



D3 Teknik Informatika (TI)  
**Jurusan Teknik Informatika**  
Politeknik Negeri Indramayu

# UI Swing Responsif

Fachrul Pralienka Bani Muhamad, S.ST., M.Kom.  
fachrul.pbm@polindra.ac.id

**2025**

# Tujuan

Setelah materi ini disampaikan, mahasiswa diharapkan mampu:

- **Menyajikan** tampilan aplikasi yang responsif berdasarkan ukuran jendela saat ini melalui penerapan event handling

# Pengantar UI Swing Responsif

- Aplikasi desktop yang tidak modern seringkali didesain dengan ukuran jendela yang tetap (kaku)
- Aplikasi desktop yang modern, fleksibel, dan ramah pengguna, memastikan pengalaman yang konsisten terlepas dari lingkungan tampilan pengguna
- Swing memiliki kemampuan untuk mengubah tata letak (layout), ukuran, dan visibilitas komponennya secara dinamis sebagai respons terhadap perubahan dimensi jendela aplikasi

# ComponentListener

- Event handler yang mendengarkan perubahan ukuran jendela secara real-time
- Penghubung antara tindakan pengguna (mengubah ukuran jendela) dan logika responsifitas aplikasi
- Memicu logika perubahan tata letak
- Setiap kali lebar atau tinggi jendela berubah, metode `componentResized()` dipanggil
- Dapat diperoleh ukuran jendela Saat Ini
- Memvalidasi ulang tata letak (`revalidate()` dan `repaint()`)

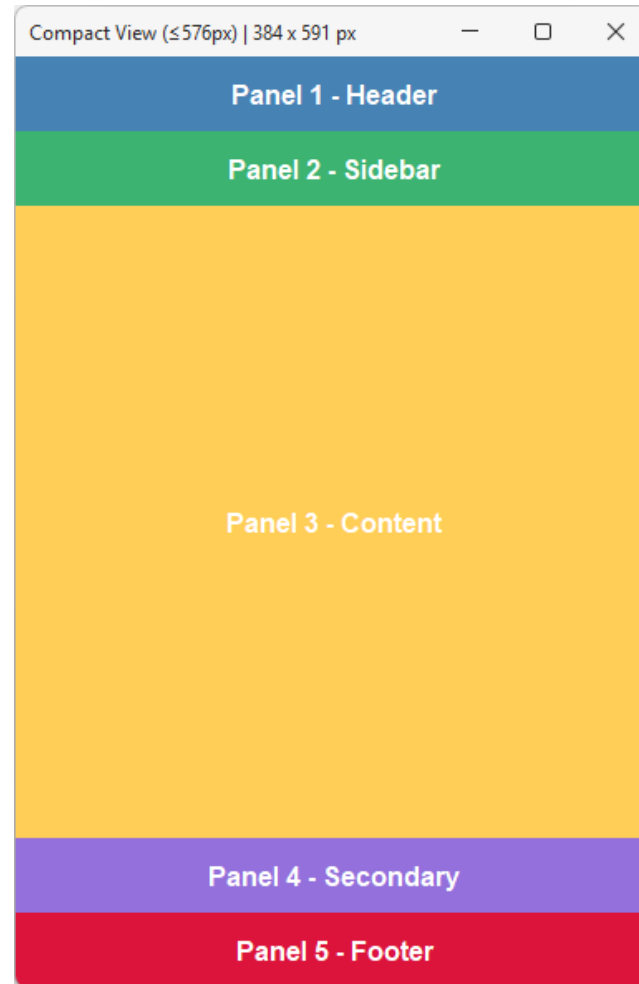
# Threshold Lebar (Width) Jendela Aplikasi

No	Kategori	Treshold (px)
1	Compact View	$\leq 576$ px
2	Split View	576 - 768 px
3	Intermediate View	768 - 992 px
4	Desktop Small	992 - 1200 px
5	Desktop Standard	1200 - 1400 px)
6	Desktop Large	$\geq 1400$ px

