

# REFERENSI CEPAT VUE.JS

Template, reaktivitas, komponen, Composition API, router

## Sintaks Template

### Teks dan Ekspresi

```
<span>{{ message }}</span>
<span>{{ count + 1 }}</span>
<span>{{ ok ? 'Yes' : 'No' }}</span>
<span v-html="rawHtml"></span>
```

### Direktif

```
{{ expr }}          Interpolasi teks
v-bind:attr / :attr  Ikat atribut ke ekspresi
v-on:event / @event Pasang event listener
v-model             Binding dua arah (form)
v-if / v-else-if / v-else Rending kondisional
v-show             Toggle CSS display (tetap di DOM)
v-for              Rending daftar
v-slot / #name     Konten slot bernama
```

### Binding Atribut

```

<div :class="{ active: isActive }"></div>
<div :style="{ color: textColor }"></div>
<button :disabled="isLoading">Submit</button>
```

## Reaktivitas

### ref (Primitif)

```
import { ref } from 'vue'

const count = ref(0)
console.log(count.value) // 0
count.value++           // pembaruan reaktif
```

### reactive (Objek)

```
import { reactive } from 'vue'

const state = reactive({ count: 0, name: 'Vue' })
state.count++ // tidak perlu .value
```

### ref vs reactive

```
ref()              Semua tipe; akses via ` .value ` di script
reactive()        Hanya objek/array; akses properti langsung
Template          Keduanya auto-unwrap (tidak perlu ` .value `)
Destructure      `reactive` kehilangan reaktivitas; gunakan `toRefs`
```

## Computed dan Watcher

### Properti Computed

```
import { ref, computed } from 'vue'

const items = ref([1, 2, 3, 4, 5])
const evenItems = computed(() =>
  items.value.filter(n => n % 2 === 0)
)
```

Nilai computed di-cache dan hanya dievaluasi ulang saat dependensi berubah

### Watcher

```
import { ref, watch, watchEffect } from 'vue'

const query = ref('')

// Pantau sumber tertentu
watch(query, (newVal, oldVal) => {
  console.log('Changed: ${oldVal} -> ${newVal}')
})

// Lacak dependensi secara otomatis
watchEffect(() => {
  console.log('Query is: ${query.value}')
})
```

### Opsi Watch

```
immediate: true  Jalankan callback segera saat dibuat
deep: true       Pantau mendalam objek bersarang
flush: 'post'    Jalankan setelah pembaruan DOM
once: true       Dipicu sekali lalu berhenti
```

## Komponen

### Single-File Component (SFC)

```
<script setup>
import { ref } from 'vue'
const count = ref(0)
</script>

<template>
<button @click="count++">{{ count }}</button>
</template>

<style scoped>
button { font-size: 1.2em; }
</style>
```

### Mendaftarkan Komponen

```
<!-- Auto-import dengan <script setup> -->
<script setup>
import MyButton from './MyButton.vue'
</script>

<template>
<MyButton label="Click me" />
</template>
```

### Blok SFC

```
<script setup> Composition API (disarankan)
<template>      Template HTML
<style scoped> CSS berskope ke komponen
<style module>  CSS Modules (objek $style)
```

## Props dan Event

### Mendefinisikan Props

```
<script setup>
const props = defineProps({
  title: String,
  count: { type: Number, default: 0 },
  items: { type: Array, required: true }
})
</script>
```

### Emit Event

```
<script setup>
const emit = defineEmits(['update', 'delete'])

function handleClick() {
  emit('update', { id: 1, value: 'new' })
}
</script>
```

## Penggunaan di Komponen Induk

```
<ChildComponent
  :title="pageTitle"
  :count="total"
  @update="handleUpdate"
  @delete="handleDelete"
/>
```

### v-model pada Komponen

```
<!-- Induk -->
<CustomInput v-model="search" />

<!-- CustomInput.vue -->
<script setup>
const model = defineModel()
</script>
<template>
<input :value="model" @input="model = $event.target.value" />
</template>
```

## Slot

### Slot Default

```
<!-- Card.vue -->
<template>
<div class="card">
  <slot>Konten fallback</slot>
</div>
</template>
```

```
<!-- Penggunaan -->
<Card><p>Konten kustom di sini</p></Card>
```

### Slot Bernama

```
<!-- Layout.vue -->
<template>
<header><slot name="header" /></header>
<main><slot /></main>
<footer><slot name="footer" /></footer>
</template>
```

```
<!-- Penggunaan -->
<Layout>
<template #header><h1>Judul</h1></template>
<p>Konten utama</p>
<template #footer><span>Footer</span></template>
</Layout>
```

