

# Hypertext Markup Language (HTML)

## Pertemuan - 02

Matakuliah Pemrograman Web (TIW213)  
Semester Genap 2015/2016

Yuan Lukito dan Antonius Rachmat C



Prodi Teknik Informatika  
Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana



# Topik Bahasan

- Perkembangan HTML
  - HTML4.x, HTML5, XHTML, ....
  - Hubungan HTML, CSS dan Javascript
- Struktur Dokumen HTML
  - Bagian-bagian penyusun dokumen HTML
- Document Object Model (DOM)
- Element pada HTML
  - Inline Element dan Block Element
- Beberapa Element pada HTML
  - Header, Paragraph, Link, Div, Span, Table, Form

# WWW Inventor: Sir Tim Berners Lee

1980 : ESQUIRE

A system for CERN researchers to  
**use and share documents**

1989 : Memo berjudul  
Information Management: A  
Proposal

Internet based hypertext system

1990 : HTML Specification,  
Browser and Server Software

1991-... : Spesifikasi HTML dijadikan standar dan  
terus berkembang



# HTML: Hypertext Markup Language

- **HTML Document**

- Document = File yang memiliki struktur tertentu, pada umumnya berisi data dalam bentuk teks (string/text)



- Hypertext Markup Language

# HTML

- Design goals:
  - **Platform independence:** pages can be viewed using a variety of different computers and browsers.
  - **Universality:** can be accessed by all browsers
  - **Convenient** linking from one page to another (hypertext).
  - HTML conveys the **structure** of the document, *not its precise appearance*.

# HTML

- License free (US\$ 0) to use
- Easy to create, edit and read using any text editor
- Standardized (W3C)
  - <http://www.w3.org/standards/webdesign/htmlcss.html>
  - Semua Web Browser dan Web Server diharapkan mengikuti standar yang sudah ditentukan oleh W3C
- Works across many systems
  - Device: Desktop, Tablet, Smartphone, Embedded, ...
  - Operating System: Windows, Mac, Linux, ...

# Perkembangan HTML

- **HTML Tags** - 1990-1992  
(<http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>)
- **HTML 2.0** RFC 1866  
(<http://www.ietf.org/rfc/rfc1866.txt>) - 1994-1995  
drafts published as Proposed Standard November 24, 1995 (<http://www.w3.org/MarkUp/html-spec/>), made obsolete by RFC 2854 in June 2000 (<http://tools.ietf.org/html/rfc2854>)
- ~~**HTML 3.0**~~ - March 1995 draft **never advanced** to a Recommendation  
(<http://www.w3.org/MarkUp/html3/>)
- The IETF HTML Working Group was closed September 12, 1996 and work on HTML specifications was moved to the **World Wide Web Consortium (W3C)**

# Perkembangan HTML

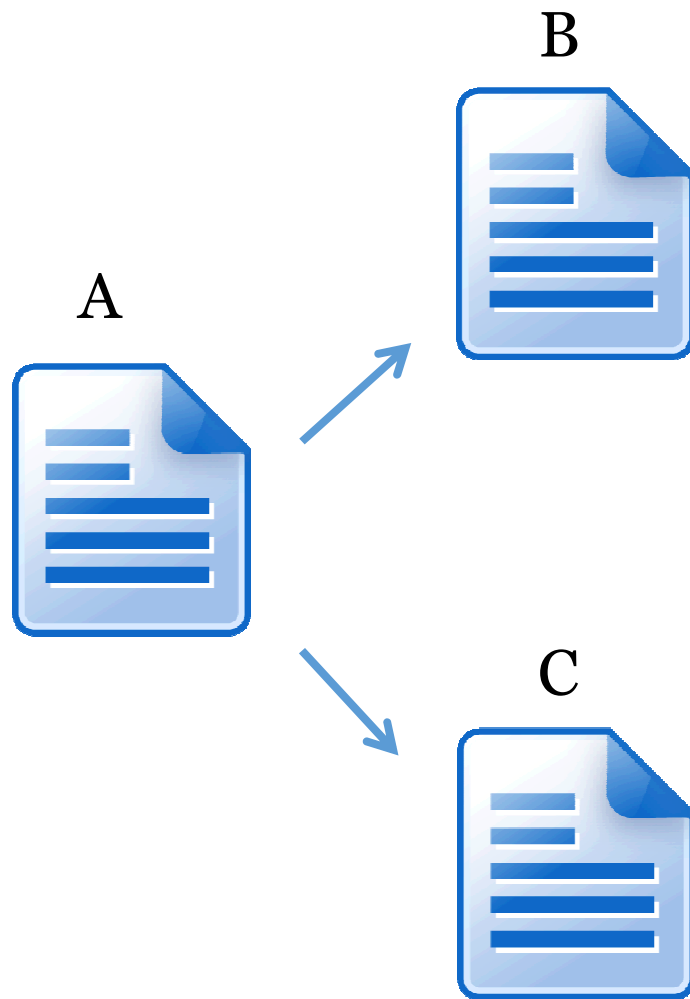
- **HTML 3.2** Reference Specification - January 14, 1997  
(<http://www.w3.org/TR/REC-html32.html>)
- **HTML 4.0** W3C Recommendation December 18, 1997  
(<http://www.w3.org/TR/REC-html40-971218/>)  
revised April 24, 1998  
(<http://www.w3.org/TR/1998/REC-html40-19980424/>)
- **HTML 4.01** W3C Recommendation December 24, 1999  
(<http://www.w3.org/TR/1999/REC-html401-19991224/>)
- **XHTML 1.0** W3C Recommendation - January 26, 2000  
(<http://www.w3.org/TR/2002/REC-xhtml1-20020801/>)
- **XHTML 1.1** W3C Recommendation for modular HTML  
- May 31, 2001 (<http://www.w3.org/TR/2001/REC-xhtml11-20010531/>)



# Perkembangan HTML

- **XHTML 1.0 Second Edition** W3C Recommendation - August 1, 2002 (<http://www.w3.org/TR/2002/REC-xhtml1-20020801/>)
- **XHTML 1.1 Second Edition** W3C Recommendation for modular HTML - November 23, 2010 (<http://www.w3.org/TR/2010/REC-xhtml11-20101123/>)
- **WHATWG HTML 5** - October 27, 2007 to December, 2010
- **WHATWG HTML Living Standard** - January 19, 2011 and beyond, ongoing
- **HTML 5 W3C Recommendation** - last call expected in 2011

# Hypertext Markup Language



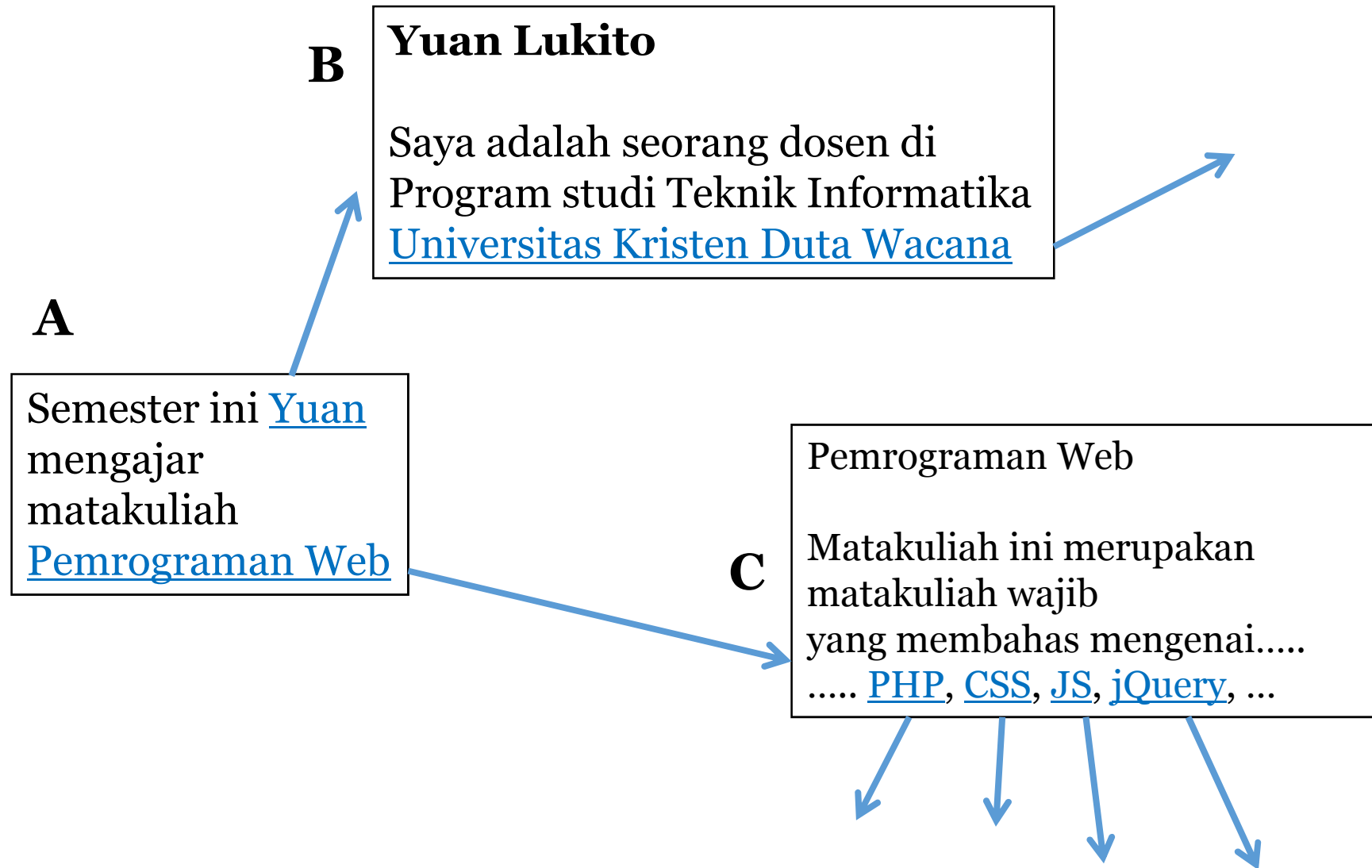
Bagaimana suatu document bisa “terhubung” dengan document lainnya?