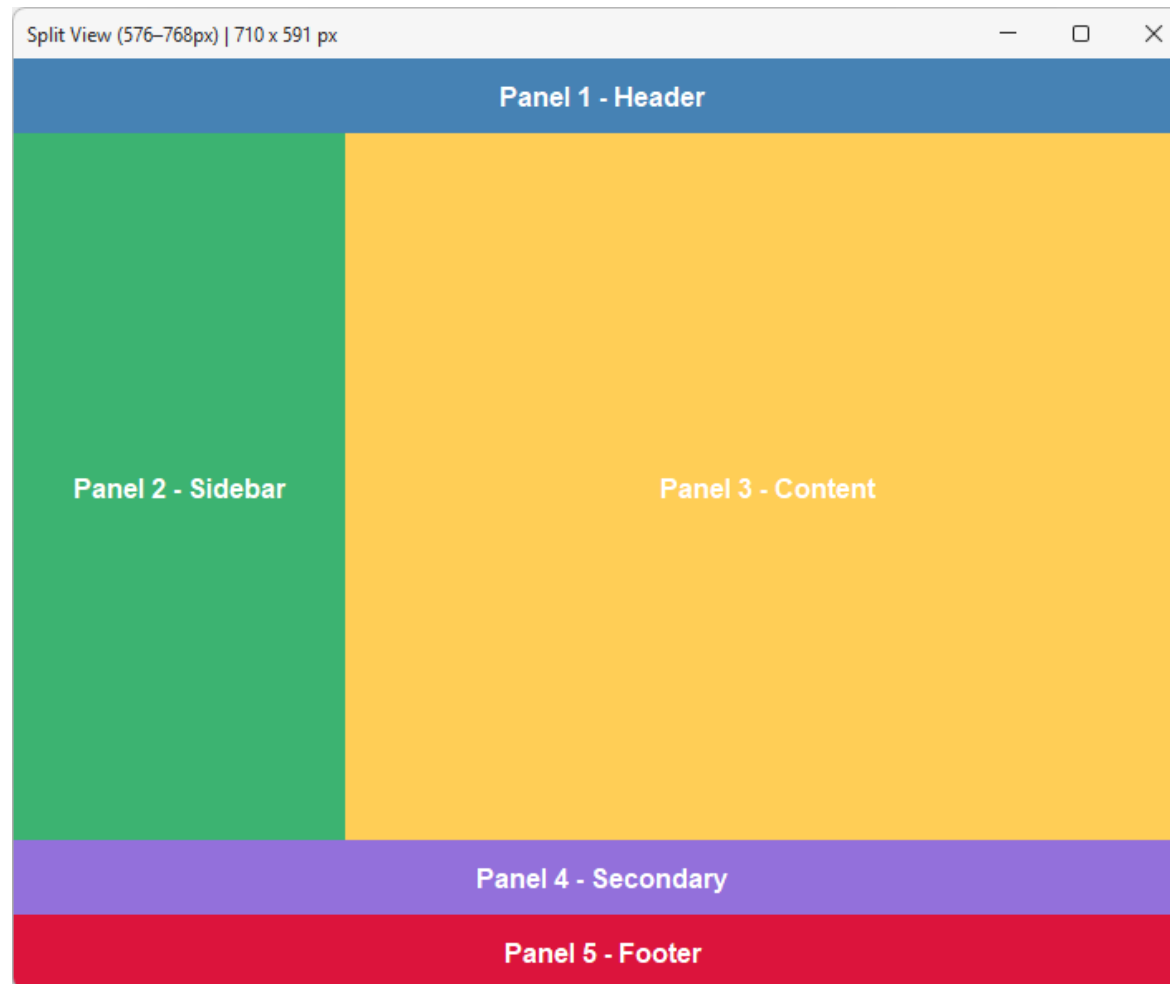
# Langkah Penerapan

1. Merancang tata letak (layout) awal
2. Menambahkan event listener (`componentListener`) dengan Adapter `componentResized`
3. Menerapkan logika threshold ukuran jendela
4. Mengubah tata letak (layout) berdasarkan threshold

