

LAPORAN AKHIR
PRAKTIKUM LABORATORIUM TI

Nama Praktikum : Rekaya Perangkat lunak 2
Kelas : 4IA26
Praktikum ke : 6
Tanggal : 18 November 2025
Materi : Jasperreport
NPM : 50422301
Nama : Aura Salsabila
Ketua Asisten :
Jumlah Lembar : 7



LABORATORIUM INFORMATIKA
UNIVERSITAS GUNADARMA
2025

LISTING

```
 1 // Data untuk laporan
 2 const reportData = {
 3   sales: {
 4     title: "Laporan Penjualan Bulanan",
 5     company: "PT. NEXUS SOLUTIONS",
 6     department: "Departemen Penjualan",
 7     columns: ["No", "Produk", "Kuantitas", "Harga Satuan", "Total"],
 8     data: [
 9       { no: 1, product: "Laptop Dell XPS 13", qty: 5, price: 15000000, total: 75000000 },
10       { no: 2, product: "Monitor LG 27 inch", qty: 12, price: 3500000, total: 42000000 },
11       { no: 3, product: "Keyboard Mechanical", qty: 20, price: 1200000, total: 24000000 },
12       { no: 4, product: "Mouse Gaming", qty: 15, price: 450000, total: 6750000 },
13       { no: 5, product: "Webcam HD", qty: 8, price: 800000, total: 6400000 },
14       { no: 6, product: "Printer HP LaserJet", qty: 3, price: 5000000, total: 15000000 },
15       { no: 7, product: "Network Switch", qty: 4, price: 2000000, total: 8000000 },
16       { no: 8, product: "UPS 2000VA", qty: 6, price: 3500000, total: 21000000 },
17     ],
18     summary: {
19       totalQty: 73,
20       totalRevenue: 198150000,
21       avgValue: 24768750,
22       topProduct: "Laptop Dell XPS 13"
23     }
24   },
25   inventory: {
26     title: "Laporan Stok Barang",
27     company: "PT. NEXUS SOLUTIONS",
28     department: "Departemen Gudang",
29     columns: ["No", "Kode Barang", "Nama Barang", "Stok", "Min. Stok", "Status"],
30     data: [
31       { no: 1, code: "BRG001", name: "Laptop Dell XPS 13", stock: 15, min: 5, status: "✓ Normal" },
32       { no: 2, code: "BRG002", name: "Monitor LG 27 inch", stock: 8, min: 10, status: "⚠ Rendah" },
33       { no: 3, code: "BRG003", name: "Keyboard Mechanical", stock: 45, min: 20, status: "✓ Normal" },
34       { no: 4, code: "BRG004", name: "Mouse Gaming", stock: 3, min: 10, status: "⚠ Rendah" },
35       { no: 5, code: "BRG005", name: "Webcam HD", stock: 12, min: 5, status: "✓ Normal" },
36       { no: 6, code: "BRG006", name: "Printer HP LaserJet", stock: 2, min: 3, status: "⚠ Rendah" },
37       { no: 7, code: "BRG007", name: "Network Switch", stock: 25, min: 10, status: "✓ Normal" },
38       { no: 8, code: "BRG008", name: "UPS 2000VA", stock: 18, min: 8, status: "✓ Normal" },
39     ],
40     summary: {
41       totalItems: 128,
42       lowStock: 3,
43       normalStock: 5,
44       warehouseValue: "Rp. 450.000.000"
45     }
46   },
47   financial: {
48     title: "Laporan Keuangan Kuartal",
49     company: "PT. NEXUS SOLUTIONS",
50     department: "Departemen Keuangan",
51     columns: ["No", "Kategori", "Periode", "Nilai", "Persentase"],
52     data: [
53       { no: 1, category: "Pendapatan Operasional", period: "Q1 2024", value: 500000000, percentage: "35%" },
54       { no: 2, category: "Pendapatan Penjualan", period: "Q1 2024", value: 450000000, percentage: "31%" },
55       { no: 3, category: "Biaya Operasional", period: "Q1 2024", value: -250000000, percentage: "-17%" },
56       { no: 4, category: "Biaya Personel", period: "Q1 2024", value: -180000000, percentage: "-12%" },
57       { no: 5, category: "Biaya Utilitas", period: "Q1 2024", value: -50000000, percentage: "-3%" },
58       { no: 6, category: "Laba Kotor", period: "Q1 2024", value: 470000000, percentage: "32%" },
59     ],
60     summary: {
61       totalIncome: 950000000,
62       totalExpense: 480000000,
63       netProfit: 470000000,
64       margin: "49.47%"
65     }
66   };
67 };
68
69 // Format currency
70 function formatCurrency(value) {
71   return new Intl.NumberFormat('id-ID', {
72     style: 'currency',
73     currency: 'IDR',
74     minimumFractionDigits: 0,
75     maximumFractionDigits: 0,
76   }).format(value);
77 }
78
79 // Format number
80 function formatNumber(value) {
81   return new Intl.NumberFormat('id-ID').format(value);
82 }
83
84 // Get current date
85 function getCurrentDate() {
86   const today = new Date();
87   const options = { year: 'numeric', month: 'long', day: 'numeric', weekday: 'long' };
88   return today.toLocaleDateString('id-ID', options);
89 }
90
```

```

1 let currentReport = 'sales';
2
3 // Show report berdasarkan tipe
4 function showReport(reportType) {
5     currentReport = reportType;
6
7     // Update active menu
8     document.querySelectorAll('.menu-item').forEach(item => {
9         item.classList.remove('active');
10    });
11    event.target.classList.add('active');
12
13    generateReport();
14 }
15
16 // Generate laporan
17 function generateReport() {
18     const reportType = currentReport;
19     const data = reportData[reportType];
20
21     if (!data) {
22         showToast('Laporan tidak tersedia', 'error');
23         return;
24     }
25
26     const reportDate = document.getElementById('reportDate').value || new Date().toISOString().split('T')[0];
27     const department = document.getElementById('department').value || data.department;
28
29     let html = `
30         <div class="report" id="printableReport">
31             <!-- Header Laporan -->
32             <div class="report-header">
33                 <div class="report-title">${data.title}</div>
34                 <div class="report-subtitle">${data.company}</div>
35                 <div class="report-subtitle">${department}</div>
36                 <div class="report-date">Tanggal: ${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long', day: 'numeric' })}</div>
37             </div>
38
39             <!-- Info Laporan -->
40             <div class="report-info">
41                 <div class="info-item">
42                     <span class="info-label">Perusahaan:</span>
43                     <span class="info-value">${data.company}</span>
44                 </div>
45                 <div class="info-item">
46                     <span class="info-label">Departemen:</span>
47                     <span class="info-value">${department}</span>
48                 </div>
49                 <div class="info-item">
50                     <span class="info-label">Periode:</span>
51                     <span class="info-value">${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long' })}</span>
52                 </div>
53                 <div class="info-item">
54                     <span class="info-label">Dicetak:</span>
55                     <span class="info-value">${getCurrentDate()}</span>
56                 </div>
57             </div>
58         `;
59
60     // Render tabel data
61     html += renderTable(data);
62
63     // Summary box
64     html += renderSummary(data);
65
66     // Footer dan signature
67     html += `
68         <div class="report-footer">
69             <p>Laporan ini adalah dokumen resmi dan bersifat rahasia perusahaan.</p>
70             <p>Dibuat oleh: Sistem Laporan Terpadu | Halaman: <span class="page-number">1</span></p>
71         </div>
72
73         <div class="signature-section">
74             <div class="signature-item">
75                 <div class="signature-name">Manajer Divisi</div>
76                 <div class="signature-date">Jakarta, ${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long', day: 'numeric' })}</div>
77             </div>
78             <div class="signature-item">
79                 <div class="signature-name">Kepala Departemen</div>
80             </div>
81             <div class="signature-item">
82                 <div class="signature-name">Direktur Utama</div>
83             </div>
84         </div>
85     `;
86
87     document.getElementById('reportContainer').innerHTML = html;
88     showToast('Laporan berhasil dibuat', 'success');
89 }
90
91 // Render tabel
92 function renderTable(data) {
93     let html = `
94         <div class="table-responsive">
95             <table>
96                 <thead>
97                     <tr>
98                 `;
99
100

