**DESAIGN PEMPROGRAMAN WEB**

****

**POLINEMA**

Disusun oleh :

TI-2A / MOCHAMMAD RIJAL DZAKI RIFKI AFIFUDIN / 244107020240

**PRODI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-2: CSS**  **Mata Kuliah Desain dan Pemrograman Web**  Pengampu: Tim Ajar Desain dan Pemrograman Web  *September 2023* |

**Topik**

* CSS

**Tujuan**

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat web statis menggunakan HTML dan CSS
2. Mahasiswa mampu menerapkan konsep CSS *Display*
3. Mahasiswa mampu menerapkan konsep CSS Box Model
4. Mahasiswa mampu menerapkan konsep CSS *Flex Box*
5. Mahasiswa mampu menerapkan konsep CSS *Grid*

**Perhatian**

Jobsheet ini harus dikerjakan step-by-step sesuai langkah-langkah praktikum yang sudah diberikan.

**Apakah CSS ?**

CSS merupakan singkatan dari “*Cascading Style Sheets*“. Sesuai dengan namanya CSS memiliki sifat ”*style sheet language*” yang berarti bahasa pemrograman yang di gunakan untuk web design. CSS adalah bahasa pemrograman yang di gunakan untuk mendesign sebuah halaman website. Dalam mendesign halaman website, CSS menggunakan penanda yaitu **id** dan **class**. CSS dapat mengubah font, ukuran font, warna dan format font, mengatur ukuran layout, lebar, tinggi dan warna element, mengubah tampilan form, membuat halaman website yang *responsive* dan masih banyak lagi.

Untuk mendesain font dapat dilakukan dengan mendefinisikan font, untuk mengatur warna bisa menggunakan color, margins digunakan untuk mengatur jarak pada luar element tertentu. mengatur warna atau gambar pada latar belakang bisa menggunakan “*background*”. mengatur ukuran *font* gunakan “*font size*”. jenis *font* menggunakan “*font-family*” dan banyak lagi lainnya.

**Cara Menggunakan CSS**

File css di simpan dengan ekstensi **.css**. kemudian di import atau di hubungkan kedalam file HTML atau PHP yang ingin kita design dengan CSS menggunakan syntax berikut ini:

<link rel="stylesheet" type="text/css" href="style.css"/>

Tag atas di gunakan untuk menghubungkan file HTML dengan file CSS. Syntax di letakkan pada file html. Pada atribut rel dan type di tag link di gunakan untuk mendefinisikan bahwa yang di panggil atau yang di hubungkan adalah file stylesheet atau CSS, kemudian atribut href di gunakan untuk meletakkan letak file CSS. pada contoh di atas file style.css terletak satu folder atau satu direktori dengan file html. jika file css terletak di luar folder maka bisa menghubungkanya dengan:

href="../style.css"

Jika file css terletak dalam sebuah folder, misalkan nama foldernya adalah ”assets“, maka untuk menghubungkannya dengan:

href="../assets/style.css"

**Praktikum Bagian 1. Menghubungkan HTML dengan CSS**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah satu file baru di dalam direktori dasarWeb, beri nama index.html. |
| 2 | Ketikkan ke dalam file index.html tersebut kode di bawah ini. |
| 3 |  |
| 4 | Buat lagi satu file baru di dalam direktori dasarWeb, beri nama style.css. Ketikkan kode di bawah ini di dalam file style.css. |
| 5 |  |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb |
| 7 | Dalam struktur halaman web, index.html akan selalu menjadi halaman utama sebuah web. Sehingga jika dalam sebuah direktori terdapat index.html maka dia akan selalu dipanggil dan ditampilkan di browser. Itulah sebabnya pada langkah 6 hanya diketik halaman localhost/dasarWeb tanpa nama file. |
| 8 | Pada kode program di dalam index.html terdapat tag <link> di bagian head di mana atribut href merujuk ke halaman style.css. |
| 9 | style.css berisi kode program untuk mengatur tampilan di dalam halaman yang merujuk ke file style.css, dalam praktikum ini adalah index.html. Sehingga tampilan di dalam index.html akan sesuai dengan properti yang diatur di dalam style.css. |

**Section menggunakan tag <div>**

Tag <div> mendefinisikan sebuah bagian dalam dokumen HTML. Elemen <div> sering digunakan sebagai sebuah kontainer untuk elemen HTML lain untuk menambahkan *style* dengan CSS atau untuk menampilkan tugas tertentu menggunakan JavaScript.