Misal suatu document A berisi sebuah informasi yang dikutip dari document B dan C.

Bagaimana caranya supaya dari A, bisa **ditelusuri langsung** ke B atau C?

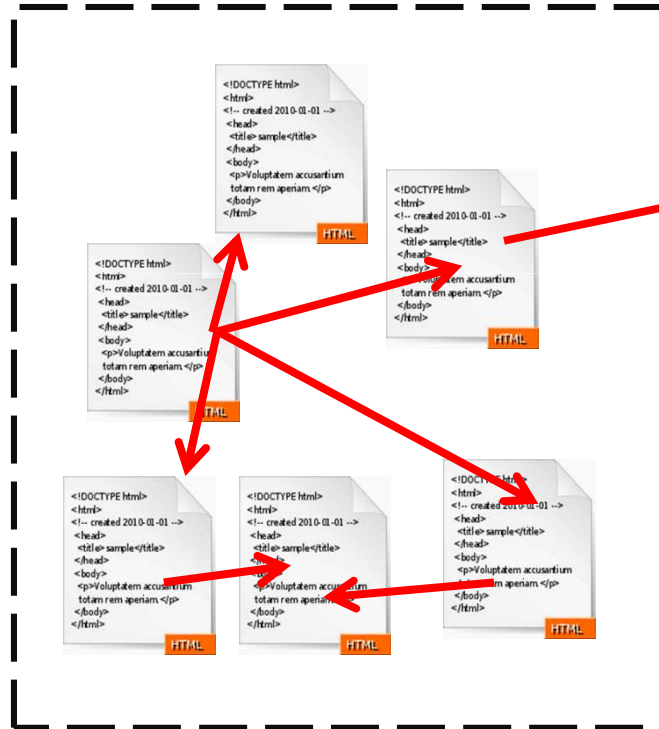
Document B dan C letaknya bisa saja di **jaringan / komputer yang berbeda, di belahan dunia yang berbeda**

# Hypertext Markup Language

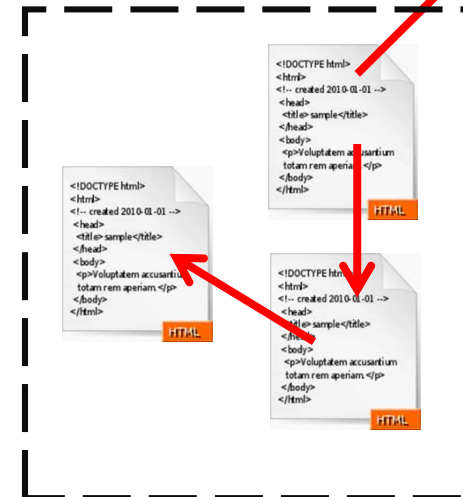
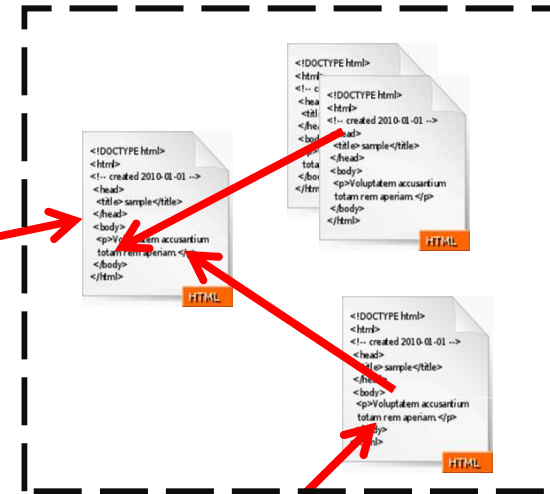