# Compact View ( $\leq 576$ px)

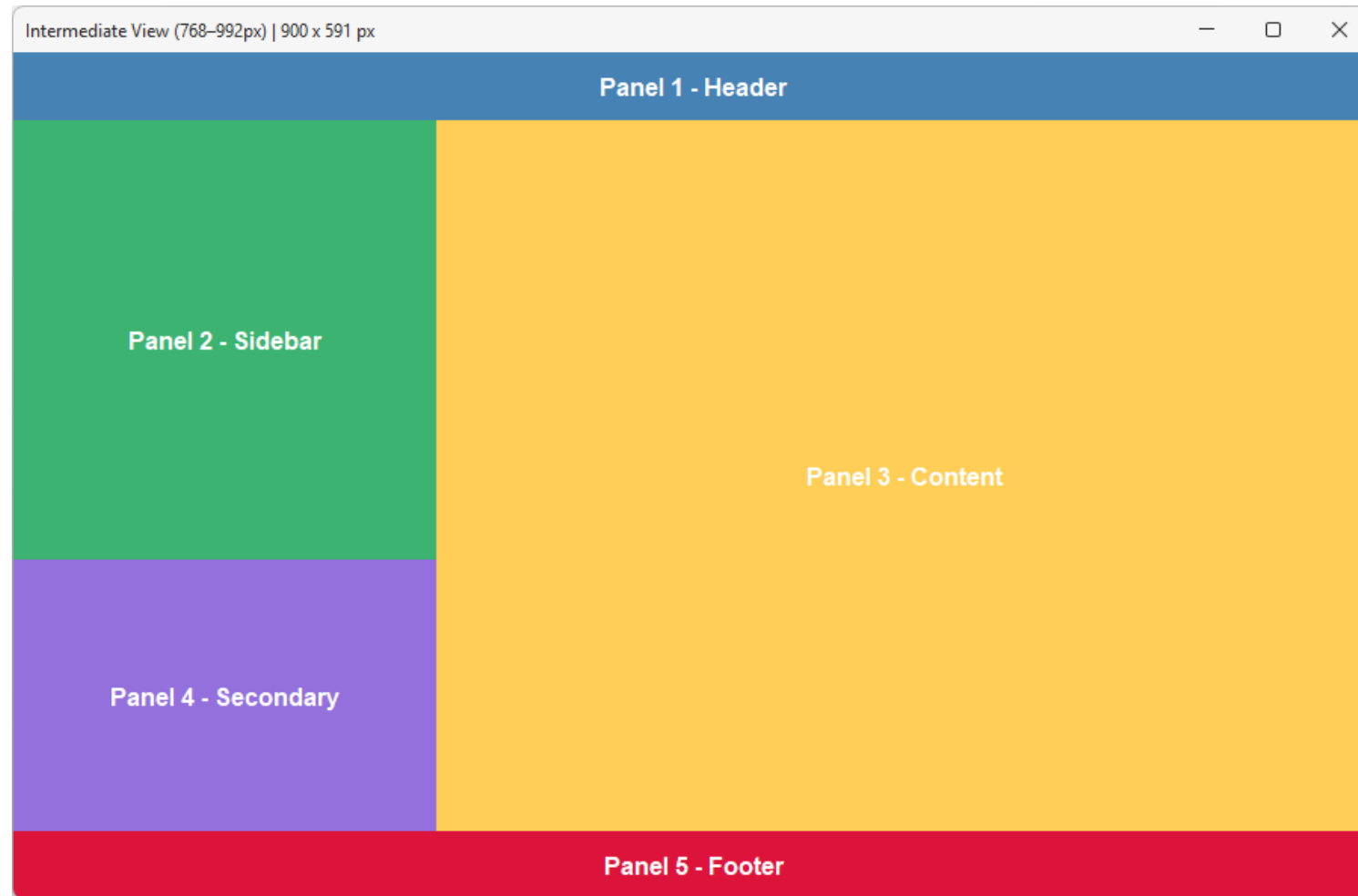


