

PRAKTIK PER. 4

Langkah 1 : Setup Project

Jika belum ada project

```
npm create vite@latest praktik-4 -- --template react
```

```
cd praktik-4
```

```
npm run dev
```

Langkah 2: Counter App - State Dasar

File: src/Counter.jsx

```
1 import React, { useState } from 'react';
2 import './Counter.css';
3
4 const Counter = () => {
5   // State untuk counter
6   const [count, setCount] = useState(0);
7
8   // Handler functions
9   const increment = () => {
10     setCount(count + 1);
11   };
12
13   const decrement = () => {
14     setCount(count - 1);
15   };
16
17   const reset = () => {
18     setCount(0);
19   };
20
21   // Functional update (recommended)
22   const incrementByFive = () => {
23     setCount(prevCount => prevCount + 5);
24   };
25
26   return (
27     <div className="counter-container">
28       <h2>Counter App</h2>
29       <div className="counter-display">
30         <span className="count-value">{count}</span>
31       </div>
32       <div className="counter-buttons">
33         <button onClick={decrement} className="btn btn-decrement">
34           -1
35         </button>
36         <button onClick={reset} className="btn btn-reset">
37           Reset
38         </button>
39         <button onClick={increment} className="btn btn-increment">
40           +1
41         </button>
42         <button onClick={incrementByFive} className="btn btn-increment-
43           five">
44           +5
45         </button>
46       </div>
47       <div className="counter-status">
48         <p>Status: {count > 0 ? 'Positive' : count < 0 ? 'Negative' :
49           'Zero'}</p>
50       </div>
51     </div>
52   );
53 }
```

```
48         <p>{count % 2 === 0 ? 'Genap' : 'Ganjil'}</p>
49     </div>
50 </div>
51 );
52 };
53
54 export default Counter;
```

File: src/Counter.css

```
1  .counter-container {
2      text-align: center;
3      padding: 20px;
4      max-width: 400px;
5      margin: 0 auto;
6      border: 2px solid #007bff;
7      border-radius: 10px;
8      background-color: #f8f9fa;
9  }
10
11  .counter-display {
12      margin: 20px 0;
13  }
14
15  .count-value {
16      font-size: 3rem;
17      font-weight: bold;
18      color: #007bff;
19      padding: 10px 20px;
20      border: 2px solid #dee2e6;
21      border-radius: 8px;
22      background-color: white;
23      display: inline-block;
24      min-width: 100px;
25  }
26
27  .counter-buttons {
28      display: grid;
29      grid-template-columns: repeat(2, 1fr);
30      gap: 10px;
31      margin: 20px 0;
```

```
31     margin: 20px 0;
32 }
33
34 .btn {
35     padding: 10px 15px;
36     border: none;
37     border-radius: 5px;
38     font-size: 1rem;
39     cursor: pointer;
40     transition: all 0.3s ease;
41 }
42
43 .btn:hover {
44     transform: translateY(-2px);
45     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
46 }
47
48 .btn-decrement {
49     background-color: #dc3545;
50     color: white;
51 }
52
53 .btn-reset {
54     background-color: #6c757d;
55     color: white;
56 }
57
58 .btn-increment {
59     background-color: #28a745;
60     color: white;
61 }
62
63 .btn-increment-five {
64     background-color: #007bff;
65     color: white;
66     grid-column: span 2;
67 }
```

```
67 }
68
69 .counter-status {
70     margin-top: 20px;
71     padding: 10px;
72     background-color: #e9ecef;
73     border-radius: 5px;
74 }
75
76 .counter-status p {
77     margin: 5px 0;
78     font-weight: bold;
79 }
```