```

```
 1 // Header table
 2     data.columns.forEach(col => {
 3         html += `<th>${col}</th>`;
 4     });
 5 
 6     html += `
 7         </tr>
 8     </thead>
 9     <tbody>
10 `;
11 
12 // Data rows
13 data.data.forEach((row, index) => {
14     html += `<tr>`;
15     data.columns.forEach(col => {
16         let value = row[col.toLowerCase()].replace(/\s+/g, ' ') || row[col.toLowerCase()];
17 
18         // Format currency jika nilai numerik besar
19         if (typeof value === 'number' && value > 1000) {
20             value = formatCurrency(value);
21         } else if (typeof value === 'number') {
22             value = formatNumber(value);
23         }
24 
25         html += `<td>${value}</td>`;
26     });
27     html += `</tr>`;
28 });
29 
30 html += `
31     </tbody>
32     </table>
33 </div>
34 `;
35 
36 return html;
37 }
38 
39 // Render summary
40 function renderSummary(data) {
41     let html = `<div class="summary-box">`;
42 
43     if (data.summary) {
44         const keys = Object.keys(data.summary);
45         keys.forEach(key => {
46             const value = data.summary[key];
47             const label = key
48                 .replace(/([A-Z])/g, ' $1')
49                 .replace(/\./, str => str.toUpperCase())
50                 .trim();
51 
52             let displayValue = value;
53             if (typeof value === 'number' && value > 1000) {
54                 displayValue = formatCurrency(value);
55             }
56 
57             html += `
58                 <div class="summary-item">
59                     <div class="summary-label">${label}</div>
60                     <div class="summary-value">${displayValue}</div>
61                 </div>
62             `;
63         });
64     }
65 
66     html += `</div>`;
67     return html;
68 }
69 
70 // Print laporan
71 function printReport() {
72     const printableReport = document.getElementById('printableReport');
73 
74     if (!printableReport) {
75         showToast('Laporan belum dibuat', 'error');
76         return;
77     }
78 
79     const printWindow = window.open('', '', 'width=800,height=600');
80     printWindow.document.write(`
81         <!DOCTYPE html>
82         <html>
83         <head>
84             <title>Print Laporan</title>
85             <link rel="stylesheet" href="styles.css">
86             <style>
87                 body { margin: 0; padding: 20px; }
88                 .report { box-shadow: none; }
89             </style>
90         </head>
91         <body>
92             ${printableReport.innerHTML}
93         </body>
94     `);
95 }
```

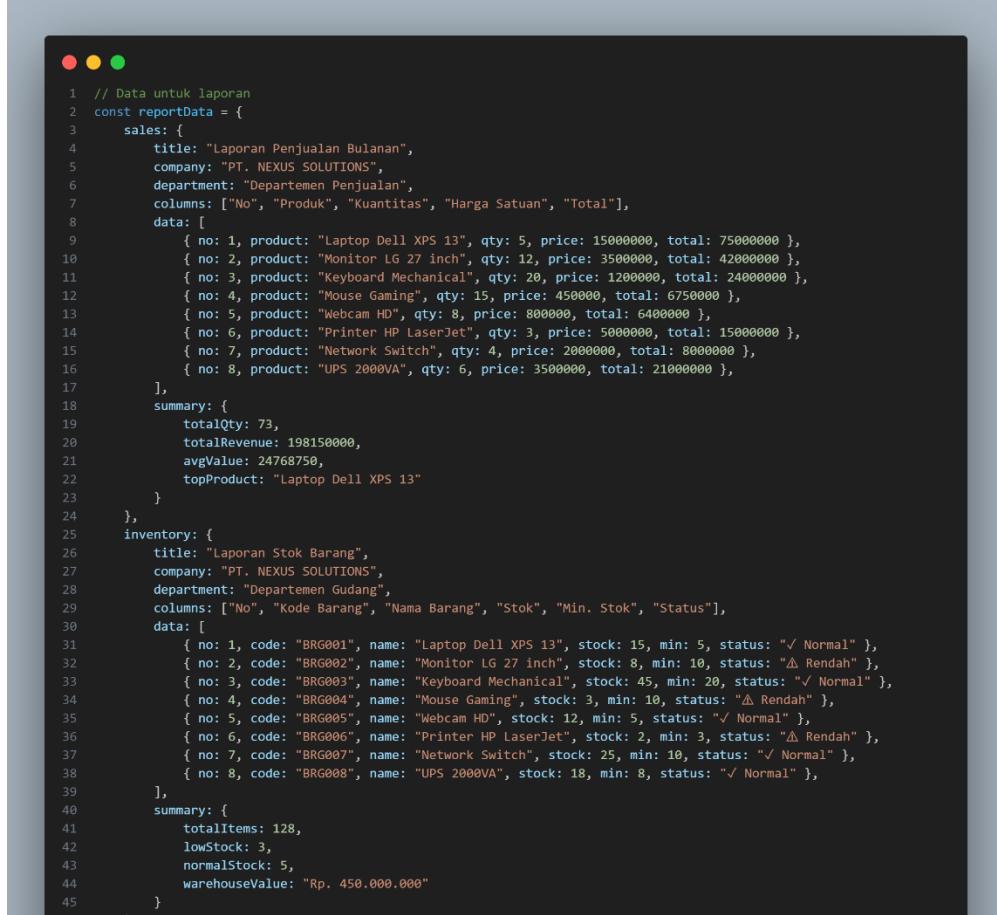
```

96     printWindow.document.close();
97
98     setTimeout(() => {
99         printWindow.print();
100    }, 250);
101
102     showToast('Membuka dialog cetak...', 'success');
103 }
104
105 // Export ke PDF
106 function exportToPDF() {
107     const reportContainer = document.getElementById('printableReport');
108
109     if (!reportContainer) {
110         showToast('Laporan belum dibuat', 'error');
111         return;
112     }
113
114     showToast('Memproses PDF...', 'success');
115
116     const element = reportContainer;
117     const opt = {
118         margin: [10, 10, 10, 10],
119         filename: `Laporan-${currentReport}-${new Date().toISOString().split('T')[0]}.pdf`,
120         image: { type: 'jpeg', quality: 0.98 },
121         html2canvas: { scale: 2, useCORS: true, allowTaint: true },
122         jsPDF: {
123             orientation: 'portrait',
124             unit: 'mm',
125             format: 'a4',
126             compress: true
127         },
128         pagebreak: { mode: ['avoid-all', 'css', 'legacy'] }
129     };
130
131     // Gunakan html2pdf jika tersedia
132     if (typeof html2pdf !== 'undefined') {
133         html2pdf().set(opt).from(element).save();
134         showToast('PDF berhasil diunduh!', 'success');
135     } else {
136         // Fallback: gunakan jsPDF + html2canvas
137         html2canvas(element, { scale: 2, allowTaint: true }).then(canvas => {
138             const { jsPDF } = window.jspdf;
139             const pdf = new jsPDF({
140                 orientation: 'portrait',
141                 unit: 'mm',
142                 format: 'a4',
143                 compress: true
144             });
145
146             const imgData = canvas.toDataURL('image/jpeg', 0.98);
147             const imgWidth = 190;
148             const imgHeight = (canvas.height * imgWidth) / canvas.width;
149
150             let heightLeft = imgHeight;
151             let position = 10;
152
153             pdf.addImage(imgData, 'JPEG', 10, position, imgWidth, imgHeight);
154             heightLeft -= (297 - 20);
155
156             while (heightLeft >= 0) {
157                 position = heightLeft - imgHeight;
158                 pdf.addPage();
159                 pdf.addImage(imgData, 'JPEG', 10, position, imgWidth, imgHeight);
160                 heightLeft -= 277;
161             }
162
163             pdf.save(`Laporan-${currentReport}-${new Date().toISOString().split('T')[0]}.pdf`);
164             showToast('PDF berhasil diunduh!', 'success');
165         }).catch(err => {
166             console.error('Error generating PDF:', err);
167             showToast('Gagal membuat PDF', 'error');
168         });
169     });
170 }
171
172 // Toast notification
173 function showToast(message, type = 'success') {
174     const toast = document.getElementById('toast');
175     toast.textContent = message;
176     toast.className = `toast ${type}`;
177
178     setTimeout(() => {
179         toast.classList.remove('show');
180     }, 3000);
181 }
182
183 // Set default date to today
184 document.addEventListener('DOMContentLoaded', () => {
185     const today = new Date().toISOString().split('T')[0];
186     document.getElementById('reportDate').value = today;
187 });
188
189

