

## JOB SHEET

**Mata Pelajaran** : Rekayasa Perangkat Lunak  
**Elemen** : Pemrograman Web  
**Kelas/Semester** : XI RPL 3 / Ganjil  
**Topik** : Interpolation, Data Binding, dan Computed Property VueJS  
**Nama** : Abiyya Hamdan Nurwandha  
**Kelas** : XI RPL III

### D. Interpolation

Menampilkan Teks	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Interpolation &amp; Teks&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue@2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;h1&gt;{{ message }}&lt;/h1&gt;     &lt;p&gt;Selamat datang di {{ course }}&lt;/p&gt;   &lt;/div&gt;    &lt;script&gt;     new Vue({       el: '#app',       data: {         message: 'Halo Vue.js!',         course: 'Pemrograman Web'       }     })   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<p style="text-align: center;"><b>Halo Vue.js!</b></p> <p style="text-align: center;">Selamat datang di Pemrograman Web</p>

Menghitung dengan Ekspresi	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;</pre>	<p style="text-align: center;">Umur saya: 17</p> <p style="text-align: center;">Tahun lahir saya: 2008</p>

```

<title>Interpolation 2</title>
<script
src="https://cdn.jsdelivr.net/npm/vue@2"></script>
</head>
<body>
  <div id="app">
    <p>Umur saya: {{ age }}</p>
    <p>Tahun lahir saya: {{ currentYear - age }}</p>
  </div>

  <script>
    new Vue({
      el: '#app',
      data: {
        age: 17,
        currentYear: 2025
      }
    });
  </script>
</body>
</html>

```

### Menggabungkan String

Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Interpolation 3&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue@2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;p&gt;Halo, nama saya {{ firstName + " " + lastName }}&lt;/p&gt;   &lt;/div&gt;    &lt;script&gt;     new Vue({       el: '#app',       data: {         firstName: "Andi",         lastName: "Saputra"       }     });   &lt;/script&gt; </pre>	<p>Halo, nama saya Andi Saputra</p>

```
    });
  </script>
</body>
</html>
```

## E. Data Binding

Binding ke Atribut HTML ( <b>v-bind:href</b> )	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Data Binding 1&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue@2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;h2&gt;Contoh v-bind:href&lt;/h2&gt;     &lt;a v-bind:href="website"&gt;Kunjungi Website Vue.js&lt;/a&gt;     &lt;!-- cara singkat --&gt;     &lt;a :href="website"&gt;Link dengan sintaks singkat&lt;/a&gt;   &lt;/div&gt;    &lt;script&gt;     new Vue({       el: '#app',       data: {         website: "https://vuejs.org"       }     });   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Contoh v-bind:href</b></p> <p><a href="#">Kunjungi Website Vue.js Link dengan sintaks singkat</a></p>

Binding ke Class	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Data Binding 2&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue @2"&gt;&lt;/script&gt;   &lt;style&gt;     .aktif { color: green; font-weight: bold; }     .non-aktif { color: red; }   &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;h2&gt;Contoh v-bind:class&lt;/h2&gt;     &lt;p :class="isActive ? 'aktif' : 'non-aktif'"&gt;Status User&lt;/p&gt;   &lt;/div&gt;  &lt;script&gt;   new Vue({     el: '#app',     data: {       isActive: true     }   }); &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Contoh v-bind:class</b></p> <p><b>Status User</b></p>

Binding ke Style	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Data Binding 3&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue @2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;</pre>	<p><b>Contoh v-bind:style</b></p> <p><b>Teks dengan gaya dinamis</b></p>

```

<div id="app">
  <h2>Contoh v-bind:style</h2>
  <p :style="{ color: textColor,
fontSize: fontSize + 'px' }">
    Teks dengan gaya dinamis
  </p>
</div>

<script>
  new Vue({
    el: '#app',
    data: {
      textColor: "blue",
      fontSize: 24
    }
  });
</script>
</body>
</html>

```

## #Notes

Jadi urutannya:

1. **v-bind:href** → untuk atribut.
2. **v-bind:class** → untuk kondisi class CSS.
3. **v-bind:style** → untuk manipulasi style langsung.