Langkah 3: Form dengan Multiple State

File: src/UserForm.jsx

```
1 import React, { useState } from 'react';
2 import './UserForm.css';
3
4 const UserForm = () => {
5   // Multiple state variables
6   const [name, setName] = useState('');
7   const [email, setEmail] = useState('');
8   const [age, setAge] = useState('');
9   const [isSubscribed, setIsSubscribed] = useState(false);
10  const [users, setUsers] = useState([]);
11
12  // Form submission handler
13  const handleSubmit = (e) => {
14    e.preventDefault();
15
16    // Validasi sederhana
17    if (!name || !email || !age) {
18      alert('Harap isi semua field!');
19      return;
20    }
21
22    // Tambah user baru
23    const newUser = {
24      id: Date.now(),
25      name,
26      email,
27      age: parseInt(age),
28      isSubscribed,
29      registeredAt: new Date().toLocaleDateString()
30    };
31
32    // Update state users
33    setUsers(prevUsers => [...prevUsers, newUser]);
34
35    // Reset form
36    setName('');
37    setEmail('');
38    setAge('');
39    setIsSubscribed(false);
40  };
41
42  // Delete user
43  const deleteUser = (userId) => {
44    setUsers(prevUsers => prevUsers.filter(user => user.id !== userId));
45  };
46
47  return (
48    <div className="form-container">
49      <h2>User Registration Form</h2>
```

```

50
51 <form onSubmit={handleSubmit} className="user-form">
52   <div className="form-group">
53     <label htmlFor="name">Nama:</label>
54     <input
55       type="text"
56       id="name"
57       value={name}
58       onChange={(e) => setName(e.target.value)}
59       placeholder="Masukkan nama"
60     />
61   </div>
62
63   <div className="form-group">
64     <label htmlFor="email">Email:</label>
65     <input
66       type="email"
67       id="email"
68       value={email}
69       onChange={(e) => setEmail(e.target.value)}
70       placeholder="Masukkan email"
71     />
72   </div>
73
74   <div className="form-group">
75     <label htmlFor="age">Usia:</label>
76     <input
77       type="number"
78       id="age"
79       value={age}
80       onChange={(e) => setAge(e.target.value)}
81       placeholder="Masukkan usia"
82       min="1"
83       max="120"
84     />
85   </div>
86
87   <div className="form-group checkbox-group">
88     <label>
89       <input
90         type="checkbox"
91         checked={isSubscribed}
92         onChange={(e) => setIsSubscribed(e.target.checked)}
93       />
94       Subscribe to newsletter
95     </label>
96   </div>
97
98   <button type="submit" className="submit-btn">
99     Tambah User
100   </button>
101 </form>
102
103 {/* Display registered users */}

```



```

104     <div className="users-list">
105       <h3>Registered Users ({users.length})</h3>
106       {users.length === 0 ? (
107         <p className="no-users">Belum ada user yang terdaftar</p>
108       ) : (
109         <div className="users-grid">
110           {users.map(user => (
111             <div key={user.id} className="user-card">
112               <button
113                 className="delete-btn"
114                 onClick={() => deleteUser(user.id)}
115               >
116                 x
117             </button>
118             <h4>{user.name}</h4>
119             <p>Email: {user.email}</p>
120             <p>Usia: {user.age} tahun</p>
121             <p>Subscribed: {user.isSubscribed ? 'Y' : 'N'}
122             <small>Registered: {user.registeredAt}</small>
123           </div>
124         )]}
125       </div>
126     )}
127   </div>
128 </div>
129 );
130 };
131
132 export default UserForm;

```

File: src/UserForm.css

```
1  .form-container {
2      max-width: 800px;
3      margin: 0 auto;
4      padding: 20px;
5  }
6
7  .user-form {
8      background-color: #f8f9fa;
9      padding: 20px;
10     border-radius: 10px;
11     margin-bottom: 30px;
12     border: 1px solid #dee2e6;
13 }
14
15 .form-group {
16     margin-bottom: 15px;
17 }
18
19 .form-group label {
20     display: block;
21     margin-bottom: 5px;
22     font-weight: bold;
23     color: #495057;
24 }
25
26 .form-group input[type="text"],
27 .form-group input[type="email"],
28 .form-group input[type="number"] {
29     width: 100%;
30     padding: 10px;
31     border: 1px solid #ced4da;
32     border-radius: 5px;
33     font-size: 1rem;
34     box-sizing: border-box;
35 }
36
37 .form-group input:focus {
38     outline: none;
39     border-color: #007bff;
40     box-shadow: 0 0 0 2px rgba(0, 123, 255, 0.25);
41 }
42
43 .checkbox-group label {
44     display: flex;
45     align-items: center;
46     gap: 8px;
47     cursor: pointer;
48 }
49
```

```
50 .checkbox-group input[type="checkbox"] {
51     width: auto;
52 }
53
54 .submit-btn {
55     background-color: #007bff;
56     color: white;
57     padding: 12px 24px;
58     border: none;
59     border-radius: 5px;
60     font-size: 1rem;
61     cursor: pointer;
62     transition: background-color 0.3s ease;
63 }
64
65 .submit-btn:hover {
66     background-color: #0056b3;
67 }
68
69 .users-list {
70     margin-top: 30px;
71 }
72
73 .no-users {
74     text-align: center;
75     color: #6c757d;
76     font-style: italic;
77     padding: 20px;
78 }
79
80 .users-grid {
81     display: grid;
82     grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
83     gap: 15px;
84     margin-top: 15px;
85 }
86
87 .user-card {
88     background-color: white;
89     border: 1px solid #dee2e6;
90     border-radius: 8px;
91     padding: 15px;
92     position: relative;
93     box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
94     transition: transform 0.2s ease;
95 }
96
97 .user-card:hover {
98     transform: translateY(-2px);
99     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.15);
100 }
101
102 .delete-btn {
103     position: absolute;
```

```
104     top: 10px;
105     right: 10px;
106     background-color: #dc3545;
107     color: white;
108     border: none;
109     border-radius: 50%;
110     width: 25px;
111     height: 25px;
112     cursor: pointer;
113     font-size: 14px;
114     display: flex;
115     align-items: center;
116     justify-content: center;
117 }
118
119 .user-card h4 {
120     margin: 0 0 10px 0;
121     color: #007bff;
122 }
123
124 .user-card p {
125     margin: 5px 0;
126     font-size: 0.9rem;
127 }
128
129 .user-card small {
130     color: #6c757d;
131     font-size: 0.8rem;
132 }
```