```

```
1  /* RESET */
2  * {
3  margin: 0;
4  padding: 0;
5  box-sizing: border-box;
6  }
7
8 body {
9  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
10 background-color: #eef2f7;
11 color: #333;
12 }
13
14 /* Layout Utama */
15 .container {
16  display: flex;
17  min-height: 100vh;
18 }
19
20 /* Sidebar */
21 .sidebar {
22  width: 250px;
23  background: #0d47a1;
24  color: white;
25  padding: 20px;
26  box-shadow: 2px 0 10px rgba(0, 0, 0, 0.1);
27  position: fixed;
28  height: 100vh;
29  overflow-y: auto;
30 }
31
32 .logo {
33  margin-bottom: 40px;
34  padding-bottom: 20px;
35  border-bottom: 1px solid rgba(255, 255, 255, 0.2);
36 }
37
38 .logo h2 {
39  font-size: 24px;
40  font-weight: 600;
41 }
42
43 .menu ul {
44  list-style: none;
45 }
46
47 .menu ul li {
48  margin-bottom: 10px;
49 }
50
51 .menu-item {
52  display: block;
53  padding: 12px 16px;
54  color: white;
55  text-decoration: none;
56  border-radius: 6px;
57  transition: all 0.3s ease;
58 }
59
60 .menu-item:hover,
61 .menu-item.active {
62  background-color: rgba(255, 255, 255, 0.15);
63  transform: translateX(5px);
64 }
65
66 /* Main Content */
67 .main-content {
68  margin-left: 250px;
69  flex: 1;
70  padding: 0;
71 }
72
73 /* Header */
74 .header {
75  background: #0d47a1;
76  color: white;
77  padding: 30px;
78  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
79 }
80
81 .header-content h1 {
82  font-size: 28px;
83  margin-bottom: 8px;
84 }
85
86 .header-content p {
87  font-size: 14px;
88  opacity: 0.9;
89 }
90
91 /* Report Section */
92 .report-section {
93  padding: 30px;
94 }
95
96 .report-controls {
97  background: white;
98  padding: 20px;
99  border-radius: 8px;
100 margin-bottom: 30px;
101 box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);
102 display: flex;
103 gap: 20px;
104 flex-wrap: wrap;
105 align-items: flex-end;
106 }
107
108 .control-group {
109  display: flex;
110  flex-direction: column;
111  gap: 8px;
112 }
113
114 .control-group label {
115  font-weight: 600;
116  font-size: 13px;
117  color: #555;
118 }
119
120 .control-group input,
121 .control-group select {
122  padding: 10px 12px;
123  border: 1px solid #cbd5e1;
124  border-radius: 4px;
125  font-size: 14px;
126  min-width: 150px;
127  transition: border-color 0.3s ease;
128 }
129
130 .control-group input:focus,
131 .control-group select:focus {
132  outline: none;
133  border-color: #2196f3;
134  box-shadow: 0 0 3px rgba(33, 150, 243, 0.2);
135 }
136
137 /* Buttons */
138 .btn {
139  padding: 10px 20px;
140  border: none;
141  border-radius: 6px;
142  font-size: 14px;
143  font-weight: 600;
144  cursor: pointer;
145  transition: all 0.3s ease;
146  white-space: nowrap;
147 }
148
```

