

# REFERENSI CEPAT SASS/SCSS

Variabel, nesting, mixin, fungsi, alur kontrol

## Sintaks

### SCSS vs Sass

```
// SCSS (superset CSS – menggunakan kurung kurawal)
.nav { display: flex; }

// Sass (berdasarkan indentasi – tanpa kurung atau titik koma)
.nav
  display: flex
```

SCSS adalah sintaks yang paling umum digunakan

### Perbandingan

**SCSS (.scss)** Kompatibel dengan CSS, kurung kurawal & titik koma  
**Sass (.sass)** Berbasis indentasi, tanpa kurung kurawal  
**Output** Keduanya dikompilasi menjadi CSS standar  
**Disarankan** SCSS (adopsi lebih luas, migrasi lebih mudah)

## Variabel

### Mendefinisikan & Menggunakan

```
$primary: #3498db;
$spacing: 16px;
$font-stack: "Helvetica", Arial, sans-serif;

.btn {
  color: $primary;
  padding: $spacing;
  font-family: $font-stack;
}
```

### Scope Variabel

```
$color: red; // global
.card {
  $color: blue; // lokal untuk .card
  color: $color; // blue
}
.other { color: $color; } // red
```

### Flag

**!default** Set hanya jika belum didefinisikan  
**!global** Promosikan variabel lokal ke scope global

## Nesting

### Nesting Selector

```
.nav {
  ul { list-style: none; }
  li { display: inline-block; }
  a { text-decoration: none; }
  &:hover { color: blue; } // & = selector parent
}
```

### Selector Parent (&)

```
.btn {
  &-primary { background: blue; } // BEM: .btn-primary
  &-icon { margin-right: 4px; } // BEM: .btn_icon
  &-dark { color: white; } // .dark .btn
}
```

### Nesting Properti

```
.box {
  border: { width: 1px; style: solid; color: #ccc; }
  // dikompilasi menjadi: border-width, border-style, border-color
}
```

## Mixin

### Definisikan & Include

```
@mixin flex-center($direction: row) {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: $direction;
}

.hero { @include flex-center(column); }
```

### Content Block

```
@mixin responsive($breakpoint) {
  @media (min-width: $breakpoint) { @content; }
}

.sidebar {
  width: 100%;
  @include responsive(768px) { width: 300px; }
}
```

### Fitur Mixin

**@mixin name(\$args)** Definisikan blok style yang dapat digunakan ulang  
**@include name()** Gunakan mixin  
**Default params** 'Sarg: value' untuk parameter opsional  
**\$args...** Argumen variabel (rest params)  
**@content** Sisipkan blok konten dari pemanggil

## Fungsi

### Fungsi Kustom

```
@function rem($px, $base: 16) {
  @return math.div($px, $base) * 1rem;
}

.title { font-size: rem(24); } // 1.5rem
```

### Fungsi Bawaan

**darken(\$color, 10%)** Gelapkan warna  
**lighten(\$color, 10%)** Cerahkan warna  
**mix(\$c1, \$c2, 50%)** Campurkan dua warna  
**rgba(\$color, 0.5)** Set saluran alpha  
**math.div(\$a, \$b)** Pembagian (menggantikan '/')  
**math.round(\$n)** Bulatkan angka  
**string.quote(\$s)** Tambahkan tanda kutip pada string  
**if(\$cond, \$t, \$f)** Kondisional inline

## Extend

### Extend & Placeholder

```
%flex-center { // placeholder – tidak dipancarkan
  display: flex;
  justify-content: center;
  align-items: center;
}

.hero { @extend %flex-center; }
.modal { @extend %flex-center; }
// Keduanya berbagi satu aturan CSS via pengelompokan selector
```

## Extend vs Mixin

**@extend** Kelompokkan selector — output CSS lebih kecil  
**@mixin** Salin deklarasi — mendukung argumen  
**% placeholder** Hanya untuk extend (tidak dipancarkan jika tidak digunakan)  
**Rekomendasi** Lebih baik gunakan mixin untuk style berparameter

## Partial & Import

### Organisasi File

```
// variables.scss (partial – tidak dikompilasi sendiri)
$primary: #3498db;

// main.scss
@use "variables"; // modern: dengan namespace
.btn { color: variables.$primary; }

@use "variables" as v; // alias
.btn { color: v.$primary; }
```

### Sistem Modul

**@use 'file'** Muat modul dengan namespace  
**@use 'file' as \*** Muat tanpa namespace  
**@use 'file' as alias** Namespace kustom  
**@forward 'file'** Re-export anggota modul  
**\_partial.scss** File yang tidak dikompilasi secara mandiri  
**@import** sudah deprecated — gunakan @use dan @forward

## Alur Kontrol

### Kondisional

```
@mixin theme($mode) {
  @if $mode == dark {
    background: #333; color: #fff;
  } @else {
    background: #fff; color: #333;
  }
}
```

### Loop

```
@for $i from 1 through 4 {
  .col-#{$i} { width: 25% * $i; }
}

@each $name, $color in (primary: blue, danger: red) {
  .text-#{$name} { color: $color; }
}
```

### Direktif

**@if / @else if / @else** Logika kondisional  
**@for \$i from a through b** Loop numerik (inklusif)  
**@for \$i from a to b** Loop numerik (akhir eksklusif)  
**@each \$item in \$list** Iterasi list atau map  
**@while** Loop selama kondisi true  
**#{var:}** Interpolasi dalam selector/properti

## Map & List

### Map

```
$colors: (primary: #3498db, danger: #e74c3c, success: #2ecc71);

.alert { color: map.get($colors, danger); }

@each $name, $color in $colors {
  .bg-#{$name} { background: $color; }
}
```

### List

```
$sizes: 8px 16px 24px 32px;
.box { padding: list.nth($sizes, 2); } // 16px
```

### Fungsi Map & List

**map.get(\$map, \$key)** Ambil nilai berdasarkan key  
**map.merge(\$m1, \$m2)** Gabungkan dua map  
**map.keys(\$map)** Daftar semua key  
**map.has-key(\$map, \$key)** Periksa apakah key ada  
**list.nth(\$list, \$n)** Ambil item pada indeks (berbasis 1)  
**list.length(\$list)** Jumlah item  
**list.append(\$list, \$val)** Tambahkan item ke list

## Pola Umum

### Breakpoint Responsif

```
$breakpoints: (sm: 576px, md: 768px, lg: 992px, xl: 1200px);

@mixin bp($name) {
  @media (min-width: map.get($breakpoints, $name)) {
    @content;
  }
}

.sidebar { width: 100%; @include bp(md) { width: 300px; } }
```

### Generator Utility

```
$spaces: (0: 0, 1: 4px, 2: 8px, 3: 16px, 4: 32px);

@each $key, $val in $spaces {
  .mt-#{$key} { margin-top: $val; }
  .mb-#{$key} { margin-bottom: $val; }
  .p-#{$key} { padding: $val; }
}
```

### Mode Gelap

```
@mixin dark { @media (prefers-color-scheme: dark) { @content; } }

body {
  background: #fff;
  @include dark { background: #1a1a1a; color: #eee; }
}
```