# Hypertext Markup Language

**imdb.com**

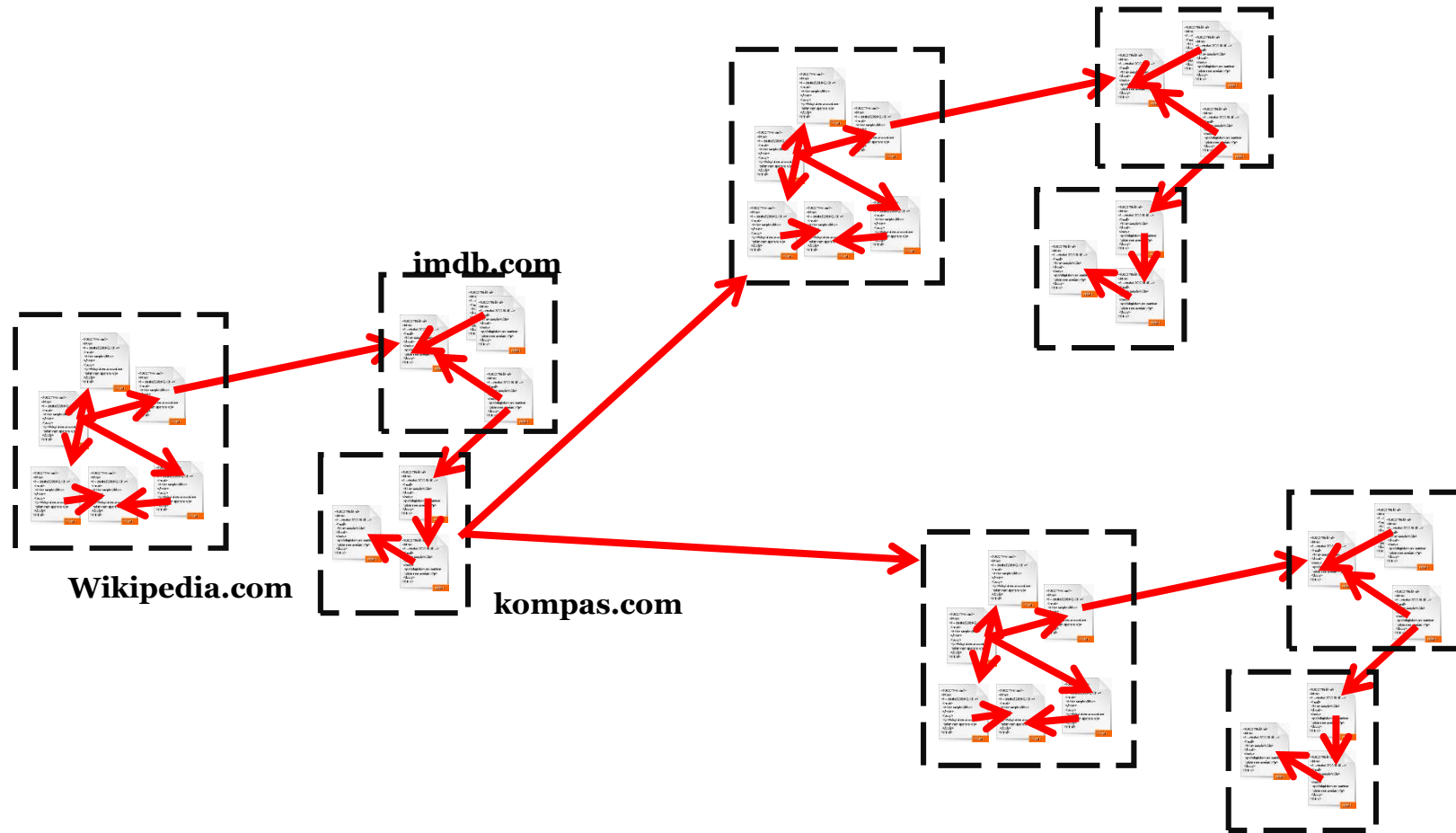


**Wikipedia.com**



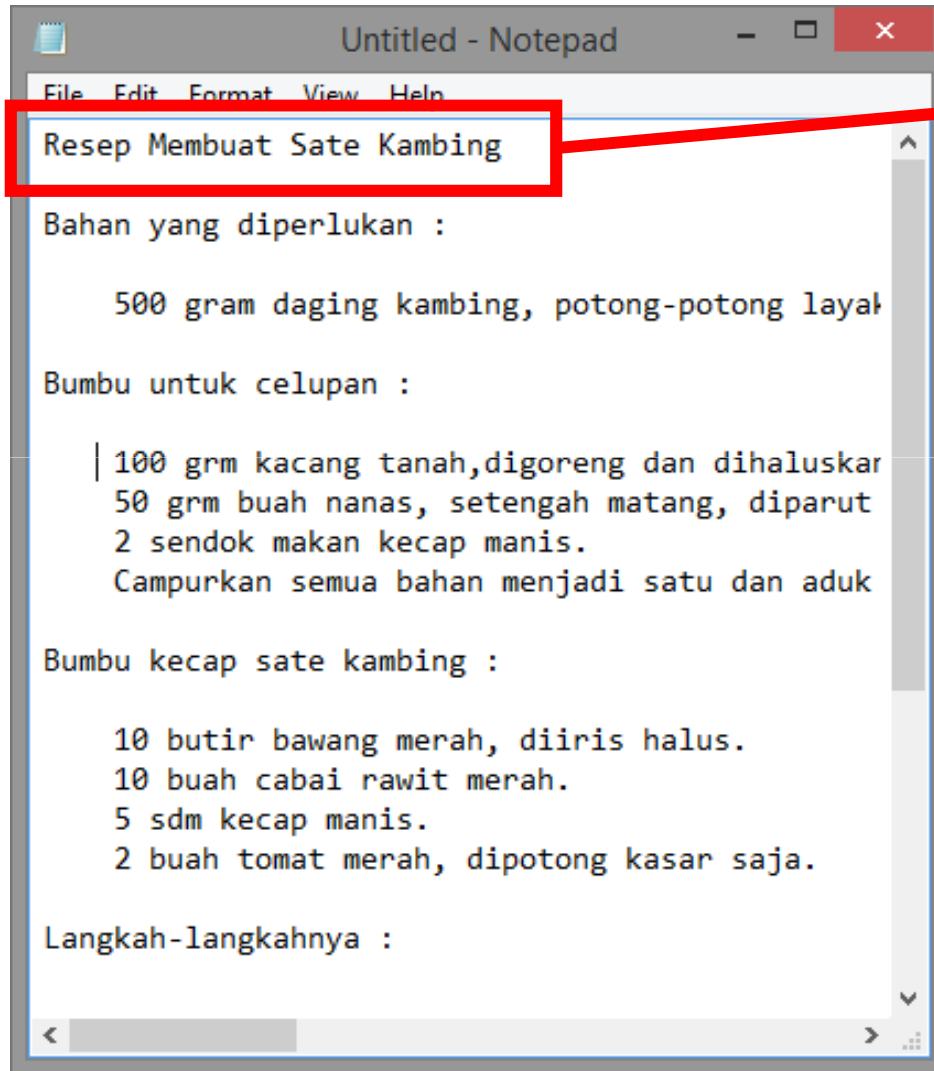
**kompas.com**

# Hypertext Markup Language



**Hypertext:** text which contains links to other text

# Hypertext Markup Language

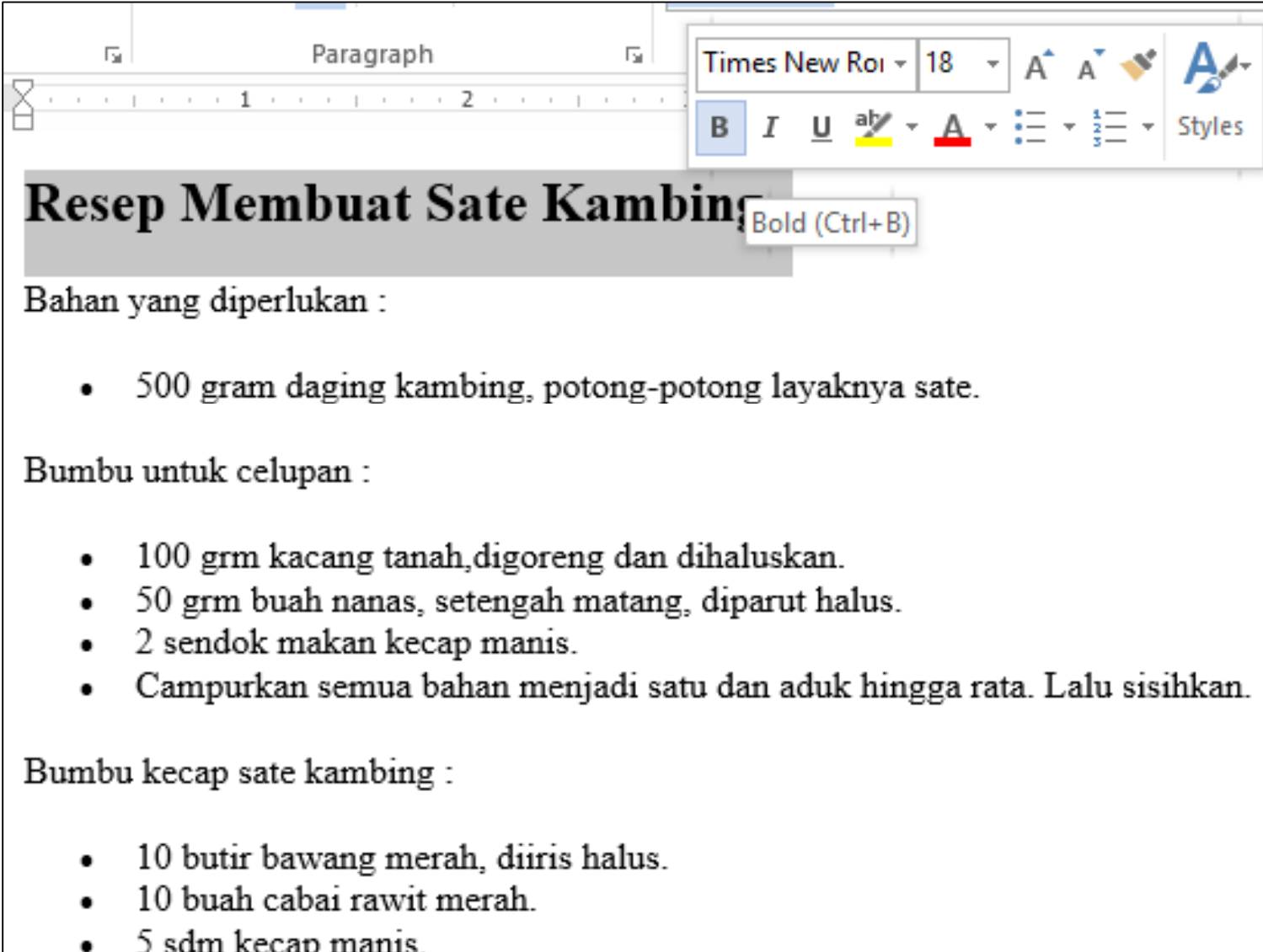


Judul/nama resep  
(tidak ada perbedaan  
dengan baris lainnya)

Text file

Flat file (tidak ada  
strukturnya),  
hanya berisi plain  
text

# Hypertext Markup Language



The image shows a screenshot of a Microsoft Word document. At the top, the title bar reads 'Paragraph'. Below it, a ruler shows the first two inches. The main text area has a title 'Resep Membuat Sate Kambing' in bold black font. A floating toolbar is positioned over the title, showing options for font face (Times New Roman), size (18), bold (B), italic (I), underline (U), text color (A), background color (shaded), bullet points, numbered list, and styles. Below the title, the text 'Bahan yang diperlukan :' is followed by a bulleted list: '500 gram daging kambing, potong-potong layaknya sate.' Then, 'Bumbu untuk celupan :' is followed by another bulleted list: '100 grm kacang tanah, digoreng dan dihaluskan.', '50 grm buah nanas, setengah matang, diparut halus.', '2 sendok makan kecap manis.', and 'Campurkan semua bahan menjadi satu dan aduk hingga rata. Lalu sisihkan.' Finally, 'Bumbu kecap sate kambing :' is followed by a third bulleted list: '10 butir bawang merah, diiris halus.', '10 buah cabai rawit merah.', and '5 sdm kecap manis.'