## F. Computed Property

Menggabungkan Nama Lengkap	
Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Computed 1&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue@2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;h2&gt;Nama Lengkap: {{ fullName }}&lt;/h2&gt;   &lt;/div&gt;  &lt;script&gt;   new Vue({     el: '#app',     computed: {       getFullName() {         return this.firstName + ' ' + this.lastName       }     }   }) &lt;/script&gt; </pre>	<b>Nama Lengkap: Andi Saputra</b>

```

        data: {
            firstName: "Andi",
            lastName: "Saputra"
        },
        computed: {
            fullName: function() {
                return this.firstName + " "
+ this.lastName;
            }
        );
    </script>
</body>
</html>

```

**Keterangan:** fullName otomatis diperbarui jika firstName atau lastName berubah.

### Menghitung Tahun Lahir dari Umur

Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;     &lt;meta charset="UTF-8"&gt;     &lt;title&gt;Computed 2&lt;/title&gt;     &lt;script src="https://cdn.jsdelivr.net/npm/vue @2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;     &lt;div id="app"&gt;         &lt;p&gt;Umur saya: {{ age }}&lt;/p&gt;         &lt;p&gt;Tahun lahir saya: {{ birthYear } }&lt;/p&gt;     &lt;/div&gt;      &lt;script&gt;         new Vue({             el: '#app',             data: {                 age: 17,                 currentYear: 2025             },             computed: {                 birthYear: function() {                     return this.currentYear - this.age;                 }             }         })     &lt;/script&gt; </pre>	<p>Umur saya: 17</p> <p>Tahun lahir saya: 2008</p>

```

    });
</script>
</body>
</html>

```

**Keterangan:** birthYear dihitung otomatis dari age dan currentYear.

Menghitung Panjang Pesan	
Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;title&gt;Computed 3&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue @2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;p&gt;Pesan: "{{ message }}"&lt;/p&gt;     &lt;p&gt;Panjang Pesan: {{ messageLength }} karakter&lt;/p&gt;   &lt;/div&gt;    &lt;script&gt;     new Vue({       el: '#app',       data: {         message: "Halo Vue.js"       },       computed: {         messageLength: function() {           return this.message.length;         }       });     &lt;/script&gt; &lt;/body&gt; &lt;/html&gt; </pre>	<p>Pesan: "Halo Vue.js"</p> <p>Panjang Pesan: 11 karakter</p>

**Keterangan:** messageLength selalu menyesuaikan dengan isi message.

Dengan 3 praktikum ini, siswa bisa belajar bahwa computed property:

1. Bisa digunakan untuk menggabungkan data.
2. Bisa dipakai untuk perhitungan matematis.
3. Bisa dipakai untuk analisis data (misalnya panjang string).

## Membalikkan Teks dengan methods vs computed

methods-vs-computed.html	
Pseudocode	Output
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8" /&gt;   &lt;title&gt;Methods vs Computed&lt;/title&gt;   &lt;script src="https://cdn.jsdelivr.net/npm/vue@2"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="app"&gt;     &lt;h2&gt;Teks Asli: {{ message }}&lt;/h2&gt;      &lt;!-- Menggunakan Computed --&gt;     &lt;p&gt;&lt;b&gt;Computed:&lt;/b&gt; {{ reversedByComputed }}&lt;/p&gt;      &lt;!-- Menggunakan Methods --&gt;     &lt;p&gt;&lt;b&gt;Methods:&lt;/b&gt; {{ reversedByMethod() }}&lt;/p&gt;      &lt;!-- Input untuk mengubah teks --&gt;     &lt;input v-model="message" placeholder="Ubah teks di sini" /&gt;   &lt;/div&gt;  &lt;script&gt;   new Vue({     el: '#app',     data: {       message: "Halo Vue.js"     },     computed: {       reversedByComputed: function() {         console.log('computed dijalankan');         return this.message.split('').reverse().join('');       }     },     methods: {       reversedByMethod: function() {         console.log('method')       }     }   }) </pre>	<p><b>Teks Asli:</b> Halo Vue.js</p> <p><b>Computed:</b> olaH euv.sj</p> <p><b>Methods:</b> olaH euv.sj</p> <div style="border: 1px solid black; padding: 5px; text-align: center;">Halo Vue.js</div>

```

dijalankan');
        return
this.message.split(' ').reverse().join
(' ');
    }
}
);
</script>
</body>
</html>

