

# Vue.js Component

By Mulia Nasution (mul14@refactory.id)



What is Component?



Component are reusable Vue instances



# Component Example

```
<div>  
  <TopBar />  
  <SideBar />  
</div>
```

```
<div>  
  <head-bar></head-bar>  
  <side-bar></side-bar>  
</div>
```

# Illustration

<App>

<Topbar />

<Sidebar />

</App>

<Main>

<Article>

<Form>

<Button> Update </Button>

<Button> Save </Button>

<Button> Delete </Button>

</Form>

</Article>

</Main>

# Each component have their own instance

```
<div>  
  <weather-widget value="jogjakarta" />  
  <weather-widget value="jakarta" />  
  <weather-widget value="bogor" />  
</div>
```

# Single Root Element

**// Good**

```
<template>
  <div>
    <weather-widget value="jogjakarta" />
    <weather-widget value="jakarta" />
  </div>
</template>
```

**// Bad**

```
<template>
  <weather-widget value="jogjakarta" />
  <weather-widget value="jakarta" />
</template>
```

# Naming Convention

1. Don't use same name with HTML tag
2. Two words

Read more at

- ❏ <https://vuejs.org/v2/style-guide>
- ❏ <https://vuejs.org/v2/guide/components-registration.html#Name-Casing>





SFC (Single File Component)

# SFC Structures

```
<template></template>
```

```
<script></script>
```

```
<style></style>
```

# Load external file

```
<template src="./views/index.html"></template>
```

```
<script src="./external/file.js"></script>
```

```
<style src="./external/style.css"></style>
```

# SFC with JSX - advanced

```
export default {  
  data() {  
    return {}  
  },  
  
  render() {  
    <div>Hello World!</div>  
  }  
}
```

[https://vuejs.org/v2/guide/\*\*single-file-components\*\*.html](https://vuejs.org/v2/guide/single-file-components.html)

# Component Registration

# Component Registration (local - per component)

```
<script>
  import WidgetChart = './components/WidgetChart.vue'

  export default {
    components: {
      WidgetChart
    }
  }
</script>

<template>
  <div>
    <WidgetChart />
    <widget-chart />
  </div>
</template>
```

# Component custom name

```
<script>
  import WidgetChart = './components/WidgetChart.vue'

  export default {
    components: {
      'custom-name': WidgetChart
    }
  }
</script>

<template>
  <div>
    <custom-name />
  </div>
</template>
```



# Component Global Registration

```
// main.js
Vue.component('WidgetWeather', () => {
  props: ['place']
})
```

```
<template>
  <widget-weather place="Jogja">
</template>
```

<https://vuejs.org/v2/guide/components-registration.html>



Passing props to child-component



# Passing props to child-component

```
// child-component
<template>
  <div>
    {{ value }}
  </div>
</template>

<script>
  export default {
    name: 'widget-weather',
    props: ['value']
  }
</script>
```

```
// parent
<template>
  <widget-weather
    value="Jakarta"
  />
</template>
<script>
  ...
  export default {
    components: {
      WidgetWeather,
    }
  }
</script>
```



Emit value to parent

# Emit value to parent

```
// child-component
<template>
  <button @click="sendToParent">
    Send to parent
  </button>
</template>
<script>
  export default {
    methods: {
      sendToParent() {
        this.$emit('update', '!!!')
      }
    }
  }
</script>
```

```
// parent
<template>
  <widget-weather
    @update="retrieved"
  />
</template>
<script>
  export default {
    methods: {
      retrieved(value) {
        console.log(value) // !!!
      }
    }
  }
</script>
```



You might need **Vuex** or another state management to **handle more complex states or nested components**

# Nested Components

<App>

<Topbar />

<Sidebar />

<Main>

<Settings>

<PersonalSettings>

Sidebar Toggle: on/off

</PersonalSettings>

</Settings>

</Main>

</App>



<https://vuejs.org/v2/guide/components.html>

Q & A



Thank you