**Resep Membuat Sate Kambing**

Bahan yang diperlukan :

- 500 gram daging kambing, potong-potong layaknya sate.

Bumbu untuk celupan :

- 100 grm kacang tanah, digoreng dan dihaluskan.
- 50 grm buah nanas, setengah matang, diparut halus.
- 2 sendok makan kecap manis.
- Campurkan semua bahan menjadi satu dan aduk hingga rata. Lalu sisihkan.

Bumbu kecap sate kambing :

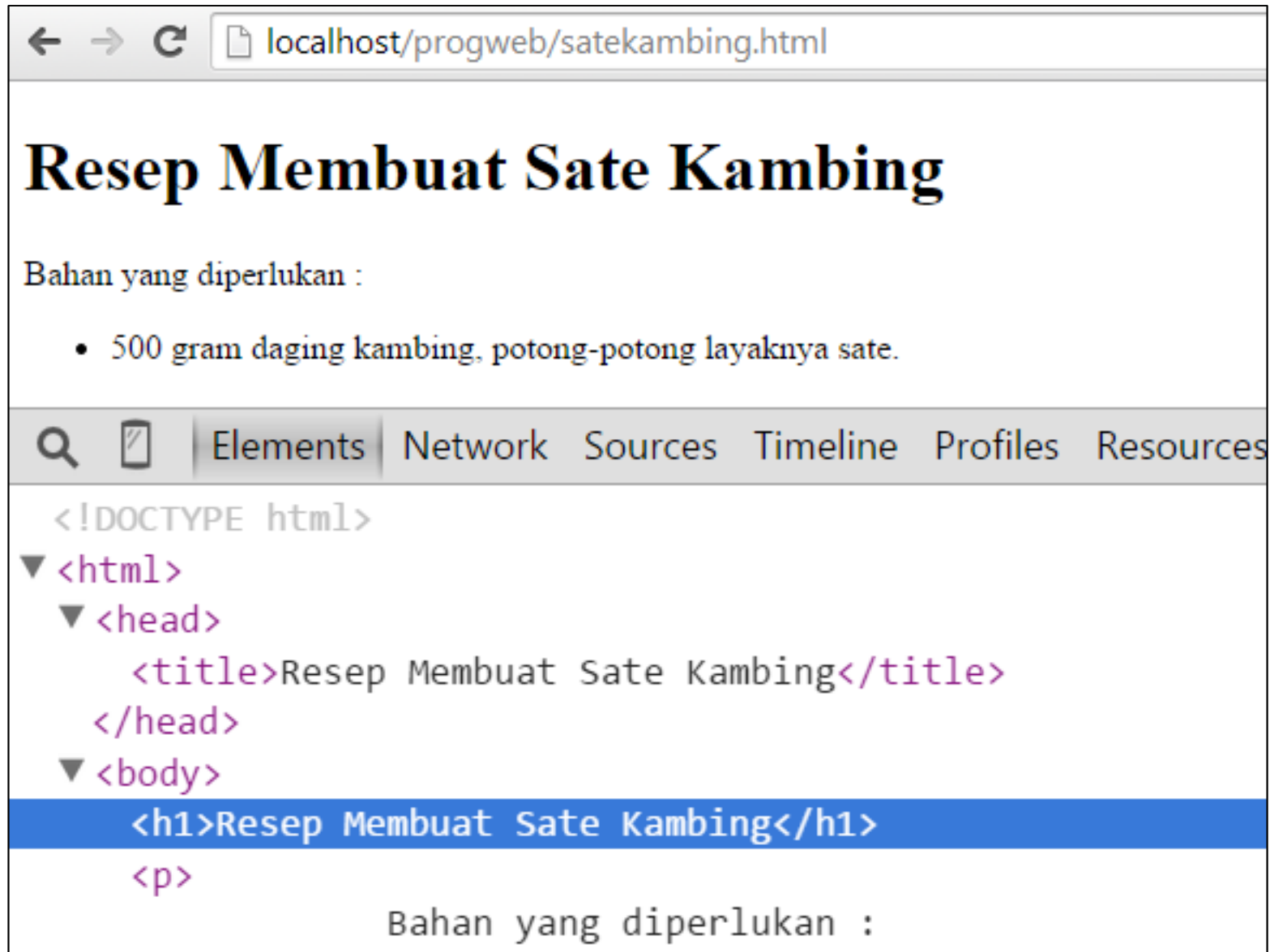
- 10 butir bawang merah, diiris halus.
- 10 buah cabai rawit merah.
- 5 sdm kecap manis.

# Hypertext Markup Language

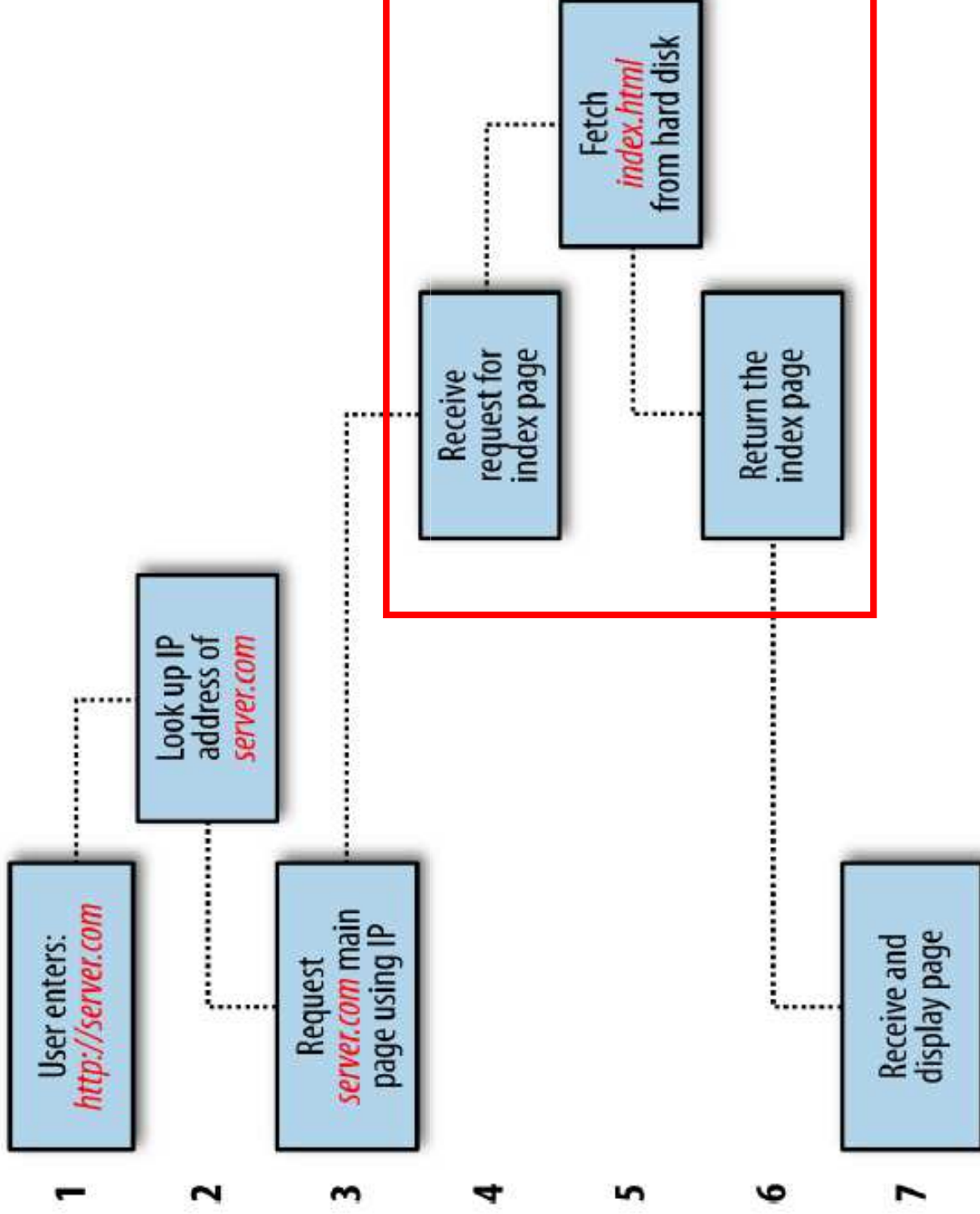
- Markup Language
  - A modern system for annotating document in a way that is syntactically distinguishable from the text
    - WordStar, RTF, HTML, XML, dsb...
  - Annotation: informasi tambahan berisi makna, relasi atau keterangan dari suatu bagian text
- Pada HTML, markup diimplementasikan dalam bentuk tag
  - Contoh: **<h1>Resep Membuat Sate Kambing </h1>**
- Tag terdiri dari opening, content dan closing
  - Opening: **<h1>**
  - Content: **Resep Membuat Sate Kambing**
  - Closing: **</h1>**