```

### Perbedaan yang terlihat:

#### 1. Computed Property

- o Dihitung sekali saja dan hasilnya di-cache sampai data (message) berubah.
- o Pada contoh di atas, reversedByComputed hanya dihitung ulang kalau message berubah.

#### 2. Methods

- o Selalu dieksekusi ulang setiap kali dipanggil saat rendering.
- o Pada contoh di atas, reversedByMethod() akan selalu dipanggil ulang meskipun message tidak berubah.

### Kalau dicek di Console (F12), akan terlihat log:

&quot;Computed dijalankan&quot; muncul hanya saat message diubah.

&quot;Method dijalankan&quot; muncul setiap kali halaman di-render.

### 1. Praktekkan interpolation, data binding, dan computed property dengan 3 file yang berbeda.

computed-property.html	
Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;     &lt;meta charset="UTF-8"&gt;     &lt;title&gt;Vue Computed Property&lt;/title&gt;     &lt;script src="https://unpkg.com/vue@3/dist/vue .global.js"&gt;&lt;/script&gt;     &lt;style&gt;         body { font-family: Arial; padding: 20px; }         .box { background: #fff0f6; padding: 15px; margin: 10px 0; border-radius: 5px; }         input { padding: 8px; margin: 5px 0; width: 100%; max-width: 200px; }     &lt;/style&gt; &lt;/head&gt; &lt;body&gt; </pre>	<p> <b>Computed Property Example</b></p> <div style="background-color: #fff0f6; padding: 10px; border-radius: 5px; display: flex; justify-content: space-between;"> <input type="text" value="Abiyya"/> <input type="text" value="Hamdan"/> </div> <p><b>Nama Lengkap:</b> Abiyya Hamdan</p> <p><i>Perhatikan console — computed hanya jalan saat data berubah!</i></p>

```

<div id="app">
    <h2>Computed Property
Example</h2>
    <div class="box">
        <input v-model="firstName"
placeholder="Nama Depan" />
        <input v-model="lastName"
placeholder="Nama Belakang" />
        <p><strong>Nama
Lengkap:</strong> {{ fullName }}</p>
        <p><em>Perhatikan console –
computed hanya jalan saat data
berubah!</em></p>
    </div>
</div>

<script>
    const { createApp } = Vue;

    createApp({
        data() {
            return {
                firstName: '',
                lastName: ''
            }
        },
        computed: {
            fullName() {
                console.log('Computed:
fullName dihitung ulang');
                return `${this.firstName}
${this.lastName}`.trim();
            }
        }
    }).mount('#app');
</script>
</body>
</html>

```

**data-binding.html**

<b>Pseudocode</b>	<b>Output</b>
-------------------	---------------

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Vue Data Binding</title>
    <script
src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
    <style>
        body { font-family: Arial;
padding: 20px; }
        .box { background: #e8f4fd;
padding: 15px; margin: 10px 0;
border-radius: 5px; }
        input { padding: 8px; margin: 5px 0; width: 100%; max-width: 300px; }
        button { padding: 8px 16px;
margin-top: 10px; }
    </style>
</head>
<body>
    <div id="app">
        <h2>🔗 Data Binding Example</h2>
        <div class="box">
            <input v-model="inputText"
placeholder="Ketik sesuatu..." />
            <p>Anda mengetik: <strong>{{ inputText }}</strong></p>

