,
                icon: const Icon(Icons.delete),
                onPressed: () {
                    setState(() {
                        image = null;
                    });
                },
            ),
        ),
    ],
)
else
    const Text("No image selected"),
    TextFormField(
        controller: nameController,
```

```
        decoration: const InputDecoration(
            hintText: 'Enter Your Name', labelText: 'Name'),
        ),
        TextFormField(
            controller: birthdayController,
            decoration: const InputDecoration(
                hintText: 'Enter Your Birthday', labelText: 'Birthday'),
            keyboardType: TextInputType.datetime,
        ),
        TextFormField(
            controller: ageController,
            decoration: const InputDecoration(
                hintText: 'Enter Your Age', labelText: 'Age'),
            keyboardType: TextInputType.number,
        ),
        TextFormField(
            controller: phoneController,
            decoration: const InputDecoration(
                hintText: 'Enter Your Phone Number',
                labelText: 'Phone Number'),
            keyboardType: TextInputType.number,
        ),
        TextFormField(
            controller: addressController,
            decoration: const InputDecoration(
                hintText: 'Enter Your Address', labelText: 'Address'),
        ),
    ],
),
SizedBox(height: 5.h),  
  
//Input Product  
Center(  
    child: Text(  
        'Buy Product',  
        style: TextStyle(  
            fontFamily: 'Raleway',  
            fontWeight: FontWeight.bold,  
            fontSize: 15.sp,
        ),
    ),
),
SizedBox(height: 2.h),
Padding(  
    padding: EdgeInsets.symmetric(horizontal: 2.h),
    child: Column(  
        mainAxisAlignment: CrossAxisAlignment.center,  
        children: [

```

```
SizedBox(
    height: 180,
    child: ListView.builder(
        shrinkWrap: true,
        itemBuilder: (context, index) {
            final currentProduct = products[index];
            return Row(
                children: [
                    Expanded(child: Text(currentProduct.name)),
                    Expanded(
                        child:
                            Text("Price: Rp
${currentProduct.price.toStringAsFixed(2)}")),
                    Expanded(
                        child: Row(
                            children: [
                                Expanded(
                                    child: IconButton(
                                        onPressed: () {
                                            setState(() =>
currentProduct.amount++);
                                        },
                                        icon: const Icon(Icons.add),
                                    ),
                                ),
                                Expanded(
                                    child: Text(
                                        currentProduct.amount.toString(),
                                        textAlign: TextAlign.center,
                                    ),
                                ),
                                Expanded(
                                    child: IconButton(
                                        onPressed: () {
                                            setState(() => currentProduct.amount--;
);
                                        },
                                        icon: const Icon(Icons.remove),
                                    ),
                                )
                            ],
                        ),
                    );
                ],
                itemCount: products.length,
            ),
        ),
    ),
)
```

```
        ),
        Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [const Text("PPN"), Text("Rp ${getPPN()}")],
        ),
        Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [const Text("Total"), Text("Rp ${getTotal()}")
        €")],
    ),
),
),
),
),
),
Container(
    margin: EdgeInsets.symmetric(vertical: 2.h),
    child: ElevatedButton(
        onPressed: () {
            if (image == null ||
                nameController.text.isEmpty ||
                birthdayController.text.isEmpty ||
                ageController.text.isEmpty ||
                phoneController.text.isEmpty ||
                addressController.text.isEmpty) {
                showDialog(
                    context: context,
                    builder: (context) => AlertDialog(
                        title: const Text('Warning'),
                        content: const Text(
                            'Please fill in all the fields.'),
                        actions: [
                            TextButton(
                                onPressed: () => Navigator.pop(context),
                                child: const Text('OK'),
                            ),
                        ],
                    );
                );
                return; // Stop PDF creation if fields are empty
            } else {
                createPdf(nameController,
                birthdayController, ageController, phoneController, addressController, id,
                image!, context, products);
                setState(() {
                    const uuid = Uuid();
                    id = uuid.v1();
                });
            }
        },
    ),
),
```