**Praktikum 2: Penggunaan div**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama div.html di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam div.html |
| 2 |  |
| 3 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/div.html |
| 4 | Apa yang anda pahami dari penggunaan div pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1) |

**Mengenal class dan id pada HTML**

Class dan id digunakan sebagai penanda pada html, yang dimaksud dengan penanda di sini adalah element-element html dapat di beri tanda dengan class atau id. Elemen-elemen dalam HTML diberi tanda agar dapat di manipulasi menggunakan css atau javascript. Bayangan sederhananya jika anda memiliki lima buah kotak, semua kotak yang anda miliki berwarna biru, kemudian anda ingin mengubah warna kotak yang ketiga, nah di sinilah letak kegunaan class dan id, untuk memberikan tanda atau nama pada kotak anda agar bisa di ubah dan kotak yang lain tidak akan berubah.

Perbedaan dari class dan id adalah class di panggil pada css atau javascript dengan menggunakan tanda titik “.”, dan id di panggil pada css atau javascript dengan tanda pagar “#”, ada kelebihan di sini untuk class, yaitu class dapat di berikan pada banyak element html dan dapat di panggil sekaligus, sedangkan id hanya dapat bekerja pada satu penandaan saja, maksudnya satu nama id hanya bisa di berikan pada satu element saja.

**Praktikum Bagian 3: Mengenal class dan id pada HTML**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Lengkapi kode program di dalam index.html menjadi seperti kode pada langkah 2 |
| 2 |  |
| 3 | Lengkapi kode program di dalam file style.css. sehingga menjadi seperti kode pada langkah 4 |
| 4 |  |
| 5 | Simpan kedua file tersebut, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 6 | Apa yang anda pahami dari penggunaan class dan id dalam index.html. Catat di bawah ini pemahaman anda. (soal no 2)  id digunakan untuk menandai satu elemen yang unik di seluruh halaman, sedangkan class digunakan untuk mengelompokkan beberapa elemen yang memiliki gaya atau fungsi serupa. |

**Cara Penulisan CSS**

CSS menggunakan selector (id dan class) untuk menentukan element yang akan di modifikasi atau yang akan di beri sentuhan css, jika di ibaratkan HTML sebagai tiang pada sebuah bangunan rumah, maka CSS berfungsi sebagai cat dan dekorasi pada bangunan rumah tersebut. Ada tiga teknik metode penulisan CSS, yaitu:

* *Inline CSS Style*

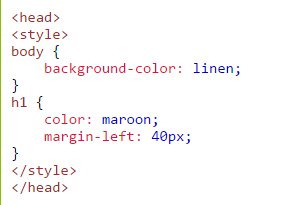
Adalah CSS yang dibuat dalam sebuah tag HTML yang hanya berlaku untuk dokumen yang diapitnya saja. Biasanya teknik ini digunakan pada pemformatan khusus pada sebuah elemen HTML dan tidak digunakan untuk memformat seluruh elemen dalam dokumen web.