            <button
:disabled="!inputText">Submit</button>
            <p :style="{ color: textColor }">Warna teks ini berubah berdasarkan panjang input.</p>
        </div>
    </div>

    <script>
        const { createApp } = Vue;

        createApp({
            data() {
                return {
                    inputText: ''
                }
            },
            computed: {
                textColor() {
                    return
this.inputText.length > 5 ? 'green' :
'red';
                }
            }
        }).mount('#app');
    </script>

```

## 🔗 Data Binding Example

Haloo World

Anda mengetik: **Haloo World**

Submit

Warna teks ini berubah berdasarkan panjang input.

## 🔗 Data Binding Example

Haloo

Anda mengetik: **Haloo**

Submit

Warna teks ini berubah berdasarkan panjang input.

```

        }
    }
}) .mount('#app');
</script>
</body>
</html>

```

## interpolation.html

Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;     &lt;meta charset="UTF-8"&gt;     &lt;title&gt;Vue Interpolation&lt;/title&gt;     &lt;script src="https://unpkg.com/vue@3/dist/vue.global.js"&gt;&lt;/script&gt;     &lt;style&gt;         body { font-family: Arial; padding: 20px; }         .box { background: #f0f0f0; padding: 15px; margin: 10px 0; border-radius: 5px; }     &lt;/style&gt; &lt;/head&gt; &lt;body&gt;     &lt;div id="app"&gt;         &lt;h2&gt;✨ Interpolation Example&lt;/h2&gt;         &lt;div class="box"&gt;             &lt;p&gt;Halo, {{ nama }}!&lt;/p&gt;             &lt;p&gt;5 + 3 = {{ 5 + 3 }}&lt;/p&gt;             &lt;p&gt;Pesan: {{ message.toUpperCase() }}&lt;/p&gt;         &lt;/div&gt;     &lt;/div&gt;      &lt;script&gt;         const { createApp } = Vue;          createApp({             data() {                 return {                     nama: 'Pengguna Vue',                     message: 'selamat belajar vue.js'                 }             }         }) .mount('#app');     </pre>	<p style="text-align: center;"><b>✨ Interpolation Example</b></p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Halo, Pengguna Vue!</p> <p><math>5 + 3 = 8</math></p> <p>Pesan: SELAMAT BELAJAR VUE.JS</p> </div>