```
        style: ElevatedButton.styleFrom(
            backgroundColor: Colors.amber,
            textStyle: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 15.sp,
            ),
        ),
    ),
),
child: const Text('Create PDF'),
),
),
],
),
),
);
}
}
```

File main.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:gd8_library_xxxx/home_page.dart';
import 'package:responsive_sizer/responsive_sizer.dart';

void main() {
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    // This widget is the root of your application.
    @override
    Widget build(BuildContext context) {
        return ResponsiveSizer(builder: (context, orientation, deviceType)
    {
        Device.orientation == Orientation.portrait
            ? Container( // Widget for Portrait
                width: 100.w,
                height: 20.5.h,
            )
            : Container( // Widget for Landscape
                width: 100.w,
                height: 12.5.h,
            );
        Device.screenType == ScreenType.tablet
    });
}
```

```

        ? Container( // Widget for Tablet
        width: 100.w,
        height: 20.5.h,
    )
        : Container( // Widget for Mobile
        width: 100.w,
        height: 12.5.h,
    );
)
return MaterialApp(
    title: 'Flutter Demo',
    debugShowCheckedModeBanner: false,
    theme: ThemeData(
        fontFamily: 'Poppins',
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.amber)
            .copyWith(
                secondary: Colors.amberAccent,
            ),
        materialTapTargetSize: MaterialTapTargetSize.shrinkWrap,
        appBarTheme: const AppBarTheme(
            systemOverlayStyle: SystemUiOverlayStyle.dark,
        ),
        useMaterial3: true,
    ),
    home: const MyHomePage(),
);
},
),
}
}
}

```

File pdf_view.dart

```

import 'dart:io';
import 'dart:ui' as ui;
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:pdf/pdf.dart';
import 'package:pdf/widgets.dart' as pw;
import 'package:intl/intl.dart';
import 'package:gd8_library_xxxx/invoice/get_total_invoice.dart';
import 'package:gd8_library_xxxx/invoice/model/custom_row_invoice.dart';
import 'package:gd8_library_xxxx/invoice/model/product.dart';
import 'package:barcode_widget/barcode_widget.dart';
import 'package:gd8_library_xxxx/preview_screen.dart';
import 'package:gd8_library_xxxx/invoice/item_doc.dart';
import 'package:responsive_sizer/responsive_sizer.dart';

Future<void> createPdf(
    TextEditingController nameController,

