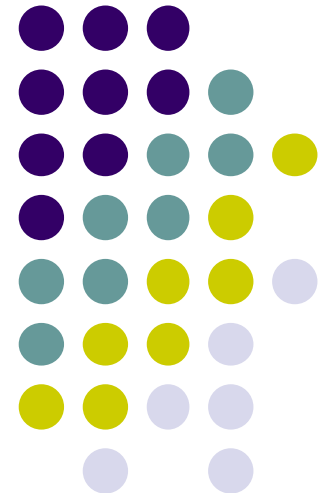
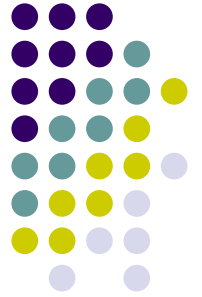


# JavaScript

---

DOM (Document Object Model)

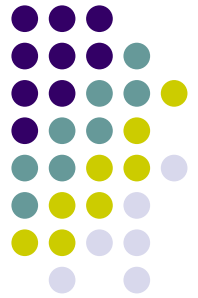




# Materi

- Object pada Javascript
- Penjelasan DOM
  - DOM Tree
  - DOM Node
- Penelusuran DOM
- Manipulasi DOM

# Object

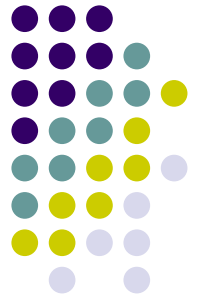


- Variabel khusus yang dapat menampung banyak data dan fungsi
- Variabel pada object disebut **property**
- Fungsi pada object disebut **method**
- Data dalam object tidak harus memiliki tipe yang sama
- Digunakan untuk menjelaskan / memberi ciri pada sesuatu

# Object 'Mahasiswa' (1)



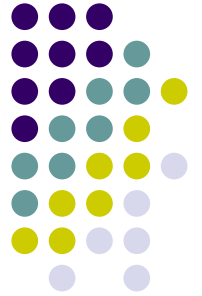
- Mahasiswa
  - NRP
  - Nama Depan
  - Nama Belakang
  - Umur
  - Nama Lengkap (*gabungan dari Nama Depan & Nama Belakang*)
  - dll.



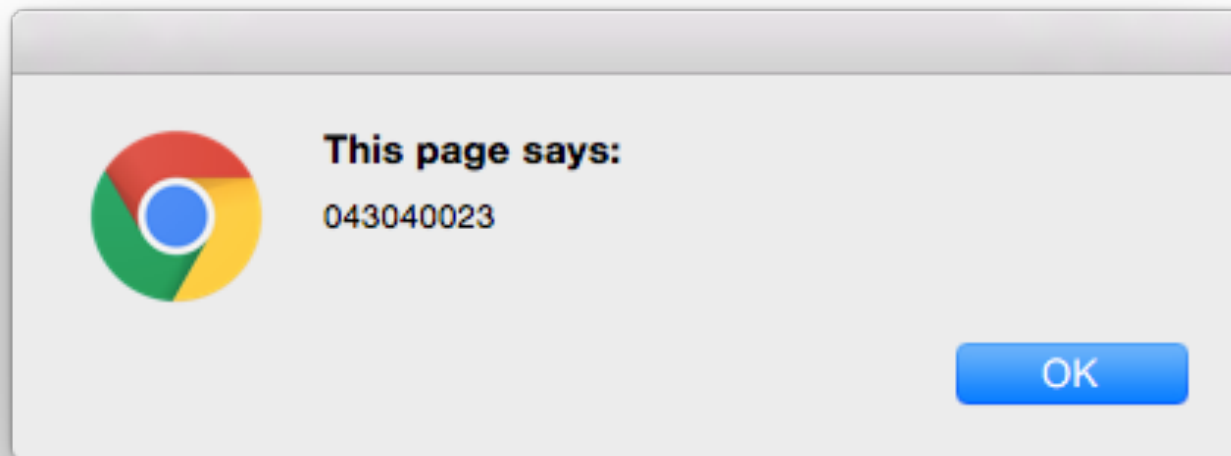
## Object 'Mahasiswa' (2)