LOGIKA



```
1 // Data untuk laporan
2 const reportData = {
3   sales: {
4     title: "Laporan Penjualan Bulanan",
5     company: "PT. NEXUS SOLUTIONS",
6     department: "Departemen Penjualan",
7     columns: ["No", "Produk", "Kuantitas", "Harga Satuan", "Total"],
8     data: [
9       { no: 1, product: "Laptop Dell XPS 13", qty: 5, price: 1500000, total: 7500000 },
10      { no: 2, product: "Monitor LG 27 inch", qty: 12, price: 3500000, total: 42000000 },
11      { no: 3, product: "Keyboard Mechanical", qty: 20, price: 1200000, total: 24000000 },
12      { no: 4, product: "Mouse Gaming", qty: 15, price: 450000, total: 6750000 },
13      { no: 5, product: "Webcam HD", qty: 8, price: 800000, total: 6400000 },
14      { no: 6, product: "Printer HP LaserJet", qty: 3, price: 5000000, total: 15000000 },
15      { no: 7, product: "Network Switch", qty: 4, price: 2000000, total: 8000000 },
16      { no: 8, product: "UPS 2000VA", qty: 6, price: 3500000, total: 21000000 },
17    ],
18    summary: {
19      totalQty: 73,
20      totalRevenue: 198150000,
21      avgValue: 24768750,
22      topProduct: "Laptop Dell XPS 13"
23    }
24  },
25  inventory: {
26    title: "Laporan Stok Barang",
27    company: "PT. NEXUS SOLUTIONS",
28    department: "Departemen Gudang",
29    columns: ["No", "Kode Barang", "Nama Barang", "Stok", "Min. Stok", "Status"],
30    data: [
31      { no: 1, code: "BRG001", name: "Laptop Dell XPS 13", stock: 15, min: 5, status: "✓ Normal" },
32      { no: 2, code: "BRG002", name: "Monitor LG 27 inch", stock: 8, min: 10, status: "⚠ Rendah" },
33      { no: 3, code: "BRG003", name: "Keyboard Mechanical", stock: 45, min: 20, status: "✓ Normal" },
34      { no: 4, code: "BRG004", name: "Mouse Gaming", stock: 3, min: 10, status: "⚠ Rendah" },
35      { no: 5, code: "BRG005", name: "Webcam HD", stock: 12, min: 5, status: "✓ Normal" },
36      { no: 6, code: "BRG006", name: "Printer HP LaserJet", stock: 2, min: 3, status: "⚠ Rendah" },
37      { no: 7, code: "BRG007", name: "Network Switch", stock: 25, min: 10, status: "✓ Normal" },
38      { no: 8, code: "BRG008", name: "UPS 2000VA", stock: 18, min: 8, status: "✓ Normal" },
39    ],
40    summary: {
41      totalItems: 128,
42      lowStock: 3,
43      normalStock: 5,
44      warehouseValue: "Rp. 450.000.000"
45    }
50  }
51}
```

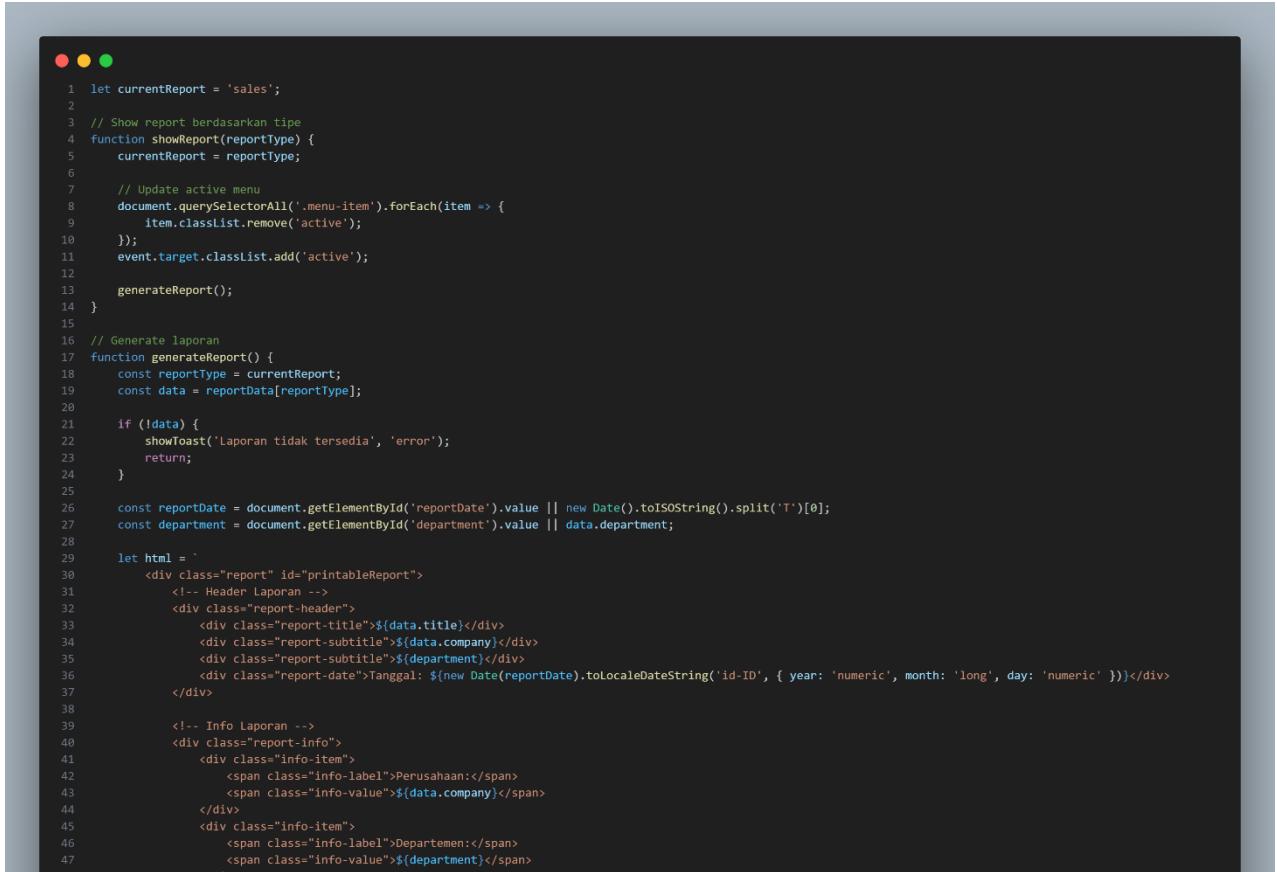
Pada laporan akhir pertemuan ke 6 ini kita membuat report data, dengan membuat dua jenis laporan, yaitu laporan penjualan dan laporan stok barang. Pada bagian penjualan, data mencatat berbagai produk seperti laptop, monitor, keyboard, mouse, dan perangkat lainnya lengkap dengan jumlah terjual, harga satuan, serta total pendapatan per produk kemudian bagian ini juga memiliki ringkasan berupa total kuantitas penjualan, total pendapatan keseluruhan, rata-rata nilai transaksi, dan produk terlaris. Selanjutnya, pada bagian inventori, laporan memuat kode barang, nama barang, jumlah stok yang tersedia, batas minimum stok, dan status stok apakah normal atau rendah lalu bagian ini juga dilengkapi ringkasan berupa total stok gudang, jumlah barang yang stoknya rendah, minimal stok, dan nilai keseluruhan stok gudang.

```

46     },
47     financial: {
48       title: "Laporan Keuangan Kuartal",
49       company: "PT. NEXUS SOLUTIONS",
50       department: "Departemen Keuangan",
51       columns: ["No", "Kategori", "Periode", "Nilai", "Persentase"],
52       data: [
53         { no: 1, category: "Pendapatan Operasional", period: "Q1 2024", value: 500000000, percentage: "35% " },
54         { no: 2, category: "Pendapatan Penjualan", period: "Q1 2024", value: 450000000, percentage: "31% " },
55         { no: 3, category: "Biaya Operasional", period: "Q1 2024", value: -250000000, percentage: "-17% " },
56         { no: 4, category: "Biaya Personel", period: "Q1 2024", value: -180000000, percentage: "-12% " },
57         { no: 5, category: "Biaya Utilitas", period: "Q1 2024", value: -50000000, percentage: "-3% " },
58         { no: 6, category: "Laba Kotor", period: "Q1 2024", value: 470000000, percentage: "32% " },
59       ],
60       summary: {
61         totalIncome: 950000000,
62         totalExpense: 480000000,
63         netProfit: 470000000,
64         margin: "49.47%"
65       }
66     }
67   };
68
69 // Format currency
70 function formatCurrency(value) {
71   return new Intl.NumberFormat('id-ID', {
72     style: 'currency',
73     currency: 'IDR',
74     minimumFractionDigits: 0,
75     maximumFractionDigits: 0,
76   }).format(value);
77 }
78
79 // Format number
80 function formatNumber(value) {
81   return new Intl.NumberFormat('id-ID').format(value);
82 }
83
84 // Get current date
85 function getCurrentDate() {
86   const today = new Date();
87   const options = { year: 'numeric', month: 'long', day: 'numeric', weekday: 'long' };
88   return today.toLocaleDateString('id-ID', options);
89 }
90