```

```
TextEditingController birthdayController,
TextEditingController ageController,
TextEditingController phoneController,
TextEditingController addressController,
String id,
File image,
BuildContext context,
List<Product> soldProducts) async {
final doc = pw.Document();
final now = DateTime.now();
final formattedDate = DateFormat('yyyy-MM-dd\n HH:mm:ss')
    .format(now); //menampilkan waktu terbuatnya dokumen PDF

//mengambil gambar dari asset untuk ditampilkan pada dokumen PDF
final imageBytesHeader =
    (await rootBundle.load("assets/college.png")).buffer.asUint8List();
final imageHeader = pw.MemoryImage(imageBytesHeader);

final imageLogo =
    (await
rootBundle.load("assets/flutter_explained_logo.jpg")).buffer.asUint8List();
final imageInvoice = pw.MemoryImage(imageLogo);

//mengambil gambar dari input galeri atau kamera
pw.ImageProvider pdfImageProvider(Uint8List imageBytes) {
    return pw.MemoryImage(imageBytes);
}

final imageBytes = image!.readAsBytesSync();

//Memberi border pada dokumen pdf
final pdfTheme = pw.PageTheme(
    pageFormat: PdfPageFormat.a4,
    buildBackground: (pw.Context context) {
        return pw.Container(
            decoration: pw.BoxDecoration(
                border: pw.Border.all(
                    color: PdfColor.fromHex('#FFBD59'),
                    width: 1.w,
                ),
            ),
        );
    },
);

final List<CustomRow> elements = [
    CustomRow("Item Name", "Item Price", "Amount", "Sub Total Product"),
    for (var product in soldProducts)
```

```
CustomRow(
    product.name,
    product.price.toStringAsFixed(2),
    product.amount.toString(),
    (product.price * product.amount).toStringAsFixed(2),
),
CustomRow(
    "Sub Total",
    "",
    "",
    "Rp ${getSubTotal(soldProducts)}",
),
CustomRow(
    "PPN Total(11%)",
    "",
    "",
    "Rp ${getPPNTotal(soldProducts)} EUR",
),
CustomRow(
    "Total",
    "",
    "",
    "Rp ${((double.parseDouble(getSubTotal(soldProducts)) +
double.parseDouble(getPPNTotal(soldProducts))).toStringAsFixed(2)} EUR",
)
];
//Pengaturan tampilan dokumen PDFnya
//Di sini menggunakan Multi Page agar dapat menampilkan lebih dari 1 halaman.
//kemudian menggunakan header untuk header dokumen PDF, build untuk isi dokumen PDFnya, dan footer untuk footer dokumen PDF
doc.addPage(
    pw.MultiPage(
        pageTheme: pdfTheme,
        header: (pw.Context context) {
            return pw.Header(
                margin: pw.EdgeInsets.zero,
                outlineColor: PdfColors.amber50,
                outlineStyle: PdfOutlineStyle.normal,
                level: 5,
                decoration: pw.BoxDecoration(
                    shape: pw.BoxShape.rectangle,
                    gradient: pw.LinearGradient(
                        colors: [
                            PdfColor.fromHex('#FCDF8A'),
                            PdfColor.fromHex('#F38381')
                        ],
                    )
                )
            );
        },
        footer: (pw.Context context) {
            return pw.Footer(
                margin: pw.EdgeInsets.all(10),
                text: "Page ${context.pageNumber} of ${context.pageCount}",
                style: pw.FontStyle(
                    color: PdfColors.amber50,
                    weight: PdfFontWeight.bold,
                    size: 14
                )
            );
        }
    )
);
```

```
        begin: pw.Alignment.topLeft,
        end: pw.Alignment.bottomRight,
    ),
),
child: pw.Container(
    child: pw.Center(
        child: pw.Row(
            mainAxisAlignment: pw.MainAxisAlignment.center,
            children: [
                pw.Center(
                    child: pw.Text(
                        'Identity Documentation',
                        style: pw.TextStyle(
                            fontWeight: pw.FontWeight.bold,
                            fontSize: 8.sp,
                        ),
                    ),
                ),
                pw.Image(imageHeader,
                    height: 5.h, width: 5.w, alignment:
pw.Alignment.topLeft)
            ],
        )),
    )));
},
build: (pw.Context context) {
    return [
        pw.Center(
            child: pw.Column(
                mainAxisAlignment: pw.MainAxisAlignment.center,
                crossAxisAlignment: pw.CrossAxisAlignment.center,
                children: [
                    pw.Container(
                        margin: pw.EdgeInsets.symmetric(
                            horizontal: 2.h, vertical: 2.h)),
                    pw.Padding(
                        padding:
                            pw.EdgeInsets.symmetric(horizontal: 2.h, vertical: 1.h),
                        child: pw.FittedBox(
                            child: pw.Image(pdfImageProvider(imageBytes), width:
33.h),
                            fit: pw.BoxFit.fitHeight,
                            alignment: pw.Alignment.center,
                        ),
                    ),
                    pw.Padding(
                        padding:
                            pw.EdgeInsets.symmetric(horizontal: 2.h, vertical: 1.h),
                    ),
                ],
            ),
        );
    ];
}
```

```
        child: pw.Text('Created at',
                      style:
                        pw.TextStyle(fontSize: 8.sp, color:
PdfColors.blue)),
      ),
      pw.Padding(
        padding: pw.EdgeInsets.symmetric(horizontal: 2.h),
        child: pw.Text(formattedDate,
                      style:
                        pw.TextStyle(fontSize: 8.sp, color:
PdfColors.blue)),
      ),
      pw.Padding(
        padding:
          pw.EdgeInsets.symmetric(horizontal: 5.h, vertical: 1.h),
        child: pw.Table(
          border: pw.TableBorder.all(),
          children: [
            pw TableRow(
              children: [
                pw.Padding(
                  padding: pw.EdgeInsets.symmetric(
                    horizontal: 1.h, vertical: 1.h),
                  child: pw.Text(
                    'Name',
                    style: pw.TextStyle(
                      fontWeight: pw.FontWeight.bold,
                      fontSize: 10.sp,
                    ),
                  ),
                ),
              ],
            ),
            pw.Padding(
              padding: pw.EdgeInsets.symmetric(
                horizontal: 1.h, vertical: 1.h),
              child: pw.Text(
                nameController.text,
                style: pw.TextStyle(
                  fontWeight: pw.FontWeight.bold,
                  fontSize: 10.sp,
                ),
              ),
            ),
          ],
        ),
        pw.TableRow(
          children: [
            pw.Padding(
              padding: pw.EdgeInsets.symmetric(