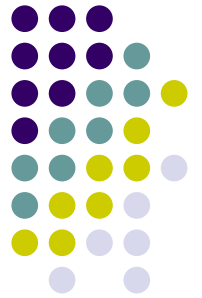
```
var mahasiswa = {  
  nrp : '043040023',  
  namaDepan : 'Sandhika',  
  namaBelakang : 'Galih',  
  umur : 30,  
  namaLengkap : function() {  
    return this.namaDepan + ' ' + this.namaBelakang;  
  }  
}
```

# Mengakses Object (1)



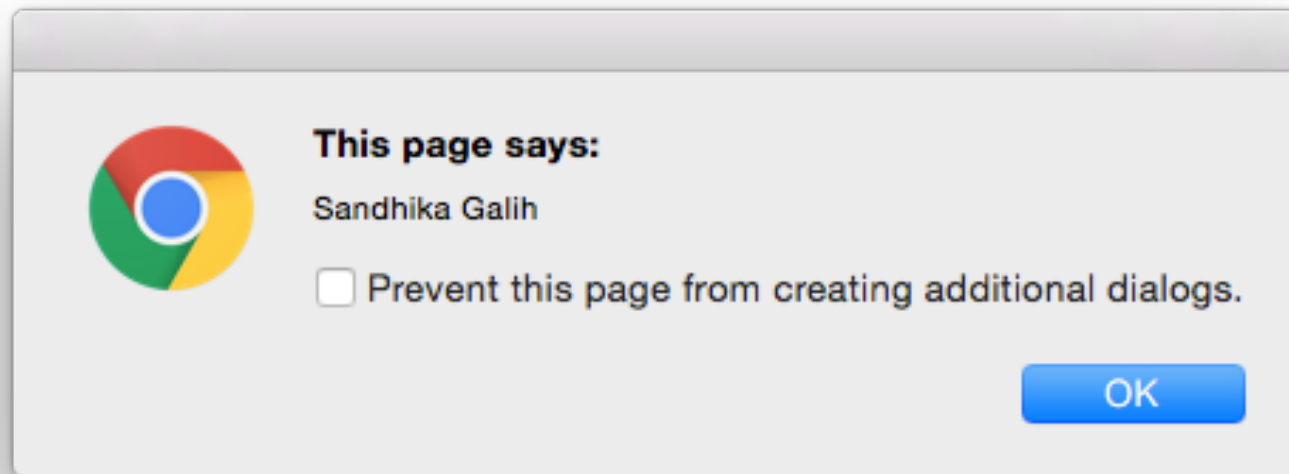
```
alert(mahasiswa.nrp);
```

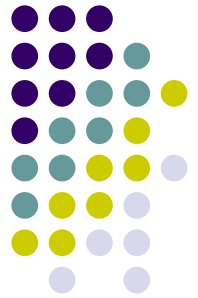




# Mengakses Object (2)

```
alert(mahasiswa.namaLengkap());
```

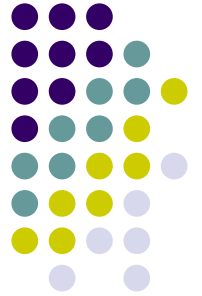




# DOM (Document Object Model)

- Sebuah API (Application Programming Interface) berbasis Objek untuk menjelaskan dokumen HTML
- DOM membuat seluruh komponen dari halaman web dapat diakses
- Komponen:
  - Elemen HTML
  - Atribut
  - Text
- DOM dapat manipulasi (dibuat, diubah dan dihapus) menggunakan javascript