# Hypertext Markup Language

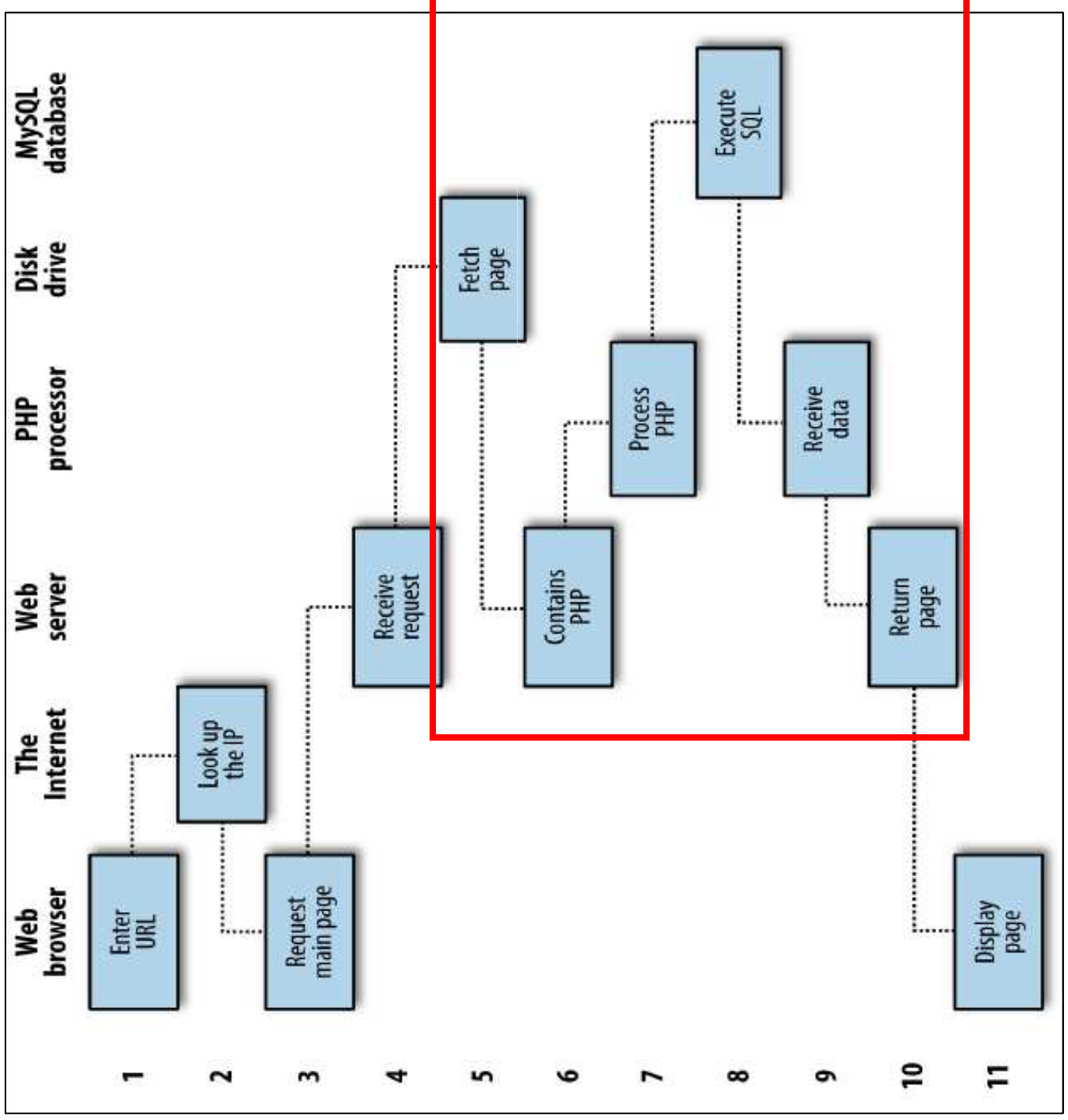


**Web browser**      **The Internet**      **Web server at *server.com***      **Disk drive at *server.com***



# Arsitektur Hypertext Markup Language

- Client (Web Browser) menerima response dari Web Server, terdiri dari bagian **header** dan **body**
  - Response header: keterangan dari hasil response dan informasi lainnya
  - Response body: bagian ini berupa teks, dalam format HTML
- Client (Web Browser) akan menampilkan (**render**) HTML yang diterima
- Kode HTML **dapat dilihat** oleh pengguna
- Apa yang terjadi jika Client (Web Browser) mengirimkan request untuk sebuah dokumen PHP?
  - Ada tambahan langkah, yaitu Web Server akan menghubungi PHP Processor untuk menjalankan script PHP
  - Termasuk menghubungi Database Server (jika diperlukan)



# Validasi HTML

- Menggunakan DTD (Document Tag Definition)
- DTD berisi spesifikasi tag-tag HTML yang benar
- Versi DTD berbeda-beda pada HTML 4
- Versi DTD sama untuk HTML 5

# Perbedaan DOCTYPE HTML dan XHTML

## HTML5

```
<!DOCTYPE html>
```

## HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```


## XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

`<!DOCTYPE>` **is not** an HTML tag. It is an **information** (a declaration) to the browser about what version the HTML is written in.

# W3C XHTML Validation Service

<http://validator.w3.org/>

**Markup Validation Service**  
Check the markup (HTML, XHTML, ...) of Web documents

Validate by **URI**

Validate by **File Upload**

Validate by **Direct Input**

### Validate by URI

Validate a document online:

Address:

► **More Options**

**Character Encoding** (detect automatically) ☐ Only if missing

**Document Type** (detect automatically) ☐ Only if missing

☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

### Validate by File Upload

Upload a document for validation:

File:

# W3C XHTML Validation Service



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#)

This document was successfully checked as XHTML 1.0 Transitional!

Result:	Passed		
Address :	<input type="text" value="http://lecturer.ukdw.ac.id/willysr/"/>		
Encoding :	iso-8859-1	<input type="text" value="(detect automatically)"/>	▼
Doctype :	XHTML 1.0 Transitional	<input type="text" value="(detect automatically)"/>	▼
Root Element:	html		
Root Namespace:	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>		



# Perbedaan HTML, CSS & Javascript

- HTML: mendefinisikan **struktur** dokumen
  - Header dan body
  - Heading, paragraph, list, table, ....
- CSS: mendefinisikan bagaimana suatu dokumen HTML **ditampilkan** di Web Browser
  - Warna, ukuran (size, margin, padding, ...), posisi, ...
  - Secara spesifik, mengatur bagaimana **suatu element tertentu** dalam HTML ditampilkan oleh **Web Browser**.
- Javascript: mendefinisikan **behavior**, bagaimana suatu dokumen HTML ‘menanggapi’ suatu tindakan (**event**). Saat ini banyak digunakan untuk interaktifitas dan komunikasi data

# HTML, CSS & Javascript

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <title>Resep Membuat Sate Kambing</title>
```

```
    <link rel="stylesheet" href="style.css">
```

```
  </head>
```

HTML

```
h1{
```

```
  color: #ff0000;
```

```
}
```

CSS



localhost/progweb/02/satekambing.html

Browser

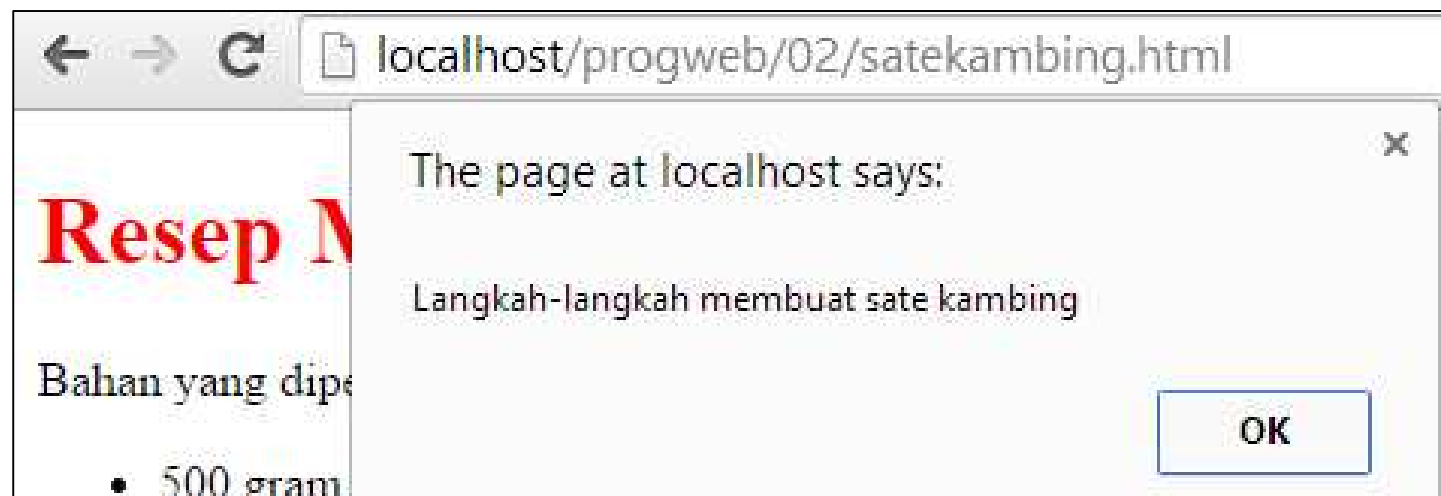
**Resep Membuat Sate Kambing**