```

```
        horizontal: 1.h, vertical: 1.h),
    child: pw.Text(
        'Birthday',
        style: pw.TextStyle(
            fontWeight: pw.FontWeight.bold,
            fontSize: 10.sp,
        ),
    ),
),
),
],
),
pw.Padding(
padding: pw.EdgeInsets.symmetric(
    horizontal: 1.h, vertical: 1.h),
child: pw.Text(
    birthdayController.text,
    style: pw.TextStyle(
        fontWeight: pw.FontWeight.bold,
        fontSize: 10.sp,
    ),
),
),
),
],
),
),
pw.TableRow(
children: [
pw.Padding(
padding: pw.EdgeInsets.symmetric(
    horizontal: 1.h, vertical: 1.h),
child: pw.Text(
    'Age',
    style: pw.TextStyle(
        fontWeight: pw.FontWeight.bold,
        fontSize: 10.sp,
    ),
),
),
),
],
),
pw.Padding(
padding: pw.EdgeInsets.symmetric(
    horizontal: 1.h, vertical: 1.h),
child: pw.Text(
    ageController.text,
    style: pw.TextStyle(
        fontWeight: pw.FontWeight.bold,
        fontSize: 10.sp,
    ),
),
),
),
],
),
```

```
pw.TableRow(
    children: [
        pw.Padding(
            padding: pw.EdgeInsets.symmetric(
                horizontal: 1.h, vertical: 1.h),
            child: pw.Text(
                'Phone Number',
                style: pw.TextStyle(
                    fontWeight: pw.FontWeight.bold,
                    fontSize: 10.sp,
                ),
            ),
        ),
        pw.Padding(
            padding: pw.EdgeInsets.symmetric(
                horizontal: 1.h, vertical: 1.h),
            child: pw.Text(
                phoneController.text,
                style: pw.TextStyle(
                    fontWeight: pw.FontWeight.bold,
                    fontSize: 10.sp,
                ),
            ),
        ),
        ],
    ),
    pw.TableRow(
        children: [
            pw.Padding(
                padding: pw.EdgeInsets.symmetric(
                    horizontal: 1.h, vertical: 1.h),
                child: pw.Text(
                    'Address',
                    style: pw.TextStyle(
                        fontWeight: pw.FontWeight.bold,
                        fontSize: 10.sp,
                    ),
                ),
            ),
        ],
        pw.Padding(
            padding: pw.EdgeInsets.symmetric(
                horizontal: 1.h, vertical: 1.h),
            child: pw.Text(
                addressController.text,
                style: pw.TextStyle(
                    fontWeight: pw.FontWeight.bold,
                    fontSize: 10.sp,
                ),
            ),
        ),
    ],
);
```

```
        ),
        ),
    ],
),
),
),
),
),
),
),
),
pw.SizedBox(height: 7.h),
pw.Padding(
    padding: const pw.EdgeInsets.all(8.0),
    child: pw.Row(
        mainAxisAlignment: pw.MainAxisAlignment.spaceBetween,
        children: [
            pw.Column(
                children: [
                    pw.Text("Customer Name"),
                    pw.Text("Customer Address"),
                    pw.Text("Customer City"),
                ],
            ),
            pw.Image(imageInvoice,
                height: 10.h, width: 10.w),
            pw.Column(
                children: [
                    pw.Text("Mahasiswa PBP"),
                    pw.Text(addressController.text),
                    pw.Text("Invoice-Id: $id")
                ],
            )
        ],
    ),
),
pw.SizedBox(height: 20.h),
pw.Padding(
    padding: const pw.EdgeInsets.all(8.0),
    child: pw.Column(children: [
        pw.Text(
            "Dear Customer, thanks for buying at Flutter
Explained, feel free to see the list of items below."),
        pw.SizedBox(height: 10.h),
        pw.Container(
            height: 20.h,
            color: PdfColor.fromHex('#BEE0EC'),
            child: itemColumn(elements)),
        pw.SizedBox(height: 10.h),
        pw.Text("Thanks for your trust, and till the next
time."),
        pw.SizedBox(height: 10.h),
```

```

        pw.Text("Kind regards,"),
        pw.SizedBox(height: 10.h),
        pw.Text("Max Weber"),
      ])),
      pw.Padding(
        padding:
          pw.EdgeInsets.symmetric(horizontal: 1.h, vertical: 1.h),
        child: pw.Stack(
          alignment: pw.Alignment.center,
          children: [
            pw.BarcodeWidget(
              barcode: pw.Barcode.qrCode(
                errorCorrectLevel: BarcodeQRCorrectionLevel.high,
              ),
              data: id,
              width: 15.w,
              height: 15.h,
            ),
            ],
          ),
        ),
      ])),
    ],
  );
},
footer: (pw.Context context) {
  return pw.Container(
    color: PdfColor.fromHex('#FFBD59'),
    child: pw.Center(child: pw.Text('-Modul 8 Library-')));
},
),
);
// Page
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => PreviewScreen(doc: doc),
  ));
}

```

File preview_screen.dart

```

import 'package:flutter/material.dart';
import 'package:pdf/widgets.dart' as pw;
import 'package:pdf/pdf.dart';
import 'package:printing/printing.dart';

class PreviewScreen extends StatelessWidget {

```

```
final pw.Document doc;

const PreviewScreen({
  Key? key,
  required this.doc,
}) : super(key: key);

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: IconButton(
        onPressed: () => Navigator.pop(context),
        icon: const Icon(Icons.arrow_back_outlined),
      ),
      centerTitle: true,
      title: const Text("Preview"),
    ),
    body: PdfPreview(
      build: (format) => doc.save(),
      allowSharing: true,
      allowPrinting: true,
      initialPageFormat: PdfPageFormat.a4,
    ),
  );
}
```