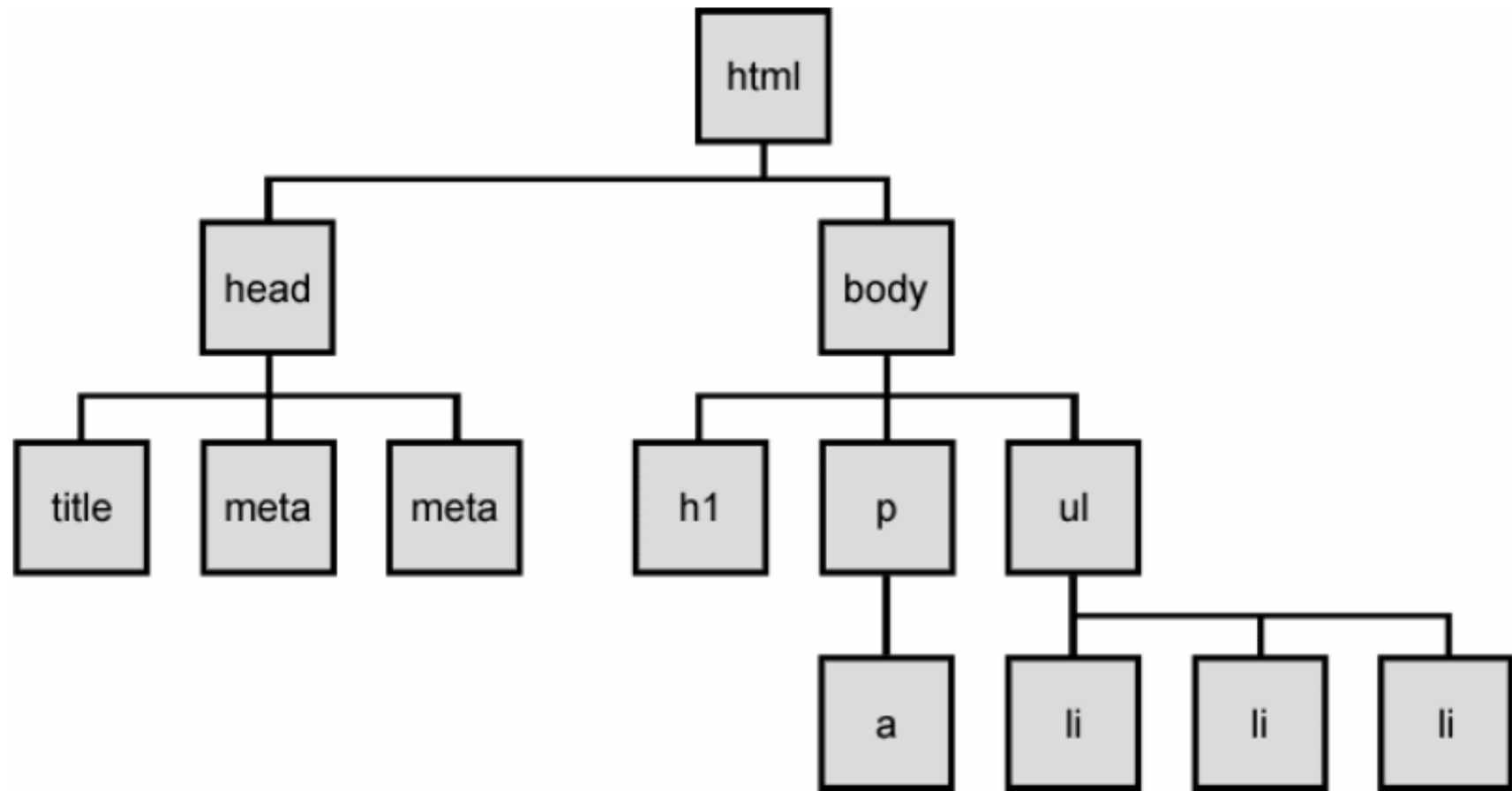
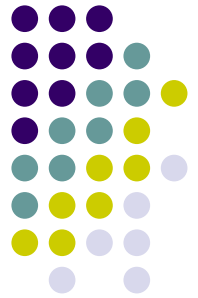