```

Kemudian pada gambar ketiga ini terlihat bahwa objek reportData juga memiliki bagian financial yang menyajikan laporan keuangan kuartal untuk perusahaan “PT. NEXUS SOLUTIONS”. Bagian ini berisi kategori-kategori seperti pendapatan operasional, pendapatan penjualan, biaya operasional, biaya personel, biaya utilitas, hingga laba kotor, lengkap dengan periode, nilai angka, serta persentasenya sehingga laporan terlihat lebih terstruktur kemudian bagian summary merangkum total pemasukan, total pengeluaran, laba bersih, dan margin keuntungan yang memudahkan pembaca memahami kondisi finansial secara cepat. Lalu di bagian bawah kode terdapat beberapa fungsi utilitas, seperti formatCurrency untuk mengubah angka menjadi format rupiah sesuai standar Indonesia, formatNumber untuk menampilkan angka dengan format numerik lokal, dan getCurrentDate untuk menghasilkan tanggal saat ini dengan format lengkap.



```
1 let currentReport = 'sales';
2
3 // Show report berdasarkan tipe
4 function showReport(reportType) {
5     currentReport = reportType;
6
7     // Update active menu
8     document.querySelectorAll('.menu-item').forEach(item => {
9         item.classList.remove('active');
10    });
11    event.target.classList.add('active');
12
13    generateReport();
14 }
15
16 // Generate laporan
17 function generateReport() {
18     const reportType = currentReport;
19     const data = reportData[reportType];
20
21     if (!data) {
22         showToast('Laporan tidak tersedia', 'error');
23         return;
24     }
25
26     const reportDate = document.getElementById('reportDate').value || new Date().toISOString().split('T')[0];
27     const department = document.getElementById('department').value || data.department;
28
29     let html = `
30         <div class="report" id="printableReport">
31             <!-- Header Laporan -->
32             <div class="report-header">
33                 <div class="report-title">${data.title}</div>
34                 <div class="report-subtitle">${data.company}</div>
35                 <div class="report-subtitle">${department}</div>
36                 <div class="report-date">Tanggal: ${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long', day: 'numeric' })}</div>
37             </div>
38
39             <!-- Info Laporan -->
40             <div class="report-info">
41                 <div class="info-item">
42                     <span class="info-label">Perusahaan:</span>
43                     <span class="info-value">${data.company}</span>
44                 </div>
45                 <div class="info-item">
46                     <span class="info-label">Departemen:</span>
47                     <span class="info-value">${department}</span>
48                 </div>
49             </div>
50         </div>
51     `;
52
53     // Insert generated HTML into the page
54     const container = document.querySelector('#printableReport');
55     container.innerHTML = html;
56 }
```

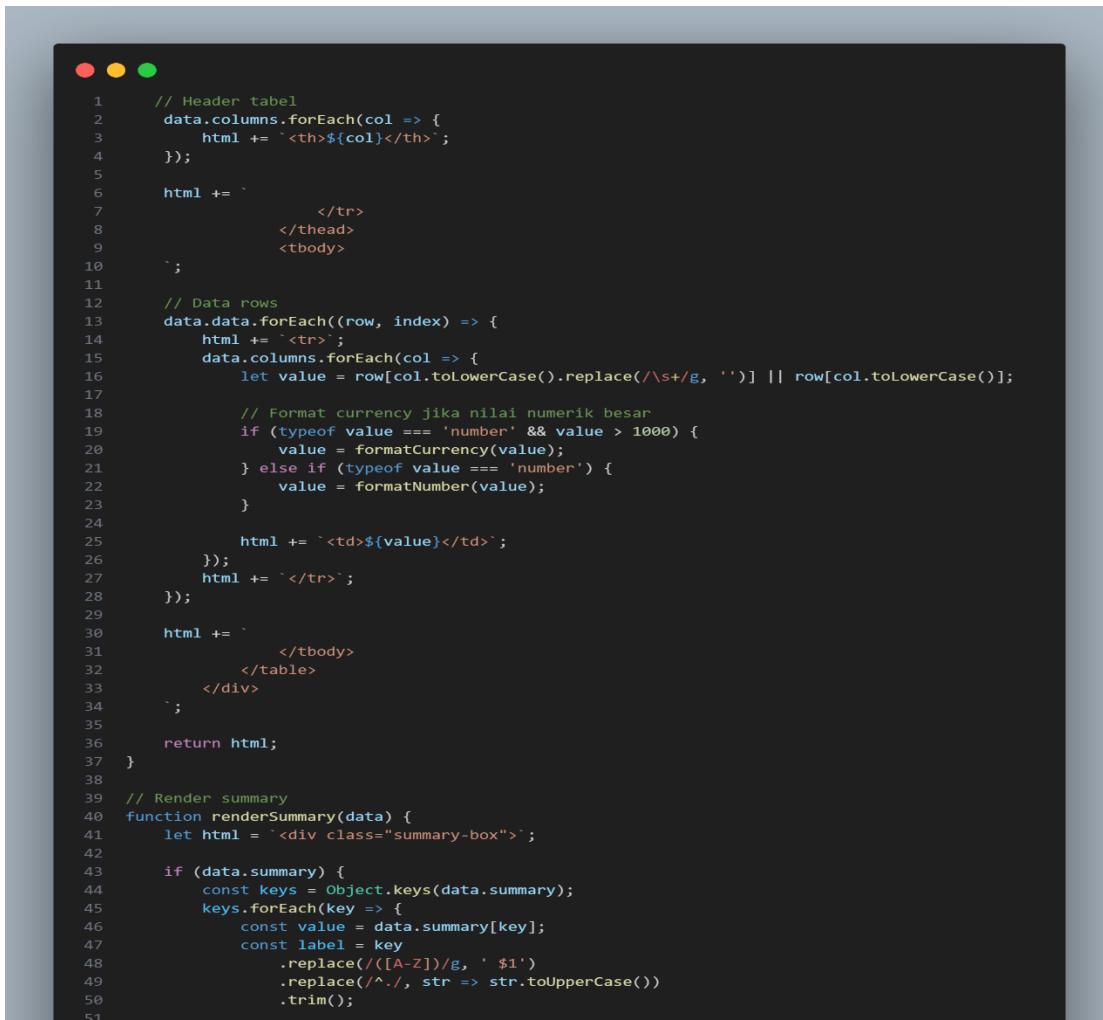
Kemudian menampilkan laporan berdasarkan tipe yang dipilih oleh pengguna. Pertama, variabel currentReport di-set ke 'sales' sebagai laporan default, kemudian fungsi showReport(reportType) digunakan untuk mengganti tipe laporan sesuai pilihan menu kemudian fungsi ini juga mengatur tampilan menu agar item yang sedang aktif diberi kelas active, dan setelah itu langsung memanggil generateReport() untuk memperbarui laporan. Selanjutnya, fungsi generateReport() bertugas mengambil data laporan dari objek reportData sesuai tipe laporan yang sedang aktif, lalu melakukan pengecekan apakah data tersedia atau tidak.

```

47         <span class="info-value">${department}</span>
48     </div>
49     <div class="info-item">
50         <span class="info-label">Periode:</span>
51         <span class="info-value">${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long' })}</span>
52     </div>
53     <div class="info-item">
54         <span class="info-label">Dicetak:</span>
55         <span class="info-value">${getCurrentDate()}</span>
56     </div>
57   </div>
58 `;
59
60 // Render tabel data
61 html += renderTable(data);
62
63 // Summary box
64 html += renderSummary(data);
65
66 // Footer dan signature
67 html += `
68     <div class="report-footer">
69         <p>Laporan ini adalah dokumen resmi dan bersifat rahasia perusahaan.</p>
70         <p>Dibuat oleh: Sistem Laporan Terpadu | Halaman: <span class="page-number">1</span></p>
71     </div>
72
73     <div class="signature-section">
74         <div class="signature-item">
75             <div class="signature-name">Manajer Divisi</div>
76             <div class="signature-date">Jakarta, ${new Date(reportDate).toLocaleDateString('id-ID', { year: 'numeric', month: 'long', day: 'numeric' })}</div>
77         </div>
78         <div class="signature-item">
79             <div class="signature-name">Kepala Departemen</div>
80             </div>
81         <div class="signature-item">
82             <div class="signature-name">Direktur Utama</div>
83             </div>
84     </div>
85   </div>
86 `;
87
88 document.getElementById('reportContainer').innerHTML = html;
89 showToast('Laporan berhasil dibuat', 'success');
90 }
91
92 // Render tabel
93 function renderTable(data) {
94     let html =
95         <div class="table-responsive">
96             <table>
97                 <thead>
98                     <tr>
99             </tr>
100