```
<!doctype html>
<html>
  <head>
    <title>Resep Membuat Sate Kambing</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    <h1 onclick="showinfo();">Resep Membuat Sate K
```

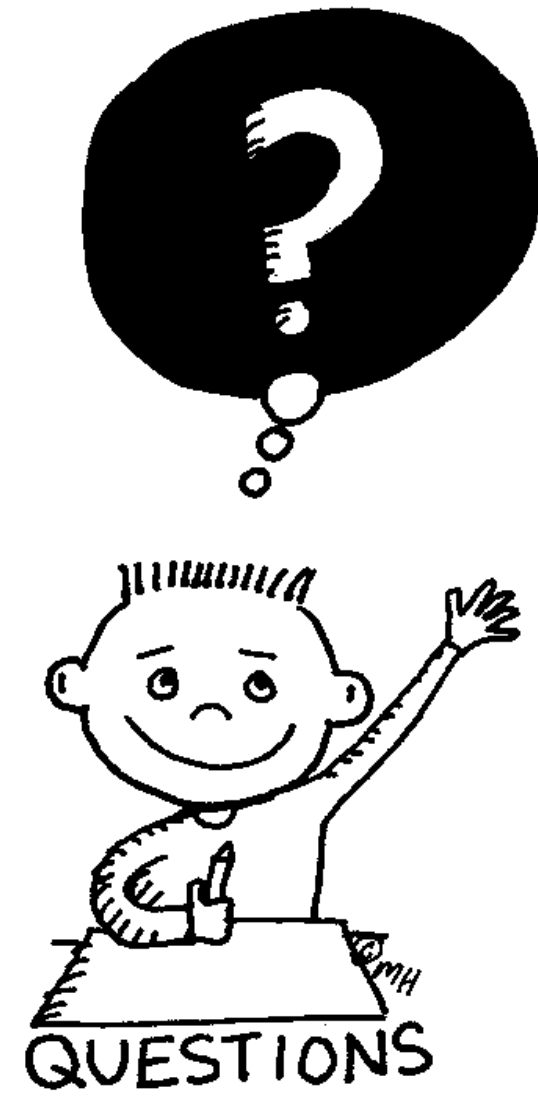
HTML

```
function showinfo(){
  alert("Langkah-langkah membuat sate kambing");
}
```

Javascript



Browser



# Struktur Dokumen HTML

```
<!DOCTYPE html>
```

→ **HTML 5**

```
<html>
```

```
<head><title></title>  
</head>
```

→ **Head**

```
<body>  
</body>
```

→ **Body**

```
</html>
```

# Struktur Dokumen HTML

- `<!DOCTYPE html>`
  - Menandakan dokumen HTML ini disusun berdasarkan spesifikasi **HTML5**
  - Case insensitive, bisa dituliskan dalam bentuk `<!doctype html>` atau `<!DOCTYPE HTML>`
  - Mulai HTML5 dan seterusnya (di masa mendatang, HTML6?) akan menggunakan doctype ini.
- Head
  - Tempat mendefinisikan properties/metadata dari dokumen HTML
  - Bagian ini tidak terlihat di Web Browser
  - Beberapa hal yang biasanya diletakkan di bagian head:
    - CSS, Javascript, title, meta, ...

# Struktur Dokumen HTML

- Body
  - Isi dari dokumen yang akan ditampilkan oleh Web Browser
  - Dalam kondisi tertentu, dimungkinkan untuk memasukkan CSS dan Javascript pada bagian Body

# Document Object Model

- The Document Object Model is a platform- and language-neutral interface that **will allow programs and scripts to dynamically access and update the content, structure and style of documents**
  - Penjelasan detail di: <http://www.w3.org/DOM/>
- Pada Web Browser, implementasi DOM dalam bentuk pohon (**tree**), biasa disebut sebagai DOM-tree
  - Setiap bagian dari DOM-tree disebut sebagai Node (atau Element)
  - Root node adalah html
- CSS dan Javascript secara umum akan mengakses DOM-tree dalam operasinya
  - **CSS Selector**



# Document Object Model Tree

```
<!DOCTYPE html>
▼ <html>
  ▼ <head>
    <title>Resep Membuat Sate Kambing</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  ▼ <body>
    <h1 onclick="showinfo();">Resep Membuat Sate Kambing</h1>
    <p>
      Bahan yang diperlukan :
    </p>
    ► <ul>...</ul>
    <p>
      Bumbu untuk celupan:
    </p>
    ► <ul>...</ul>
  </body>
</html>
```

# Element, Tag dan Attribute

`<h1>Tahu Gejrot</h1>`

**Element**

`<h1>Tahu Gejrot</h1>`

**Tag**

# Element, Tag dan Attribute

