

Pemrograman Web Modern

Insan Taufik, S.Kom., M.Kom.



Penanganan Event dan Form

Sub-Materi

Penanganan event pada React (misalnya onClick, onChange). Perbedaan penanganan event di React dan HTML biasa. Mengelola input form menggunakan Controlled Components. Validasi form sederhana.

Tujuan Pembelajaran (Capaian)

Mampu membuat elemen UI merespons interaksi pengguna dan mengelola data input form secara efektif.

Apa itu Event Handling di React?

Konsep Dasar:

- Event = Aksi yang dilakukan user (klik, ketik, submit, dll)
- Handler = Function yang menangani event
- React menggunakan Synthetic Event (wrapper browser events)

```
const Button = () => {  
  const handleClick = (event) => {  
    event.preventDefault();  
    alert('Button diklik!');  
  };  
  
  return (  
    <button onClick={handleClick}>  
      Klik Saya  
    </button>  
  );  
};
```

Perbedaan React vs HTML Event Handling

Add a little bit of body text

Aspek	HTML (Vanilla JS)	React
Contoh kode	html <button onclick="handleClick()">Klik</button>	jsx <button onClick={handleClick}>Klik</button>
Nama event	lowercase → onclick	camelCase → onClick
Handler	String (kode JS langsung)	Function reference (fungsi JS)
Prevent default	return false	event.preventDefault()
Aspek	HTML (Vanilla JS)	React

Common Events di React

React menyediakan berbagai event handler yang mirip seperti di HTML, tapi dengan penulisan camelCase dan berbentuk fungsi JavaScript, bukan string. Event-event ini dibungkus dalam objek `SyntheticEvent`, agar bisa bekerja konsisten di semua browser.

```
// Event pada form dan input
<input
  onChange={handleChange}      // Perubahan nilai
  onFocus={handleFocus}        // Fokus ke element
  onBlur={handleBlur}          // Keluar dari element
/>

<form onSubmit={handleSubmit}> // Submit form
  <button type="submit">Kirim</button>
</form>

// Event mouse
<button
  onClick={handleClick}          // Klik kiri
  onDoubleClick={handleDoubleClick} // Double click
  onContextMenu={handleContextMenu} // Klik kanan
>
  Interaksi Mouse
</button>

// Event keyboard
<input
  onKeyDown={handleKeyDown}      // Tombol ditekan
  onKeyUp={handleKeyUp}           // Tombol dilepas
/>
```

useState Hook untuk State Management

Konsep useState:

- useState returns [currentValue, setterFunction]
- State bersifat immutable (gunakan setter untuk update)
- Component re-render ketika state berubah

```
import { useState } from 'react';
```

```
const Counter = () => {  
  const [count, setCount] = useState(0);  
  const increment = () => {  
    setCount(count + 1);  
  };  
  
  return (  
    <div>  
      <p>Count: {count}</p>  
      <button onClick={increment}>Tambah</button>  
    </div>  
  );  
};
```

Controlled Components

Karakteristik Controlled Components:

- Nilai input dikontrol oleh React state
- Single source of truth
- Data flow: State → Value → onChange → State

```
const LoginForm = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleEmailChange = (event) => {
    setEmail(event.target.value);
  };

  return (
    <form>
      <input
        type="email"
        value={email}
        onChange={handleEmailChange}
        placeholder="Masukkan email"
      />
      <input
        type="password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        placeholder="Masukkan password"
      />
    </form>
  );
};
```

Form Handling dengan Multiple Inputs

```
const RegistrationForm = () => {
  const [formData, setFormData] = useState({
    firstName: '', lastName: '', email: '', password: ''
  });

  const handleInputChange = (event) => {
    const { name, value } = event.target;

    setFormData(prevData => ({ ...prevData, [name]: value }));
  };

  return (
    <form>
      <input name="firstName" value={formData.firstName} onChange={handleInputChange} />
      <input name="lastName" value={formData.lastName} onChange={handleInputChange} />
    </form>
  );
};
```


Form Handling dengan Multiple Inputs

```
const RegistrationForm = () => {
  const [formData, setFormData] = useState({
    firstName: '', lastName: '', email: '', password: ''
  });

  const handleInputChange = (event) => {
    const { name, value } = event.target;
    setFormData(prevData => ({...prevData, [name]: value}));
  };

  return (
    <form>
      <input name="firstName" value={formData.firstName} onChange={handleInputChange} />
      <input name="lastName" value={formData.lastName} onChange={handleInputChange} />
    </form>
  );
};
```

Validasi Form Sederhana

```
const ContactForm = () => {
  const [errors, setErrors] = useState({});

  const validateForm = (data) => {
    const newErrors = {};

    if (!data.name.trim()) {
      newErrors.name = 'Nama harus diisi';
    }

    if (!data.email) {
      newErrors.email = 'Email harus diisi';
    } else if (!/\S+@\S+\.\S+/.test(data.email)) {
      newErrors.email = 'Format email tidak valid';
    }

    return newErrors;
  };

  const handleSubmit = (event) => {
    event.preventDefault();
    const formErrors = validateForm(formData);

    if (Object.keys(formErrors).length === 0) {
      // Submit form
    } else {
      setErrors(formErrors);
    }
  };
};
```

Ringkasan

Yang telah dipelajari:

- ✓ Penanganan event di React (onClick, onChange, onSubmit)
- ✓ Perbedaan React vs HTML event handling
- ✓ useState Hook untuk mengelola state
- ✓ Controlled components untuk form handling
- ✓ Validasi form sederhana