```

</script>
</body>
</html>

```

## Code D, E, F Mengembalikan Teks, Note Di gabungkan ( project )

Index.html	
Pseudocode	Output
<pre> &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;     &lt;meta charset="UTF-8" /&gt;     &lt;meta name="viewport" content="width=device-width, initial-scale=1.0" /&gt;     &lt;title&gt;Vue.js - D, E, F &amp; Mengembalikan Teks&lt;/title&gt;     &lt;script src="https://unpkg.com/vue@3/dist/vue .global.js"&gt;&lt;/script&gt;     &lt;style&gt;         body {             font-family: 'Segoe UI',             Tahoma, Geneva, Verdana, sans-serif;             background: linear-gradient(135deg, #667eea, #764ba2);             padding: 30px;             color: #333;         }          .container {             max-width: 900px;             margin: 0 auto;         }          h1 {             text-align: center;             color: white;             margin-bottom: 40px;             text-shadow: 1px 1px 3px             rgba(0,0,0,0.2);         }          .section {             background: white;             border-radius: 12px;             padding: 25px;         }     </pre>	<p>The screenshot displays six separate browser windows, each showing a different part of the Vue.js application. The windows are arranged vertically and show the following:</p> <ul style="list-style-type: none"> <li><b>Section D:</b> Interpolation — Menampilkan Data &amp; Ekspresi. Shows "Hello, Developer Hebat!" and the result of the expression <math>10 + 20 = 30</math>. A note at the bottom says "Teks adil: 'vue itu keren' → jadi huruf besar: VUE ITU KEREN".</li> <li><b>Section E:</b> Data Binding — Sinkronisasi Otomatis Input &amp; Tampilan. Shows a text input field with "Klik apapun di sini...", a message input field with "Kamu mengetik: ''", and a button labeled "Kirim". Below it, a note says "Panjang teks: 0 karakter → Status: TERLAAU PENDEK".</li> <li><b>Section E:</b> Data Binding — Sinkronisasi Otomatis Input &amp; Tampilan. Shows a text input field with "Hello". Below it, a note says "Kamu mengetik: "Hello"" and "Panjang teks: 6 karakter → Status: CUKUP PANJANG".</li> <li><b>Section F:</b> Computed Property — Nilai Turunan yang Efisien. Shows two input fields for "Nama Depan" and "Nama Belakang". Below them, a note says "Nama Lengkap (Computed):" and "Nama Lengkap (Computed): Abijaya Hamdan". A note at the bottom says "Nama lengkap Anda panjang: 14 karakter".</li> <li><b>Section F:</b> Computed Property — Nilai Turunan yang Efisien. Shows two input fields for "Abijaya" and "Hamdan". Below them, a note says "Nama Lengkap (Computed): Abijaya Hamdan" and "Nama lengkap Anda panjang: 14 karakter". A note at the bottom says "Nama lengkap Anda panjang: 14 karakter".</li> <li><b>Section F:</b> Menampilkan Teks Hasil Proses (Interpolation + Computed). Shows a note "Teks akhir yang dikembalikan: "Hello Abijaya Hamdan, kamu mengetik: Hello"". Below it, a note says "Status Input: SUDAH DIISI".</li> </ul>

```
margin-bottom: 30px;
box-shadow: 0 5px 15px
rgba(0,0,0,0.1);
}

h2 {
color: #2c3e50;
margin-top: 0;
padding-bottom: 10px;
border-bottom: 2px solid #eee;
}

.tag {
display: inline-block;
background: #3498db;
color: white;
padding: 4px 10px;
border-radius: 20px;
font-size: 14px;
margin-bottom: 15px;
}

input {
padding: 12px;
margin: 8px 0;
width: 100%;
max-width: 350px;
border: 2px solid #ddd;
border-radius: 8px;
font-size: 16px;
}

input:focus {
outline: none;
border-color: #3498db;
}

button {
padding: 12px 24px;
background: #27ae60;
color: white;
border: none;
border-radius: 8px;
cursor: pointer;
font-size: 16px;
margin-top: 10px;
transition: background 0.3s;
}

button:hover:not(:disabled) {
background: #2ecc71;
}
```

```
button:disabled {  
    background: #bdc3c7;  
    cursor: not-allowed;  
}  
  
.output {  
    background: #f8f9fa;  
    padding: 15px;  
    border-left: 4px solid #3498db;  
    margin: 15px 0;  
    border-radius: 0 8px 8px 0;  
    font-size: 18px;  
}  
  
.note {  
    font-size: 14px;  
    color: #7f8c8d;  
    font-style: italic;  
    margin-top: 10px;  
}  
</style>  
</head>  
<body>  
    <div id="app">  
        <h1>Vue.js: Bagian D, E, F &  
Mengembalikan Teks</h1>  
        <div class="container">  
  
            <!-- D. INTERPOLATION -->  
            <div class="section">  
                <span class="tag">Bagian  
D</span>  
                <h2>Interpolation –  
Menampilkan Data & Ekspresi</h2>  
                <div class="output">  
                    Halo, <strong>{{ nama  
} }</strong>! 🎉  
                </div>  
                <div class="output">  
                    10 + 20 = <strong>{{ 10 +  
20 }}</strong>  
                </div>  
                <div class="output">  
                    Teks asli: "{{ pesan }}" →  
jadi huruf besar: <strong>{{  
pesan.toUpperCase() }}</strong>  
                </div>  
            </div>  
  