# Split View (576 - 768 px)

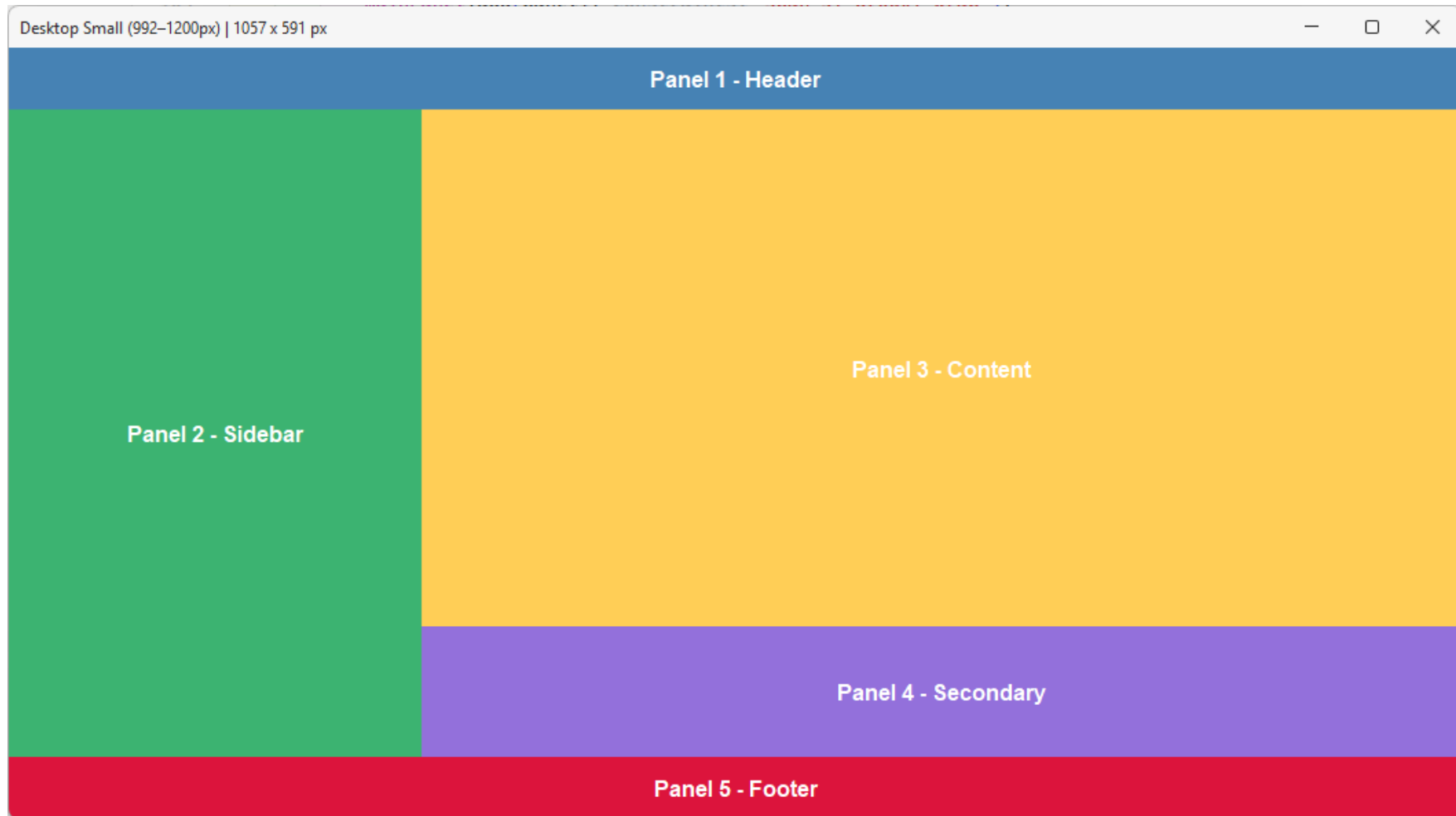