```

```

**Element**

```

```

**Tag**

**Attribute**

# Element, Tag dan Attribute

```
<ol start="8">
```

```
  <li>Pemrograman Web</li>
```

```
  <li>Sistem Operasi</li>
```

```
  <li>Struktur Data</li>
```

```
</ol>
```

# Element, Tag dan Attribute

- Element tersusun atas:
  - Opening Tag
  - Content
  - Closing Tag
- Suatu element bisa diletakkan di dalam element lainnya
  - Nested element
- Tag mendefinisikan jenis dari suatu element
  - Pada umumnya terdiri dari opening dan closing
  - Beberapa Tag tidak memerlukan closing (img, br, link, ...)
- Attribute: keterangan tambahan tentang suatu tag

# Element pada HTML

- Dalam HTML Living standard, element-element pada HTML dikelompokkan dalam banyak kategori
  - Metadata, flow, sectioning, heading, phrasing, embedded, interactive, ....
  - Lihat selengkapnya di:  
<https://html.spec.whatwg.org/multipage/indices.html#element-content-categories>
- Element-element pada HTML dapat dikelompokkan menjadi dua bagian berdasarkan visibilitasnya:
  - Non-visible element: tidak ditampilkan oleh browser
  - Visible element: ditampilkan oleh browser
- Non-visible element
  - script, link, meta, style, ....
- Visible element
  - Banyak sekali, umumnya diletakkan di bagian body

# Visible Element

- Dapat dibagi menjadi dua kategori, yaitu:
  - Block Element
  - Inline Element
- Block element
  - Takes up the full width available, with a new line before and after
- Inline element
  - Takes up only as much as it needs, and does not force new lines
- Daftar lengkap element HTML dapat dilihat di:  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

# Block Element

- `<div>` → general purpose box
- `<h1>...<h6>` → heading
- `<p>` → paragraph
- `<ul>`, `<ol>`, `<dl>` → unordered list, ordered list, definition list
- `<li>`, `<dt>`, `<dd>` → list item, definition list terms, definition list definitions
- `<table>`, `<blockquote>`, `<pre>`, `<form>`
- ..... masih banyak lainnya.



# Inline Element

- `<span>` → all purpose inline element
- `<a>` → anchor, used for links
- `<strong>` → like bold, `<b>`
- `<em>` → emphasis, replaces `<i>` (italic) tag
- `<img>` → image
- `<br>` → line break, text carries on the next line
- `<input>` → textbox, button, radiobutton, etc
- `<abbr>`, `<acronym>`
- .... masih banyak lainnya

# Inline vs Block

## Resep Sate Kambing



Bahan yang diperlukan :

- 500 gram daging kambing, potong-potong layaknya sate.

Bumbu untuk celupan:

- 100 grm kacang tanah, digoreng dan dihaluskan.
- 50 grm buah nanas, setengah matang, diparut halus.
- 2 sendok makan kecap manis.
- Campurkan semua bahan menjadi satu dan aduk hingga rata. Lalu sisihkan.

# Inline vs Block

## Resep Sate Kambing



[Index resep](#)

Bahan yang diperlukan :

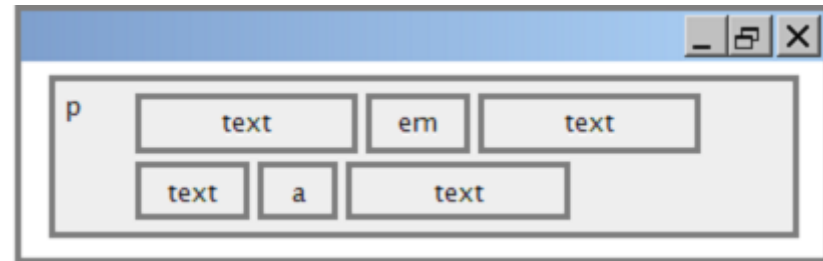
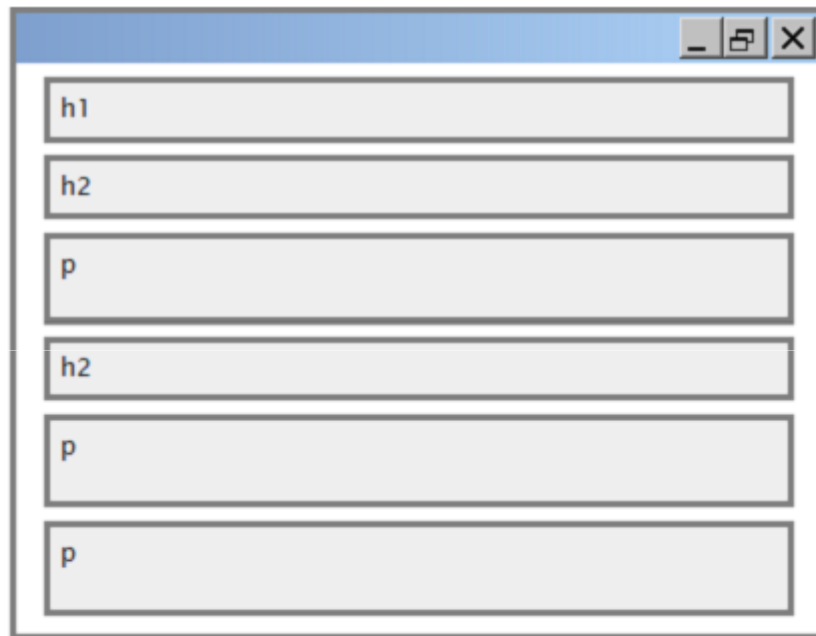
- 500 gram daging kambing, potong-potong layaknya sate.

Bumbu untuk celupan:

```

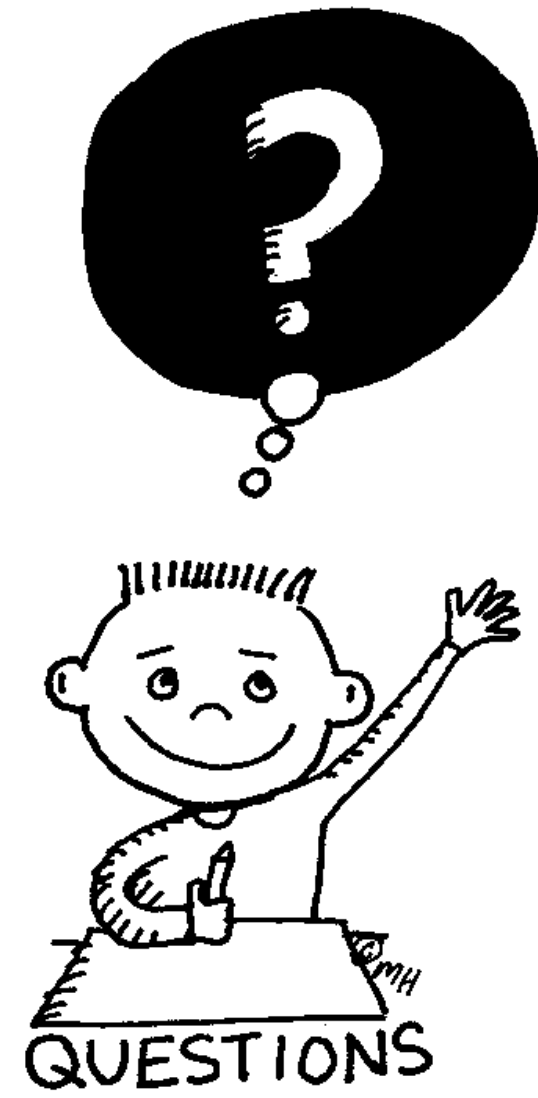
<a href="index.html">Index resep</a>
```

# Block vs Inline in picture



# Inline vs Block

- Kenapa anda perlu paham mengenai Inline dan Block element?
  - Menentukan **aliran (flow)** dari element-element pada dokumen HTML anda
  - Banyak digunakan saat mengatur tata letak (**positioning**) element saat ditampilkan oleh browser
  - Menentukan **ukuran** element saat ditampilkan
- Perilaku Inline atau Block sudah ditentukan oleh HTML standar, tapi bisa diubah dengan **CSS**
  - Misal: `h1 { display: inline; }`
  - Anda sebaiknya mengikuti standar, untuk konsistensi dan kompatibilitas.



# HTML 5 Tag

Tag	Description
<article>	For external content, like text from a news-article, blog, forum, or any other content from an external source
<aside>	For content aside from the content it is placed in. The aside content should be related to the surrounding content
<command>	A button, or a radiobutton, or a checkbox
<details>	For describing details about a document, or parts of a document
<summary>	A caption, or summary, inside the details element
<figure>	For grouping a section of stand-alone content, could be a video
<figcaption>	The caption of the figure section
<footer>	For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
<header>	For an introduction of a document or section, could include navigation
<hgroup>	For a section of headings, using <h1> to <h6>, where the largest is the main heading of the section, and the others are sub-headings
<mark>	For text that should be highlighted
<meter>	For a measurement, used only if the maximum and minimum values are known
<nav>	For a section of navigation
<progress>	The state of a work in progress
<ruby>	For ruby annotation (Chinese notes or characters)
<rt>	For explanation of the ruby annotation
<rp>	What to show browsers that do not support the ruby element
<section>	For a section in a document. Such as chapters, headers, footers, or any other sections of the document
<time>	For defining a time or a date, or both
<wbr>	Word break. For defining a line-break opportunity.

# Element heading

- Digunakan sebagai heading dari suatu bagian teks (misal judul artikel, sub judul artikel, dsb...)
- Ada 6 tingkatan:
  - h1, h2, h3, h4, h5 dan h6
  - h1 paling besar, h6 paling kecil

```
<body>  
  <h1>Heading h1</h1>  
  <h2>Heading h2</h2>  
  <h3>Heading h3</h3>  
  <h4>Heading h4</h4>  
  <h5>Heading h5</h5>  
  <h6>Heading h6</h6>  
</body>
```

**Heading h1**

**Heading h2**

**Heading h3**

**Heading h4**

**Heading h5**

**Heading h6**



# Paragraph

- Digunakan untuk isi dokumen, dalam bentuk paragraf. Merupakan block element
  - Di dalam paragraf dapat diletakkan inline element
- Contoh:

```
<body>
  <h1>Mulai Sulit Kalahkan City, Mourinho Gantikan Bungkam</h1>
  <h3>Femidiah - Detik Sport</h3>
  <p>
    Hasil seri antara <a href="http://chelsea.com">Chelsea</a>
Manchester City membuat perebutan titel Premier Legaue masih bakal seng
Rivalitas Jose Mourinho dan Manuel Pellegrini pun masih mengasyikkan ur
diikuti.
  </p>
```

# Paragraph

```
<body>
```

```
  <h1>Mulai Sulit Kalahkan City, Mourinho Gantian Bungkam</h1>
```

```
  <h3>Femidiah - Detik Sport</h3>
```

```
  <p>
```

```
    Hasil seri antara <a href="http://chelsea.com">Chelsea</a>
```

```
Manchester City membuat perebutan titel Premier Legaue masih bakal seng  
Rivalitas Jose Mourinho dan Manuel Pellegrini pun masih mengasyikkan un  
diikuti.
```