```

Kemudian pembuatan laporan mulai memasuki tahapan akhir, yaitu penyusunan tabel data, summary, serta bagian footer dan tanda tangan. Pertama, setelah informasi header laporan selesai dibuat, fungsi renderTable(data) dipanggil untuk membangun tabel utama yang berisi daftar data penjualan, inventori, atau keuangan sesuai tipe laporan kemudian hasil tabel ini ditambahkan ke variabel html yang berfungsi sebagai container keseluruhan isi laporan. Selanjutnya fungsi renderSummary(data) juga dijalankan untuk menghasilkan kotak ringkas yang menampilkan total, rata-rata, atau informasi penting lain tergantung jenis laporan, dan lalu bagian ini ikut digabungkan ke dalam struktur HTML. Setelah itu, kode menambahkan bagian footer yang berisi catatan bahwa laporan bersifat resmi, informasi halaman, serta blok tanda tangan dari Manajer Divisi, Kepala Departemen, dan Direktur Utama lengkap dengan tanggal yang diformat menggunakan toLocaleDateString agar tampil sesuai standar Indonesia. Pada akhirnya, seluruh HTML laporan tersebut dimasukkan ke elemen reportContainer dengan innerHTML, dan kemudian fungsi showToast dipanggil untuk memberi notifikasi bahwa laporan berhasil dibuat.



```
1 // Header tabel
2   data.columns.forEach(col => {
3     html += `<th>${col}</th>`;
4   });
5
6   html += `
7     </tr>
8   </thead>
9   <tbody>
10 `;
11
12 // Data rows
13 data.data.forEach((row, index) => {
14   html += `<tr>`;
15   data.columns.forEach(col => {
16     let value = row[col.toLowerCase().replace(/\s+/g, '')] || row[col.toLowerCase()];
17
18     // Format currency jika nilai numerik besar
19     if (typeof value === 'number' && value > 1000) {
20       value = formatCurrency(value);
21     } else if (typeof value === 'number') {
22       value = formatNumber(value);
23     }
24
25     html += `<td>${value}</td>`;
26   });
27   html += `</tr>`;
28 });
29
30 html += `
31   </tbody>
32 </table>
33 </div>
34 `;
35
36 return html;
37 }
38
39 // Render summary
40 function renderSummary(data) {
41   let html = `<div class="summary-box">`;
42
43   if (data.summary) {
44     const keys = Object.keys(data.summary);
45     keys.forEach(key => {
46       const value = data.summary[key];
47       const label = key
48         .replace(/([A-Z])/g, ' $1')
49         .replace(/\^/, str => str.toUpperCase())
50         .trim();
51
52       html += `<p>${label}: ${value}</p>`;
53     });
54   }
55
56   html += `</div>`;
57
58   return html;
59 }
```

Kemudian ke proses pembuatan tabel laporan secara detail, dimulai dari pembuatan header tabel hingga setiap baris data yang akan ditampilkan. Pertama, bagian header tabel dibuat dengan melakukan loop pada `data.columns`, lalu setiap kolom diubah menjadi elemen `<th>` sehingga struktur kolom bisa tercetak rapi kemudian setelah header selesai, kode melanjutkan ke bagian data rows yang melakukan iterasi pada setiap objek di dalam `data.data`. Selanjutnya, untuk setiap baris data, script melakukan loop lagi berdasarkan daftar kolom, lalu mencoba mengambil nilai sesuai nama kolom dengan cara menyesuaikan huruf kecil dan menghapus spasi agar cocok dengan key di objek dan lalu jika nilainya berupa angka besar maka otomatis diformat menjadi rupiah menggunakan `formatCurrency`, sedangkan jika hanya angka biasa maka diformat memakai `formatNumber`. Kemudian nilai tersebut dimasukkan ke dalam `<td>` sehingga membentuk sel tabel. Setelah semua baris selesai, HTML ditutup dengan tag penutup tabel dan dikembalikan. Selanjutnya, kode mengatur bagian summary melalui fungsi `renderSummary(data)`, yang akan membuat kotak ringkas berisi informasi seperti total pendapatan, total stok, margin, dan lainnya sesuai dataset laporan.

```

43     if (data.summary) {
44         const keys = Object.keys(data.summary);
45         keys.forEach(key => {
46             const value = data.summary[key];
47             const label = key
48                 .replace(/([A-Z])/g, ' $1')
49                 .replace(/^./, str => str.toUpperCase())
50                 .trim();
51
52             let displayValue = value;
53             if (typeof value === 'number' && value > 1000) {
54                 displayValue = formatCurrency(value);
55             }
56
57             html += `
58                 <div class="summary-item">
59                     <div class="summary-label">${label}</div>
60                     <div class="summary-value">${displayValue}</div>
61                 </div>
62             `;
63         });
64     }
65
66     html += `</div>`;
67     return html;
68 }
69
70 // Print laporan
71 function printReport() {
72     const printableReport = document.getElementById('printableReport');
73
74     if (!printableReport) {
75         showToast('Laporan belum dibuat', 'error');
76         return;
77     }
78
79     const printWindow = window.open('', '', 'width=800,height=600');
80     printWindow.document.write(`
81         <!DOCTYPE html>
82         <html>
83             <head>
84                 <title>Print Laporan</title>
85                 <link rel="stylesheet" href="styles.css">
86                 <style>
87                     body { margin: 0; padding: 20px; }
88                     .report { box-shadow: none; }
89                 </style>
90             </head>
91             <body>
92                 ${printableReport.innerHTML}
93             </body>
94         </html>
95     `);