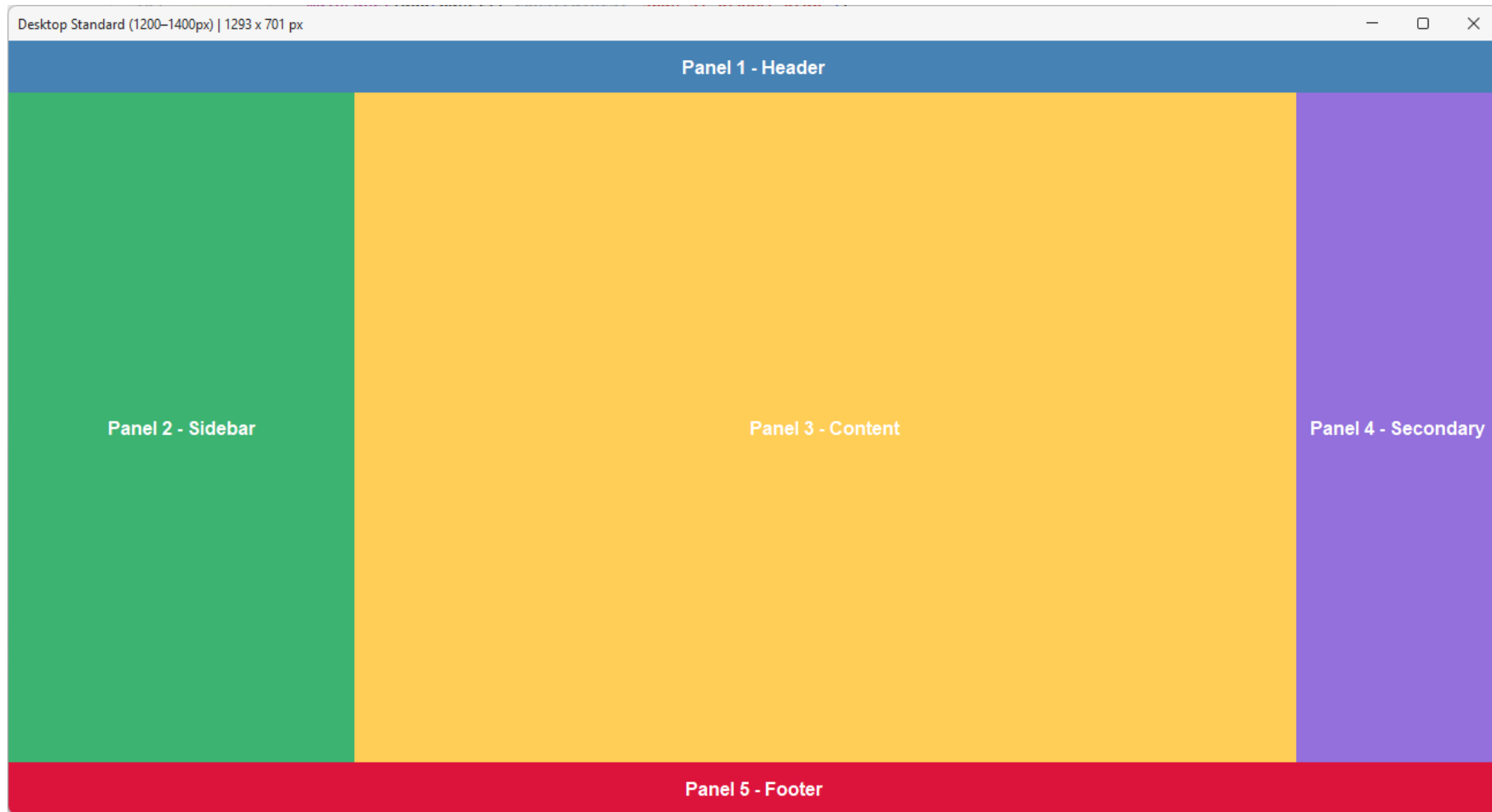
# Intermediate View (768 - 992 px)