Langkah 4: App.js Utama

File: src/App.jsx

```
1 import React, { useState } from 'react';
2 import './App.css';
3 import Counter from './Counter';
4 import UserForm from './UserForm';
5
6 function App() {
7   const [currentView, setCurrentView] = useState('counter');
8
9   return (
10     <div className="App">
11       <header className="App-header">
12         <h1>State Management dengan useState</h1>
13         <nav className="nav-tabs">
14           <button
15             onClick={() => setCurrentView('counter')}
16             className={currentView === 'counter' ? 'active' : ''}
17           >
18             Counter App
19           </button>
20           <button
21             onClick={() => setCurrentView('form')}
22             className={currentView === 'form' ? 'active' : ''}
23           >
24             User Form
25           </button>
26         </nav>
27       </header>
28
29       <main className="App-main">
30         {currentView === 'counter' && <Counter />}
31         {currentView === 'form' && <UserForm />}
32       </main>
33
34       <footer className="App-footer">
35         <p>Pertemuan 4 - State Management Dasar</p>
36       </footer>
37     </div>
38   );
39 }
40
41 export default App;
```

```
1  .App {
2    text-align: center;
3    min-height: 100vh;
4    background-color: #f5f5f5;
5  }
6
7  .App-header {
8    background-color: #282c34;
9    padding: 20px;
10   color: white;
11 }
12
13 .App-header h1 {
14   margin: 0 0 20px 0;
15   font-size: 2rem;
16 }
17
18 .nav-tabs {
19   display: flex;
20   justify-content: center;
21   gap: 10px;
22 }
23
24 .nav-tabs button {
25   padding: 10px 20px;
26   border: none;
27   border-radius: 5px;
28   background-color: #495057;
29   color: white;
30   cursor: pointer;
31   transition: background-color 0.3s ease;
32 }
33
34 .nav-tabs button.active {
35   background-color: #007bff;
36 }
37
38 .nav-tabs button:active {
```

```

38 .nav-tabs button:hover:not(.active) {
39   background-color: #6c757d;
40 }
41
42 .App-main {
43   padding: 20px;
44   min-height: calc(100vh - 200px);
45 }
46
47 .App-footer {
48   background-color: #343a40;
49   color: white;
50   padding: 10px;
51   position: relative;
52   bottom: 0;
53   width: 100%;
54 }

```

File: src/main.jsx

```

main.jsx
1  import { StrictMode } from 'react'
2  import { createRoot } from 'react-dom/client'
3  // import './index.css'
4  import App from './App.jsx'
5
6  createRoot(document.getElementById('root')).render(
7    <StrictMode>
8      <App />
9    </StrictMode>,
10 )
11

```

State Management dengan useState

Counter App User Form

Counter App

0

-1

Reset

+1

+5

Status: Zero
Genap

State Management dengan useState

Counter App

User Form

User Registration Form

Nama:

Email:

Usia:

☐ Subscribe to newsletter

Tambah User

Registered Users (0)

Belum ada user yang terdaftar

Langkah 5: Latihan Mandiri

Tugas untuk Mahasiswa:

Buat komponen `TodoList` dengan fitur:

1. **Input** untuk menambah todo baru
2. **List** todo items dengan checkbox
3. **Filter** untuk show all/active/completed
4. **Counter** total todos dan active todos

Requirements:

- Gunakan `useState` untuk todos array
- Gunakan `useState` untuk filter state
- Implement `add`, `delete`, `toggle complete`
- Show statistics

Starter Code:

```
const TodoList = () => {  
  const [todos, setTodos] = useState([]);  
  const [inputValue, setInputValue] = useState('');  
  const [filter, setFilter] = useState('all'); // all, active, completed  
  
  // Implement functions here...  
  
  return (  
    <div>  
      {/* Implement UI here */}  
    </div>  
  );  
};
```