```

Selanjutnya membuat tampilan ringkasan data dengan cara mengambil semua key dari data.summary, kemudian setiap key diolah menjadi label yang lebih rapi dengan mengganti format penulisan dan mengubah huruf awal menjadi kapital, kemudian nilai dari setiap key dicek apakah bertipe number dan lebih dari seribu supaya bisa diformat menjadi bentuk mata uang, setelah itu nilai yang sudah diformat ditampilkan sebagai elemen HTML yang berisi label dan value. Selanjutnya, seluruh elemen HTML yang sudah disusun digabung menjadi satu string besar dan dikembalikan untuk ditampilkan di halaman. Lalu, terdapat fungsi printReport() yang bertugas mencetak laporan dengan mengambil elemen HTML bertanda printableReport, jika elemen tersebut tidak ditemukan maka akan muncul notifikasi error.

```
96     printWindow.document.close();
97
98     setTimeout(() => {
100       printWindow.print();
101     }, 250);
102
103     showToast('Membuka dialog cetak...', 'success');
104   }
105
106 // Export ke PDF
107 function exportToPDF() {
108   const reportContainer = document.getElementById('printableReport');
109
110   if (!reportContainer) {
111     showToast('Laporan belum dibuat', 'error');
112     return;
113   }
114
115   showToast('Memproses PDF...', 'success');
116
117   const element = reportContainer;
118   const opt = {
119     margin: [10, 10, 10, 10],
120     filename: `Laporan-${currentReport}-${new Date().toISOString().split('T')[0]}.pdf`,
121     image: { type: 'jpeg', quality: 0.98 },
122     html2canvas: { scale: 2, useCORS: true, allowTaint: true },
123     jsPDF: {
124       orientation: 'portrait',
125       unit: 'mm',
126       format: 'a4',
127       compress: true
128     },
129     pagebreak: { mode: ['avoid-all', 'css', 'legacy'] }
130   };
131
132 // Gunakan html2pdf jika tersedia
133 if (typeof html2pdf !== 'undefined') {
134   html2pdf().set(opt).from(element).save();
135   showToast('PDF berhasil diunduh!', 'success');
136 } else {
```

Lalu mengekspor laporan menjadi file PDF dengan cara mengambil elemen HTML yang memiliki ID printableReport, kemudian mengecek apakah elemen tersebut ada jika tidak ditemukan maka sistem akan menampilkan notifikasi error agar pengguna tahu bahwa laporan belum dibuat. Kemudian, jika elemen ditemukan, sistem menampilkan toast bahwa proses pembuatan PDF sedang berjalan, lalu mendefinisikan berbagai pengaturan seperti margin, nama file yang otomatis menggunakan tanggal hari ini, kualitas gambar, konfigurasi html2canvas, hingga pengaturan PDF seperti orientasi portrait dan ukuran A4. Selanjutnya, kode melakukan pengecekan apakah library html2pdf tersedia jika tersedia maka library tersebut digunakan untuk mengonversi elemen HTML menjadi PDF dan langsung menyimpannya, serta menampilkan notifikasi bahwa PDF berhasil diunduh.

```

137     // Fallback: gunakan jsPDF + html2canvas
138     html2canvas(element, { scale: 2, allowTaint: true }).then(canvas => {
139         const { jsPDF } = window.jspdf;
140         const pdf = new jsPDF({
141             orientation: 'portrait',
142             unit: 'mm',
143             format: 'a4',
144             compress: true
145         });
146
147         const imgData = canvas.toDataURL('image/jpeg', 0.98);
148         const imgWidth = 190;
149         const imgHeight = (canvas.height * imgWidth) / canvas.width;
150
151         let heightLeft = imgHeight;
152         let position = 10;
153
154         pdf.addImage(imgData, 'JPEG', 10, position, imgWidth, imgHeight);
155         heightLeft -= (297 - 20);
156
157         while (heightLeft >= 0) {
158             position = heightLeft - imgHeight;
159             pdf.addPage();
160             pdf.addImage(imgData, 'JPEG', 10, position, imgWidth, imgHeight);
161             heightLeft -= 277;
162         }
163
164         pdf.save(`Laporan-${currentReport}-${new Date().toISOString().split('T')[0]}.pdf`);
165         showToast('PDF berhasil diunduh!', 'success');
166     }).catch(err => {
167         console.error('Error generating PDF:', err);
168         showToast('Gagal membuat PDF', 'error');
169     });
170 }
171 }
172
173 // Toast notification
174 function showToast(message, type = 'success') {
175     const toast = document.getElementById('toast');
176     toast.textContent = message;
177     toast.className = `toast show ${type}`;
178
179     setTimeout(() => {
180         toast.classList.remove('show');
181     }, 3000);
182 }
183
184 // Set default date to today
185 document.addEventListener('DOMContentLoaded', () => {
186     const today = new Date().toISOString().split('T')[0];
187     document.getElementById('reportDate').value = today;
188 });
189