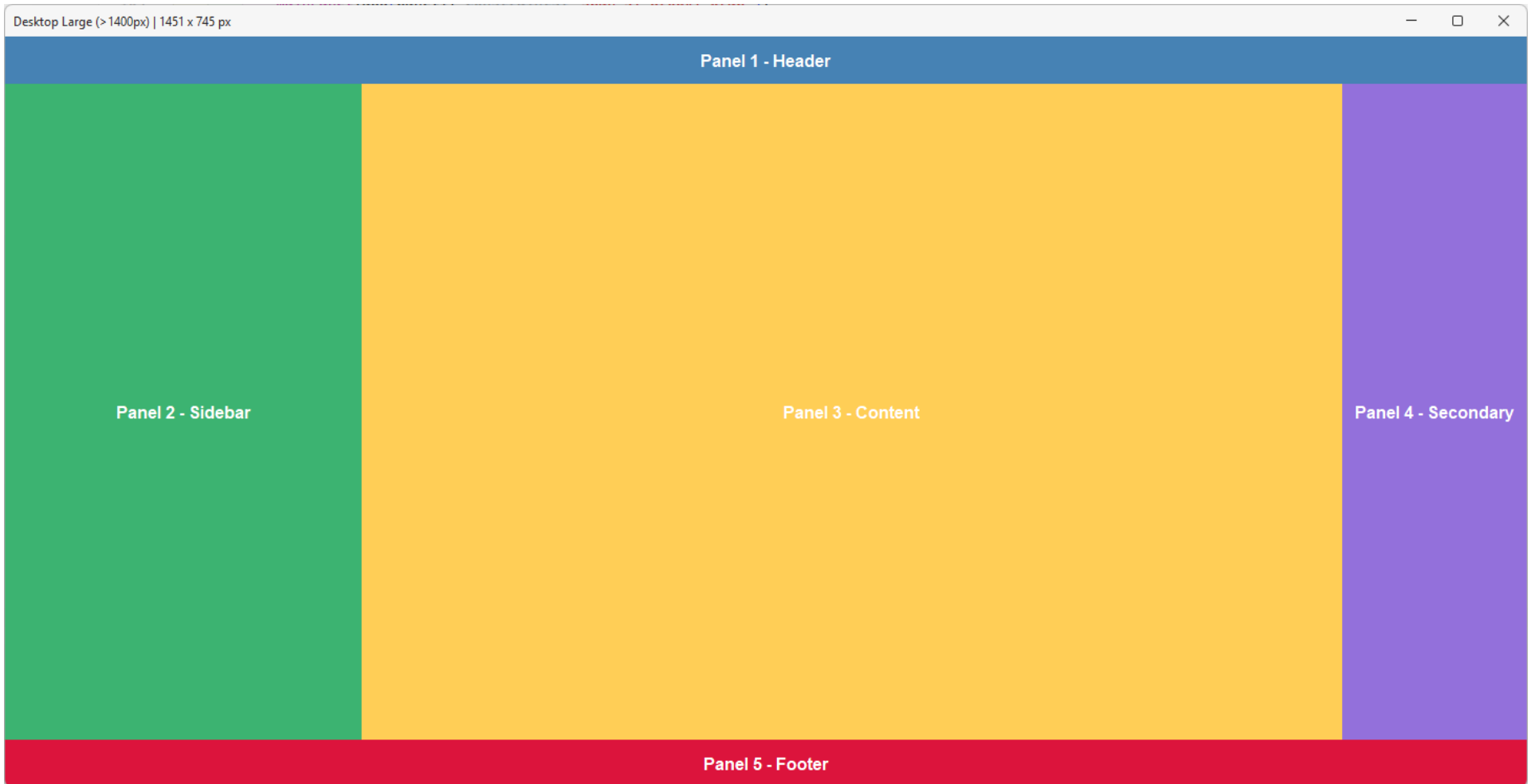
# Desktop Small (992 - 1200 px)