Contoh:

<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>

* *Internal CSS Style*

Sebuah internal style sheet untuk sebuah halaman web hanya berlaku pada sebuah halaman web tersebut saja. Internal style sheet didefinisikan pada bagian HEAD sebuah halaman HTML, di dalam tag <style> seperti berikut:



Gambar 1. Internal CSS

* *External CSS Style*

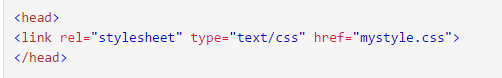
Sangat ideal digunakan pada web dengan banyak halaman. Dengan menggunakan External Style Sheet tampilan seluruh isi website dapat diubah hanya dengan mengubah satu file. *External Style Sheet* tidak boleh mengandung tag html, dan disimpan dalam file dengan ekstensi \*.css

Contoh:



Gambar 2. External CSS

Cara pemanggilan *External Style Sheet* dalam sebuah halaman web yaitu dengan menggunakan tag <link> dengan atribut rel yang diletakkan pada *section head*. Contoh:



Gambar 3. Menggunakan file external CSS ke dalam HTML

**Praktikum Bagian 4: Mengubah *Background* Halaman Web dengan CSS**

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Tambahkan potongan kode program pada langkah 2 di bawah pada file style.css |
| 2 |  |
| 3 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 3) |

**Praktikum Bagian 5: Menggunakan *image* untuk *Background***

Properti *background-image* menetapkan satu atau lebih gambar *background* untuk suatu elemen. Secara default, gambar latar belakang ditempatkan di sudut kiri atas elemen, dan diulang secara vertikal dan horizontal.

Tip: Latar belakang suatu elemen adalah ukuran total elemen, termasuk *padding* dan *border* (tetapi bukan margin).

Tip: Selalu atur warna latar untuk digunakan jika gambar tidak tersedia.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Pada file style.css ubah value dari background menjadi bunga2.jpg seperti pada potongan kode pada langkah 2 |
| 2 |  |
| 3 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4) |
| 5 | Tambahkan file gambar bernama field1.jpg dan tree1.jpg pada folder img. |
| 6 | Ubah kode program pada style.css menjadi seperti pada langkah 7 |
| 7 |  |
| 8 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Kode program anda berjalan baik jika tampilan seperti di bawah ini : |
| 10 | Catat apa yang kalian pahami dari perubahan tampilan yang terjadi. (soal no 5) |

**Praktikum Bagian 6: Margin dan Padding pada CSS**

* Mengenal Margin pada CSS

Margin adalah sisi luar dari sebuah element. misalnya anda ingin mengatur jarak antar element. anda bisa menggunakan *syntax margin* untuk mengaturnya. Terdapat beberapa sisi luar margin yaitu,

* margin atas di tuliskan pada css dengan ‘margin-top’,
* margin bawah atau jarak luar bagian bawah di tulis di CSS dengan ‘margin-bottom’,
* ‘margin-left’ sebagai jarak luar sebelah kiri element, dan
* ‘margin-right’ adalah sisi luar pada bagian sebelah kanan

Tetapi jika anda hanya menggunakan syntax ‘margin’ saja maka akan secara otomatis mengatur jarak atas,bawah, kiri dan kanan element.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi backgroundCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama styleMargin.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program berikut ke dalam styleMargin.css |
| 7 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 8 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 6) |
| 9 | Ubah nilai margin pada pengaturan .box menjadi margin : 300px;. Amati apa perbedaannya. |
| 10 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 7) |

* Mengenal *Padding* dalam CSS

Padding adalah sisi dalam dari sebuah element. Kita bisa menggunakan syntax padding untuk mengatur jarak pada sisi dalam sebuah element yang kita tentukan. Sama seperti margin yang memiliki sisi-sisi. seperti top, left, right, bottom. jenis padding yaitu padding atas di tuliskan pada css dengan ‘padding-top’ yang berarti mengatur sisi dalam sebelah atas sebuah element, pading bawah atau jarak dalam bagian bawah di tulis di CSS dengan ‘padding-bottom’, ‘padding-left’ sebagai jarak dalam sebelah kiri element, dan ‘padding-right’ adalah sisi luar pada bagian sebelah kanan. jika anda hanya menggunakan syntax ‘padding’ saja maka akan secara otomatis mengatur jarak atas,bawah, kiri dan kanan element yang bagian dalam.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi marginCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama stylePadding.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam stylePadding.css. |
| 7 |  |
| 8 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 8) |
| 10 | Ubah nilai padding pada pengaturan .box menjadi padding=200px; |
| 11 | Catat di sini kesimpulan apa yang anda amati dari kode program di atas. (soal no 9) |