```

Kemudian terdapat mekanisme fallback ketika library html2pdf tidak tersedia, yaitu dengan menggabungkan jsPDF dan html2canvas. Prosesnya dimulai dengan menangkap tampilan elemen HTML menggunakan html2canvas dengan skala tertentu agar hasilnya tetap tajam, kemudian hasil canvas tersebut dikonversi menjadi data gambar JPEG yang nantinya dimasukkan ke dalam dokumen PDF. Selanjutnya, script menentukan ukuran gambar, tinggi halaman yang tersisa, dan posisi penempatan gambar di dalam PDF. Setelah itu gambar pertama ditambahkan ke halaman pertama PDF, dan jika kontennya lebih panjang dari satu halaman, maka akan dibuat loop untuk menambahkan halaman baru secara otomatis sampai seluruh konten habis. Lalu, setelah PDF selesai disusun, file disimpan dengan nama yang memuat tanggal hari ini, serta muncul notifikasi bahwa PDF berhasil diunduh. Jika terjadi error selama proses pembuatan PDF, sistem menangkap error tersebut dan menampilkan toast gagal.

```
1 /* RESET */
2 * {
3   margin: 0;
4   padding: 0;
5   box-sizing: border-box;
6 }
7
8 body {
9   font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
10  background-color: #eef2f7;
11  color: #333;
12 }
13
14 /* Layout Utama */
15 .container {
16   display: flex;
17   min-height: 100vh;
18 }
19
20 /* Sidebar */
21 .sidebar {
22   width: 250px;
23   background: #d4a7d1;
24   color: white;
25   padding: 20px;
26   box-shadow: 2px 0 10px rgba(0, 0, 0, 0.1);
27   position: fixed;
28   height: 100vh;
29   overflow-y: auto;
30 }
31
32 .logo {
33   margin-bottom: 40px;
34   padding-bottom: 20px;
35   border-bottom: 1px solid rgba(255, 255, 255, 0.2);
36 }
37
38 .logo h2 {
39   font-size: 24px;
40   font-weight: 600;
41 }
42
43 .menu ul {
44   list-style: none;
45 }
46
47 .menu ul li {
48   margin-bottom: 10px;
49 }
50
51 .menu-item {
52   display: block;
53   padding: 12px 16px;
54   color: white;
55   text-decoration: none;
56   border-radius: 6px;
57   transition: all 0.3s ease;
58 }
59
60 .menu-item:hover,
61 .menu-item.active {
62   background-color: rgba(255, 255, 255, 0.15);
63   transform: translateX(>5px);
64 }
65
66 /* Main Content */
67 .main-content {
68   margin-left: 250px;
69   flex: 1;
70   padding: 0;
71 }
```

Kemudian dengan reset CSS yang berguna untuk menyamakan margin, padding, dan box-sizing di seluruh elemen supaya tampilan lebih konsisten, kemudian diikuti dengan styling dasar untuk body yang menggunakan font-family modern seperti Segoe UI serta memberikan warna background lembut agar tampilan lebih nyaman dilihat. Selanjutnya, bagian layout utama diatur menggunakan .container dengan display: flex dan tinggi minimum layar, lalu masuk ke bagian .sidebar yang memiliki lebar tetap 250px, background ungu tua, teks putih, serta efek shadow agar terlihat menonjol. Sidebar ini juga dibuat fixed agar selalu berada di sisi kiri saat user melakukan scroll. Lalu, terdapat styling untuk logo di sidebar, termasuk margin, padding bawah, garis bawah, serta ukuran font. Bagian menu juga diatur melalui .menu ul dan .menu ul li agar daftar menu tersusun rapi, setiap item menu diberi padding, warna teks putih, radius border kecil, serta efek hover dan active yang membuat elemen sedikit bergeser dan berubah background, sehingga memberikan pengalaman interaksi yang lebih responsif dan modern.

```
65  /* Main Content */
66  .main-content {
67    margin-left: 250px;
68    flex: 1;
69    padding: 0;
70  }
71
72  /* Header */
73  .header {
74    background: #0d47a1;
75    color: white;
76    padding: 30px;
77    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
78  }
79
80  .header-content h1 {
81    font-size: 28px;
82    margin-bottom: 8px;
83  }
84
85  .header-content p {
86    font-size: 14px;
87    opacity: 0.9;
88  }
89
90  /* Report Section */
91  .report-section {
92    padding: 30px;
93  }
94
95  .report-controls {
96    background: white;
97    padding: 20px;
98    border-radius: 8px;
99    margin-bottom: 30px;
100   box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);
101   display: flex;
102   gap: 20px;
103   flex-wrap: wrap;
104   align-items: flex-end;
105  }
106
107  .control-group {
108    display: flex;
109    flex-direction: column;
110    gap: 8px;
111  }
112
113  .control-group label {
114    font-weight: 600;
115    font-size: 13px;
116    color: #555;
117  }
118
119  .control-group input,
120  .control-group select {
121    padding: 10px 12px;
122    border: 1px solid #cbd5e1;
123    border-radius: 4px;
124    font-size: 14px;
125    min-width: 150px;
126    transition: border-color 0.3s ease;
127  }
128
129  .control-group input:focus,
130  .control-group select:focus {
131    outline: none;
132    border-color: #2196f3;
133    box-shadow: 0 0 3px rgba(33, 150, 243, 0.2);
134  }
135
136
137  /* Buttons */
138  .btn {
139    padding: 10px 20px;
140    border: none;
141    border-radius: 6px;
142    font-size: 14px;
143    font-weight: 600;
144    cursor: pointer;
145    transition: all 0.3s ease;
146    white-space: nowrap;
147  }
148
```

Kemudian fokus styling berpindah ke area main content, dimulai dari `.main-content` yang diberi margin kiri 250px agar tidak tertimpa sidebar, kemudian diberi fleksibilitas dengan `flex: 1` supaya area konten bisa menyesuaikan lebar layar. Selanjutnya, masuk ke bagian header yang menggunakan background ungu tua yang sama dengan sidebar agar konsisten, diberi padding, warna teks putih, dan shadow ringan supaya terlihat seperti panel yang menonjol. Di dalamnya, elemen seperti `h1` dan paragraf memiliki ukuran font berbeda untuk memberi hierarki visual yang jelas. Lalu, masuk ke bagian `.report-section` yang merupakan area untuk pengaturan laporan, diberi margin atas yang cukup agar kontennya tidak terlalu menempel. Pada `.report-controls`, kontainer kontrol laporan dibuat dengan background putih, padding besar, sudut melengkung, dan bayangan lembut agar terlihat seperti card modern. Elemen ini menggunakan `display: flex` dan mendukung `wrap` agar tetap responsif saat layar lebih kecil. Di dalamnya, ada `.control-group` yang mengatur label dan input secara kolom, kemudian setiap label diberi font tebal dan warna sedikit keabu-abuan agar lebih mudah dibaca.

OUTPUT

ReportCenter

Laporan Penjualan

Laporan Inventori

Laporan Keuangan

Laporan Stok Barang

PT. NEXUS SOLUTIONS
Departemen Gudang
Tanggal: 18 November 2025

Perusahaan: PT. NEXUS SOLUTIONS Departemen: Departemen Gudang

Periode: November 2025 Dicetak: Selasa, 18 November 2025

No	Kode Barang	Nama Barang	Stok	Min. Stok	Status
1	undefined	undefined	undefined	undefined	✓ Normal
2	undefined	undefined	undefined	undefined	⚠ Rendah

ReportCenter

Laporan Penjualan

Laporan Inventori

Laporan Keuangan

Dashboard Laporan
Kelola & cetak laporan Anda dengan format profesional

Tanggal Laporan Departemen

18/11/2025 Semua Departemen Generate Print PDF

Laporan Keuangan Kuartal

PT. NEXUS SOLUTIONS
Departemen Keuangan
Tanggal: 18 November 2025

Perusahaan: PT. NEXUS SOLUTIONS Departemen: Departemen Keuangan

Periode: November 2025 Dicetak: Selasa, 18 November 2025

No	Kategori	Periode	Nilai
1	Uang Muka	November 2025	1000000
2	Pemasukan	November 2025	5000000
3	Pengeluaran	November 2025	3000000
4	Bersih	November 2025	3500000

Laporan berhasil dibuat