# Desktop Standard (1200 - 1400 px)



# Desktop Large ( $\geq 1400$ px)





```
10 public class UISwingResponsif extends JFrame {
11
12     private JPanel mainPanel;
13     private JPanel panel1;
14     private JPanel panel2;
15     private JPanel panel3;
16     private JPanel panel4;
17     private JPanel panel5;
18
19     private static final int COMPACT_VIEW_MAX = 576;
20     private static final int SPLIT_VIEW_MAX = 768;
21     private static final int INTERMEDIATE_VIEW_MAX = 992;
22     private static final int DESKTOP_SMALL_MAX = 1200;
23     private static final int DESKTOP_STANDARD_MAX = 1400;
24
25     public UISwingResponsif() {
26         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27         setMinimumSize(new Dimension(400, 600));
28
29         panel1 = createColoredPanel("Panel 1 - Header", new Color(70, 130, 180));
30         panel2 = createColoredPanel("Panel 2 - Sidebar", new Color(60, 179, 113));
31         panel3 = createColoredPanel("Panel 3 - Content", new Color(255, 206, 86));
32         panel4 = createColoredPanel("Panel 4 - Secondary", new Color(147, 112, 219));
33         panel5 = createColoredPanel("Panel 5 - Footer", new Color(220, 20, 60));
34
35         mainPanel = new JPanel();
36         add(mainPanel);
```

```

38     addComponentListener(new ComponentAdapter() {
39         @Override
40         public void componentResized(ComponentEvent e) {
41             updateLayoutAndTitle();
42         }
43
44         @Override
45         public void componentMoved(ComponentEvent e) {
46             updateLayoutAndTitle();
47         }
48     });
49
50     updateLayoutAndTitle();
51
52     pack();
53     setExtendedState(JFrame.MAXIMIZED_BOTH);
54     setLocationRelativeTo(null);
55 }
56
57 private void updateLayoutAndTitle() {
58     Dimension size = getSize(); // ukuran aktual termasuk border
59     Insets insets = getInsets(); // border + title bar
60     int innerWidth = size.width - insets.left - insets.right;
61     int innerHeight = size.height - insets.top - insets.bottom;
62
63     String category = getCategory(innerWidth);
64     setTitle(category + " | " + innerWidth + " x " + innerHeight + " px");
65
66     rebuildLayout(innerWidth);
67 }

```

```

68
69     private String getCategory(int width) {
70         if (width <= COMPACT_VIEW_MAX) return "Compact View (≤576px)";
71         else if (width <= SPLIT_VIEW_MAX) return "Split View (576–768px)";
72         else if (width <= INTERMEDIATE_VIEW_MAX) return "Intermediate View (768–992px)";
73         else if (width <= DESKTOP_SMALL_MAX) return "Desktop Small (992–1200px)";
74         else if (width <= DESKTOP_STANDARD_MAX) return "Desktop Standard (1200–1400px)";
75         else return "Desktop Large (>1400px)";
76     }
77
78     private void rebuildLayout(int width) {
79         mainPanel.removeAll();
80         mainPanel.setLayout(new MigLayout("fill, insets 0, gap 0"));
81
82         if (width <= COMPACT_VIEW_MAX) {
83             mainPanel.add(panel1, "grow, wrap");
84             mainPanel.add(panel2, "grow, wrap");
85             mainPanel.add(panel3, "grow, push, h 300::, wrap");
86             mainPanel.add(panel4, "grow, wrap");
87             mainPanel.add(panel5, "grow, wrap");
88
89         } else if (width <= SPLIT_VIEW_MAX) {
90             mainPanel.add(panel1, "span 2, growx, wrap");
91             mainPanel.add(panel2, "w 200!, growy");
92             mainPanel.add(panel3, "grow, push, wrap");
93             mainPanel.add(panel4, "span 2, growx, wrap");
94             mainPanel.add(panel5, "span 2, growx, wrap");
95

```

```

196     } else if (width <= INTERMEDIATE_VIEW_MAX) {
197         mainPanel.add(panel1, "span 2, growx, wrap");
198         mainPanel.add(panel2, "w 280!, growy");
199         mainPanel.add(panel3, "grow, push, wrap, spany 2");
200         mainPanel.add(panel4, "growx, h 180!, wrap");
201         mainPanel.add(panel5, "span 2, growx, wrap");
202
203     } else if (width <= DESKTOP_SMALL_MAX) {
204         mainPanel.add(panel1, "span 3, growx, wrap");
205         mainPanel.add(panel2, "growy, span 1 2, w 300!, h 100%"); // span 1 kolom x 2 baris
206         mainPanel.add(panel3, "grow, push, wrap");
207         mainPanel.add(panel4, "h 25%, growx, wrap");
208         mainPanel.add(panel5, "span 3, growx");
209
210     } else if (width <= DESKTOP_STANDARD_MAX) {
211         mainPanel.add(panel1, "span 3, growx, wrap");
212         mainPanel.add(panel2, "w 300!, growy");
213         mainPanel.add(panel3, "grow, pushx 2, pushy");
214         mainPanel.add(panel4, "grow, wrap");
215         mainPanel.add(panel5, "span 3, growx, wrap");
216
217     } else {
218         mainPanel.add(panel1, "span 3, growx, wrap");
219         mainPanel.add(panel2, "w 340!, growy");
220         mainPanel.add(panel3, "grow, pushx 3, pushy");
221         mainPanel.add(panel4, "grow, wrap");
222         mainPanel.add(panel5, "span 3, growx, wrap");
223     }
224
225     mainPanel.revalidate();
226     mainPanel.repaint();
227 }