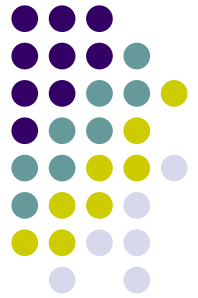


# Objek DOM

- Komponen DOM dapat diakses sebagai objek atau kumpulan dari objek
- Komponen-komponen DOM membentuk tree / *pohon* yang terdiri dari node-node / *simpul-simpul*
- Tiap-tiap simpul membentuk hirarki dan memiliki keterhubungan (*parent – child*)

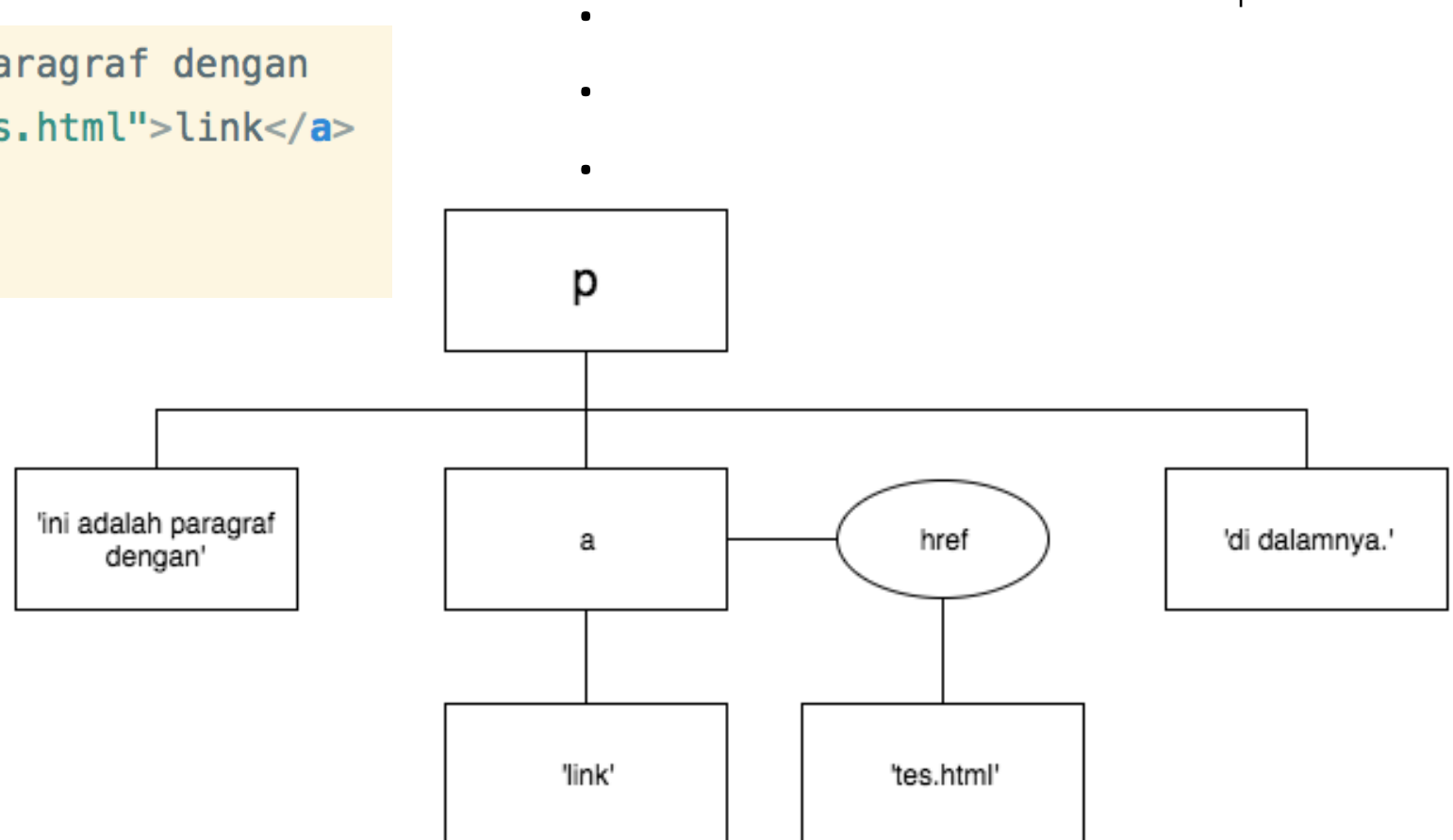
# DOM Tree (contoh)