```
  </p>
```

## **Mulai Sulit Kalahkan City, Mourinho Gantian Bungkam**

### **Femidiah - Detik Sport**

Hasil seri antara [Chelsea](http://chelsea.com) dan Manchester City membuat perebutan titel Premier Legaue masih bakal sengit. Rivalitas Jose Mourinho dan Manuel Pellegrini pun masih mengasyikkan untuk diikuti.

# Table

- Digunakan untuk menampilkan tabular data
  - Sebelum CSS banyak digunakan, Table digunakan untuk mengatur layout.
- Sebuah tabel memiliki bagian-bagian:
  - Judul tabel (caption)
  - Header
  - Body/data
  - Footer
- Pengaturan bagaimana tabel ditampilkan (ukuran border, jarak antar cell, alignment, warna, ....) sebaiknya dilakukan menggunakan CSS
- Atribut-atribut dari element Table secara lengkap dapat dilihat di: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table>

# Table

- Tag-tag yang umumnya digunakan untuk menampilkan table adalah:
  - `<table> ... </table>` → penanda block table
  - `<tr> ... </tr>` → menyatakan suatu baris (table row)
  - `<td> ... </td>` → menyatakan sebuah cell (table cell)
  - `<th> ... </th>` → menyatakan sebuah cell header
- Tag-tag lainnya yang dapat digunakan antara lain:
  - `<thead> ... </thead>` → menyatakan bagian header tabel
  - `<tbody> ... </tbody>` → menyatakan bagian body tabel
  - `<tfoot> ... </tfoot>` → menyatakan bagian footer tabel
- `<thead>`, `<tbody>` dan `<tfoot>` digunakan jika tampilan bagian header, body dan footer diinginkan berbeda, dan agar mudah diubah.

# Contoh Table

```
<table>
  <tr>
    <td>NIM</td>
    <td>Nama</td>
  </tr>
  <tr>
    <td>22001234</td>
    <td>Yuan Lukito</td>
  </tr>
</table>
```

HTML

Browser

NIM	Nama
22001234	Yuan Lukito

```
td {
  border: 1px solid #000000;
}
```

CSS

# Contoh Table

```
<table>
  <thead>
    <th>NIM</th>
    <th>Nama</th>
  </thead>
  <tbody>
    <tr>
      <td>22001234</td>
      <td>Yuan Lukito</td>
    </tr>
  </tbody>
</table>
```

HTML

Browser

NIM	Nama
22001234	Yuan Lukito

```
td { border: 1px solid #000000; }
th { border: 1px solid #ff0000; }
thead { background-color: #eeeeee; }
```

CSS

# Contoh Table

CSS

```
<table>
  <tr>
    <td>Nama</td>
    <td>Minggu 1</td>
    <td>Minggu 2</td>
    <td>Minggu 3</td>
  </tr>
  <tr>
    <td>Yuan Lukito</td>
    <td colspan="2">Aktif</td>
    <td>Pasif</td>
  </tr>
</table>
```

```
td {
  border: 1px solid #000000;
}
```

HTML

Nama	Minggu 1	Minggu 2	Minggu 3
Yuan Lukito	Aktif		Pasif

Browser

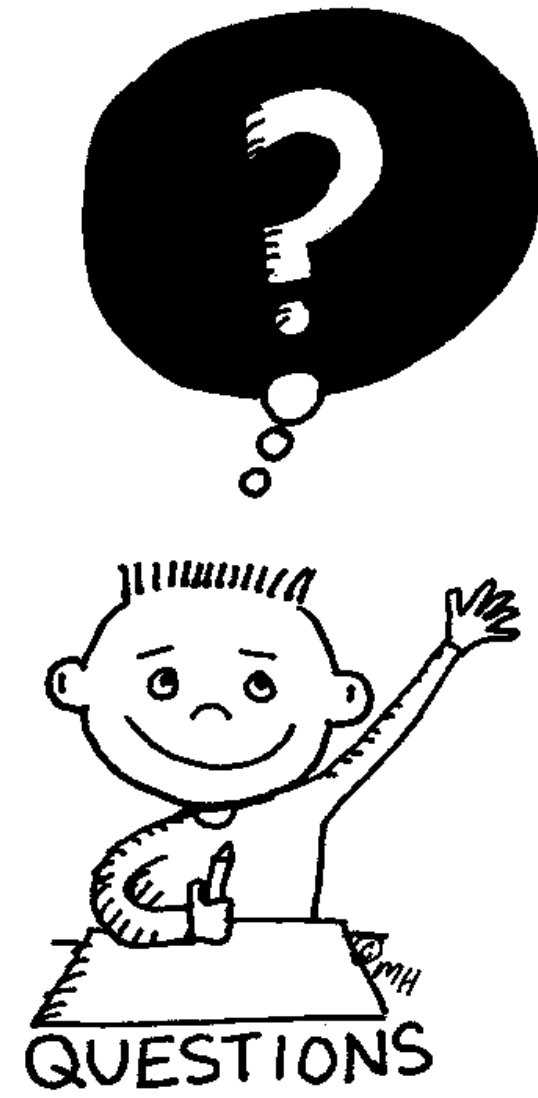
# Latihan

- Buatlah sebuah tabel yang memiliki tampilan seperti berikut:

Sesi	Senin	Selasa	Rabu	Kamis	Jum'at	
I	Kuliah	Libur	Libur	Kuliah	Libur	
II		Kuliah				Libur
III						
IV	Libur	Libur		Libur		

- Catatan: kolom dan baris harus sama persis, tampilan tidak harus sama (misal tebal garisnya beda, ukuran font-nya beda, dsb)





# Form

- Digunakan sebagai input/masukan dari pengguna aplikasi web
- Seperti umumnya pada aplikasi desktop, input/masukan dapat berupa teks, radiobutton, checkbox, combobox, ...
  - Bisa juga untuk upload file oleh pengguna
- Ada dua atribut minimal yang harus diperhatikan saat mendefinisikan form:
  - Action: menyatakan siapa yang akan memproses input tersebut. Siapa = berupa file.
  - Method: metode pengiriman input, ada dua pilihan: GET dan POST

# Contoh Form

```
<body>
<p>Masukkan NIM anda untuk masuk:</p>
<form action="proses.php" method="post">
  <label>NIM: </label>
  <input type="text" name="nim">
  <button type="submit">Masuk!</button>
</form>
</body>
```

HTML

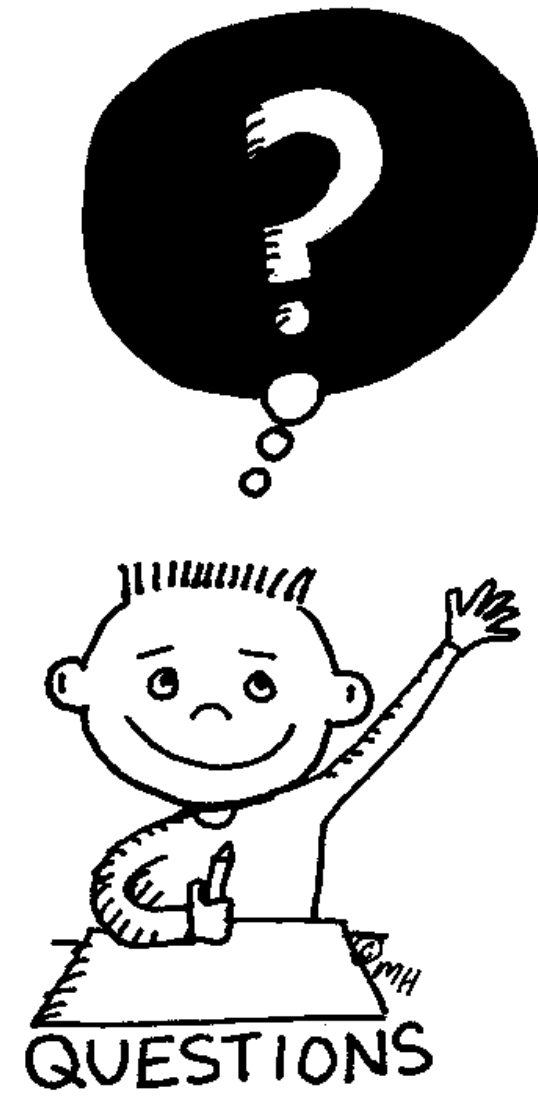
Masukkan NIM anda untuk masuk:

NIM:

Browser

# GET vs POST

- Method GET
  - Input yang anda submit akan muncul di address bar
  - ...../proses.php?nim=22001234
  - Dapat di-bookmark
- Method POST
  - Input yang anda submit tidak muncul di address bar
  - ...../proses.php
  - Tidak dapat di-bookmark
  - Jika submit data sensitive (password, ...) gunakan POST
- Demo Form handling dengan PHP
  - Perbedaan GET dan POST



# Pertemuan 03

## CSS (Cascading Stylesheet)

```
body {  
    font-size: 1.5em;  
    background-image:  
background.gif;  
}
```

```
h1{  
    font-size: 4em;  
}
```