```



```

128
129     private JPanel createColoredPanel(String title, Color background) {
130         JPanel panel = new JPanel(new MigLayout("fill, insets 0"));
131         panel.setBackground(background);
132         panel.setBorder(BorderFactory.createEmptyBorder(12, 12, 12, 12));
133         JLabel label = new JLabel(title, SwingConstants.CENTER);
134         label.setFont(new Font("Inter", Font.BOLD, 16));
135         label.setForeground(Color.WHITE);
136         label.setOpaque(false);
137         panel.add(label, "grow, align center");
138
139         return panel;
140     }
141
142     public static void main(String[] args) {
143         try {
144             UIManager.setLookAndFeel(new FlatMacLightLaf());
145         } catch (UnsupportedLookAndFeelException e) {
146             e.printStackTrace();
147         }
148         SwingUtilities.invokeLater(() -> {
149             new UISwingResponsif().setVisible(true);
150         });
151     }
152 }

```

# Latihan

- **Split View** (576 - 768 px)  
(semua panel melebar kiri ke kanan)
- **Desktop Small** (992 - 1200 px)  
(panel client dan panel additional information bersebelahan, panel information memanjang dan berada di bawah kedua panel tersebut)
- **Desktop Standard** (1200 - 1400 px)  
(normal sesuai layout awal)

Detail Proyek: RNV-JKT-AXG-001

Client

Client ID: **101**

Name: **Bapak Alex Gunawan**

Phone: **(+62) 8123456789**

Registration No: **RNV-JKT-AXG-001**

**Details**

Information

Reserve days:  **Add Days**

Buyer:

Seller:

Address:

Credit rating:  **S&P Update**

☐ Approved:

Additional Information

Estimated close:  **Edit**

Creation date: **2025-10-15**

Created by: **Admin**

Last edit date: **2025-11-16**

Last edited by: **Warnoto**

Closed date: **null**

Closed by: **null**

Product List

Renovation	Description	Part #	Quantity	List Price	Discount	Price	Wholesale Dis...	Wholesaler Pri...
Dapur	Keramik Dindi...	KW-PT-DLX-01	50	150000	0	7500000	5	7125000
Dapur	Lem Keramik I...	LMI-GRY-STD	5	50000	0	250000	0	250000
Dapur	Pipa PVC 3 inch	PVC-3IN-STD	12	35000	0	420000	10	378000
Kamar Mandi	Shower Set Mi...	SHW-MN-STL	1	850000	15	722500	5	686375
Kamar Mandi	Closet Duduk ...	CLO-DD-PRM	1	2500000	5	2375000	10	2137500
Ruang Tamu	Lampu Gantu...	LMP-HNG-CRS	2	750000	0	1500000	0	1500000

Subtotal list price: Rp 15.570.000  
Total retailer price: Rp 15.182.500  
Total wholesaler price: Rp 14.249.625

**Add**  
**Edit**  
**Delete**

Tasks

State	Task	Assigner	Executer	Creation Date	Estimated Date	Executed Date
Completed	Pengecatan ulang r...	Warnoto	Toni	2025-10-25	2025-10-28	2025-10-27
In Progress	Instalasi closet dudu...	Warnoto	Goang	2025-11-15	2025-11-17	
Delayed	Pemasangan kerami...	Warnoto	Toni	2025-11-01	2025-11-04	
Completed	Pemasangan pipa P...	Warnoto	Goang	2025-10-20	2025-10-21	2025-10-21
Not Started	Pembelian dan insta...	Warnoto	Toni	2025-11-16	2025-11-18	

**Add**  
**Edit**  
**Delete**

Comments



D3 Teknik Informatika (TI)  
**Jurusan Teknik Informatika**  
Politeknik Negeri Indramayu

# Selesai

Terima kasih 🙏