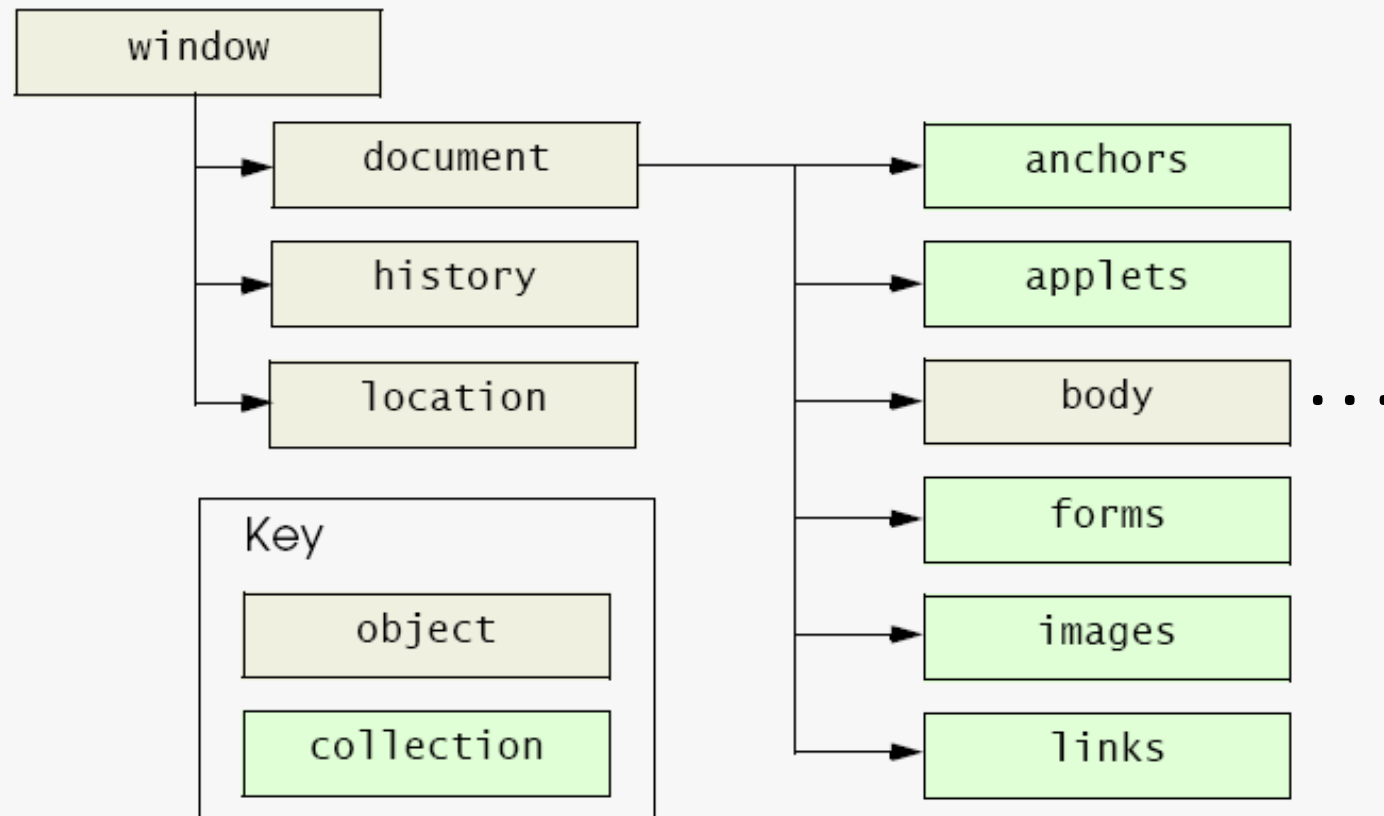


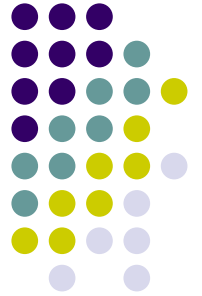
# DOM Tree (contoh 2)

```
<p>ini adalah paragraf dengan  
  <a href="tes.html">link</a>  
  di dalamnya.  
</p>
```