            <!-- E. DATA BINDING -->  
            <div class="section">
```

```
<span class="tag">Bagian  
E</span>  
    <h2>Data Binding –  
Sinkronisasi Otomatis Input &  
Tampilan</h2>  
    <input v-model="teksInput"  
placeholder="Ketik apapun di sini..."  
/>  
    <div class="output">  
        Kamu mengetik: <strong>{{  
teksInput }}</strong>  
    </div>  
    <button  
:disabled="!teksInput">Kirim</button>  
    <div class="output" :style="{'  
color: warnaTeks, fontWeight: 'bold'  
}">  
        Panjang teks: {{  
teksInput.length }} karakter →  
        Status: <span  
v-if="teksInput.length > 5">CUKUP  
PANJANG 😊</span>  
        <span  
v-else>TERLALU PENDEK 😱</span>  
    </div>  
    </div>  
  
<!-- F. COMPUTED PROPERTY -->  
<div class="section">  
    <span class="tag">Bagian  
F</span>  
    <h2>Computed Property – Nilai  
Turunan yang Efisien</h2>  
    <input v-model="namaDepan"  
placeholder="Nama Depan" />  
    <input v-model="namaBelakang"  
placeholder="Nama Belakang" />  
    <div class="output">  
        <strong>Nama Lengkap  
(Computed):</strong> <span  
style="font-size: 1.3em;">{{  
namaLengkap }}</span>  
    </div>  
    <div class="output"  
v-if="namaLengkapPanjang">  
        ✓ Nama lengkap Anda  
panjang: <strong>{{  
namaLengkapPanjang }}  
karakter</strong>  
    </div>  
    <p class="note">💡 Buka  
Console Browser – computed hanya
```

```
dijalankan saat datanya berubah!</p>
</div>

<!-- 📄 MENGEMBALIKAN TEKS -->
<div class="section">
    <span
class="tag">Mengembalikan Teks</span>
        <h2>Menampilkan Teks Hasil
Proses (Interpolation +
Computed)</h2>
        <div class="output">
            ⚡ Teks akhir yang
dikembalikan:
            <blockquote style="margin:
10px 0; font-size: 1.1em; padding:
10px; background: #e3f2fd;
border-radius: 8px;">
                "Halo {{ namaLengkap ||

'Teman' }}, kamu mengetik: <em>{{

teksInput || '-' }}</em>">
            </blockquote>
        </div>
        <div class="output" :style="{
backgroundColor: teksInput ?
'#e8f5e8' : '#ffeb3b', color:
teksInput ? '#2e7d32' : '#c62828'">
            Status Input: <strong>{{

teksInput ? 'SUDAH DIISI' : 'KOSONG'
}}</strong>
        </div>
    </div>
</div>

<script>
    const { createApp } = Vue;

    createApp({
        data() {
            return {
                // Untuk Interpolation (D)
                nama: 'Developer Hebat',
                pesan: 'vue itu keren',

                // Untuk Data Binding (E)
                teksInput: '',

                // Untuk Computed Property
(F)
                namaDepan: '',
                namaBelakang: ''
            }
        }
    }).mount('#app')

```

```
    } ;

} ,


computed: {
    // Untuk Data Binding (E)
    warnaTeks() {
        return
this.teksInput.length > 5 ? '#27ae60'
: '#e74c3c';
    },


    // Untuk Computed Property
(F)

namaLengkap() {
    console.log('[Computed]
Menghitung namaLengkap...');

    return `${this.namaDepan}
${this.namaBelakang}`.trim();
},


    // Tambahan computed untuk
"Menyalin Teks"
    namaLengkapPanjang() {
        const len =
this.namaLengkap.length;
        return len > 0 ? len :
null;
    }
}

).mount('#app');

</script>
</body>
</html>
```