### Scoped Slot

```
<!-- List.vue -->
<ul>
<li v-for="item in items" :key="item.id">
  <slot :item="item" />
</li>
</ul>
```

```
<!-- Penggunaan -->
<List :items="todogs">
<template #default="{ item }">
  <span>{{ item.text }}</span>
</template>
</List>
```

## Composition API

### Fungsi Composable

```
// useMouse.js
import { ref, onMounted, onUnmounted } from 'vue'

export function useMouse() {
  const x = ref(0)
  const y = ref(0)
  function update(e) {
    x.value = e.pageX
    y.value = e.pageY
  }
  onMounted(() => window.addEventListener('mousemove', update))
  onUnmounted(() => window.removeEventListener('mousemove', update))
  return { x, y }
}
```

### Menggunakan Composable

```
<script setup>
import { useMouse } from './useMouse'

const { x, y } = useMouse()
</script>
<template>
<p>Mouse: {{ x }}, {{ y }}</p>
</template>
```

### provide / inject

```
// Induk
import { provide, ref } from 'vue'
const theme = ref('dark')
provide('theme', theme)

// Keturunan (kedalaman berapa pun)
import { inject } from 'vue'
const theme = inject('theme', 'light') // default
```

## Router (Vue Router)

### Definisi Route

```
import { createRouter, createWebHistory } from 'vue-router'

const router = createRouter({
  history: createWebHistory(),
  routes: [
    { path: '/', component: Home },
    { path: '/about', component: About },
    { path: '/user/:id', component: User },
  ]
})
```

### Navigasi di Template

```
<router-link to="/">Home</router-link>
<router-link :to="{ name: 'user', params: { id: 1 } }">
  User 1
</router-link>
<router-view />
```

### Navigasi Programatik

```
import { useRouter, useRoute } from 'vue-router'

const router = useRouter()
const route = useRoute()

router.push('/about')
router.push({ name: 'user', params: { id: 1 } })
console.log(route.params.id)
```

### Fitur Route

```
/user/:id          Segmen dinamis ( `route.params.id` )
name: 'user'       Route bernama untuk navigasi
                  programatik
children: [...]    Route bersarang
beforeEnter        Navigation guard per route
meta: { auth: true } Metadata kustom untuk guard
redirect: '/new-path' Redirect route
```

## Lifecycle Hook

### Urutan Hook

```
onBeforeMount      Sebelum render DOM awal
onMounted           DOM siap (ambil data, tambah listener)
onBeforeUpdate      Sebelum state reaktif me-render ulang DOM
onUpdated           Setelah re-render DOM
onBeforeUnmount     Sebelum komponen dihancurkan
onUnmounted        Pembersihan (hapus listener, timer)
```

### Penggunaan

```
<script setup>
import { onMounted, onUnmounted } from 'vue'

onMounted(() => {
  console.log('Komponen ter-mount')
})

onUnmounted(() => {
  console.log('Pembersihan di sini')
})
</script>
```

## Daftar dan Kondisional

### v-for

```
<li v-for="item in items" :key="item.id">
  {{ item.name }}
</li>
<li v-for="(item, index) in items" :key="item.id">
  {{ index }}: {{ item.name }}
</li>
<div v-for="(val, key) in obj" :key="key">
  {{ key }}: {{ val }}
</div>
```

Selalu gunakan `key` dengan v-for untuk pembaruan DOM yang efisien

### v-if vs v-show

```
<v-if>            Render kondisional (tambah/hapus dari DOM)
(v-else-if)       Rantai else-if
v-else            Cabang fallback
v-show            Toggle `display: none` (tetap di DOM)
```

Gunakan v-show untuk toggle sering, v-if untuk perubahan jarang

### Contoh Kondisional

```
<div v-if="status === 'loading'">Memuat...</div>
<div v-else-if="status === 'error'">Error!</div>
<div v-else>{{ data }}</div>
```

## Penganganan Form

### Dasar v-model

```
<input v-model="text" />
<textarea v-model="message"></textarea>
<input type="checkbox" v-model="checked" />
<select v-model="selected">
  <option value="a">A</option>
  <option value="b">B</option>
</select>
```

### Modifier v-model

```
v-model.lazy      Sinkron pada `change` bukan `input`
v-model.number    Auto-typecast ke angka
v-model.trim      Auto-trim spasi
```

### Modifier Event

```
@click.prevent    Panggil `preventDefault()`
@click.stop        Panggil `stopPropagation()`
@click.once        Dipicu paling banyak sekali
@keyup.enter       Hanya pada tombol Enter
@submit.prevent    Cegah pengiriman form default
```