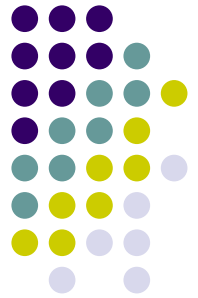
# W3C DOM





# Object DOM Khusus

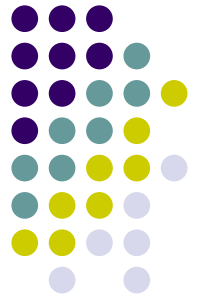
- **window**
  - browser window
- **document**
  - halaman web yang sedang diakses (berada di dalam **window**)
- **body**
  - elemen `<body>` di dalam **document**
- **history**
  - halaman yang pernah dikunjungi user
- **location**
  - URL dari **document**
  - jika mengubah ini maka akan memindahkan halaman



# Mengakses node pada DOM

- `document.getElementsByTagName( 'namatag' )`
  - mengembalikan **collection** elemen dengan nama tag tertentu
- `document.getElementById( 'id' )`
  - mengembalikan 1 elemen dengan id tertentu
- `document.getElementsByClassName( 'class' )`
  - mengembalikan **collection** elemen dengan nama kelas tertentu
- `document.querySelector( 'selector' )`
  - mengembalikan 1 elemen yang sesuai dengan selector tertentu
- `document.querySelectorAll( 'selector' )`
  - mengembalikan **collection** elemen yang sesuai dengan selector tertentu

# Menelusuri DOM berdasarkan node tertentu



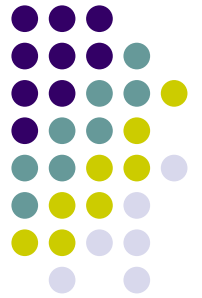
- **childNodes, children**
  - mengembalikan **collection** elemen child dari sebuah node
- **firstChild, lastChild**
  - mengembalikan 1 elemen child awal atau akhir dari sebuah node
- **nextSibling, previousSibling**
  - mengembalikan 1 elemen sejajar sebelum atau sesudah dari sebuah node
- **parentNode**
  - mengembalikan 1 elemen parent dari sebuah node

# Properti dari Node

- `attributes`
  - `getAttribute`
  - `setAttribute`
- `innerHTML`
- `style`
- `className`

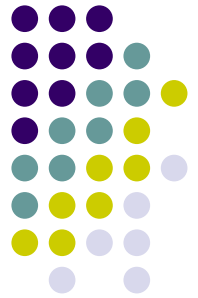






# Manipulasi DOM (1)

- **document.createElement(<tag>)**
  - Membuat elemen node / simpul baru pada DOM berupa tag.
  - Perintah diatas hanya membuat node / simpul baru saja
  - Node / simpul baru tersebut tetap harus disisipkan pada DOM menggunakan perintah lain
- **document.createTextNode(<text>)**
  - membuat elemen text <text>



# Manipulasi DOM (2)

- `<parent>.appendChild(<child>)`
  - Menyisipkan node `<child>` di **akhir** seluruh child dari sebuah `<parent>` node tertentu
- `<parent>.insertBefore(<child>, <before>)`
  - Menyisipkan node `<child>` di **awal** seluruh child dari sebuah `<parent>` node tertentu
- `<parent>.replaceChild(<childBaru>, <childLama>)`
  - Mengganti `<childLama>` dengan `<childBaru>` di dalam sebuah `<parent>` node tertentu
- `<parent>.removeChild(<child>)`
  - Menghilangkan `<child>` node dari sebuah `<parent>` node tertentu

# TERIMA KASIH

---