**Praktikum Bagian 7: Pengaturan Font pada CSS**

Beberapa syntax css yang digunakan untuk mengatur font:

* *font-size* digunakan untuk mengatur ukuran font
* *font-weight* di gunakan untuk mengatur ketebalan font
* *font-family* untuk mengubah jenis font
* *font-style* digunakan untuk merubah gaya pada font.
* *color* digunakan untuk merubah warna font

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi paddingCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama styleFont.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam styleFont.css. |
| 7 |  |
| 8 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 10) |

**Praktikum Bagian 8: Mengatur *Hyperlink* dengan CSS**

*Hyperlink* merupakan link yang dibuat untuk mengalihkan halaman saat di klik. *Hyperlink* atau link dibuat dengan menggunakan tag dan di akhiri dengan tag di HTML. Ada 4 status yang di miliki oleh *hyperlink* html dan bisa di manipulasi dengan menggunakan css. yaitu:

* *link*. merupakan link aktif biasa.
* *visited*. merupakan status sebuah link yang telah di kunjungi.
* *hover*. merupakan status sebuah link pada saat diletakkan cursor mouse di atasnya.
* *active*. merupakan status sebuah link atau *hyperlink* pada saat sudah di klik.

*Syntax* nya adalah sebagai berikut:

* *a:link* = untuk link biasa
* *a:visited* = merupakan status sebuah link yang telah di kunjungi.
* *a:hover* = merupakan status sebuah link pada saat diletakkan cursor mouse di atasnya.
* *a:active* = merupakan status sebuah link atau hyperlink pada saat sudah di klik.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi fontCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama styleLink.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam styleLink.css. |
| 7 |  |
| 8 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 11) |
| 10 | Tambahkan kode pada styleLink.cssuntuk mengubah warna link menjadi hijau muda (greenyellow) setelah link dikunjungi. |
| 11 | Tulis kode program yang anda tambahkan di sini (soal no 12)  .link{      font-size: 20pt;  }  .link:hover{      color: red;  }  .link,:link{      color: blue;  }  .link:visited{      color: greenyellow  } |

**Praktikum Bagian 9: Mengatur Format Text dengan CSS**

Pengaturan format text di dalam CSS adalah sebagai berikut:

1. *color*: untuk mengatur warna text, value yang dapat diisi berupa warna atau kode warna
2. *text-align:* untuk mengatur posisi align pada text atau rata text, value yang bisa diisi di antaranya adalah center untuk membuat text rata tengah, left untuk membuat text rata kiri, right untuk membuat text menjadi rata kanan dan justify untuk membuat text menjadi rata kanan dan rata kiri.
3. *text-decoration:* untuk mengatur dekorasi text, valuenya berupa none untuk membuat text tidak memiliki dekorasi, overline untuk membuat text memiliki garis pada bagian atas text, line-through untuk membuat garis yang mencoreng pada text, dan underline untuk membuat garis pada bawah text (garis bawah).
4. *text-transform*: untuk mengatur huruf kapital pada text, value yang bisa di gunakan diantaranya adalah uppercase untuk membuat text menjadi huruf besar, lowercase untuk membuat text menjadi huruf kecil, dan capitalize untuk membuat huruf awal pada tiap kata menjadi huruf besar.
5. *text-indent:* untuk mengatur jarak alinea pada text, value yang bisa digunakan berupa nilai pixel dan lainnya sesuai kebutuhan.
6. *letter-spacing*: untuk mengatur jarak antar karakter pada text, value yang di isi berupa nilai pixel dan lain-lain.
7. *word-spacing:* untuk mengatur jarak antar kata pada text, value yang di isi juga berupa nilai pixel.
8. *line-height*: untuk mengatur jarak antar baris pada text value yang di isi berupa nilai.
9. *text-shadow*: untuk mengatur efek bayang pada text, value yang di isikan pertama mengisi nilai untuk jarak kiri kanan, dan kedua mengisi jarak atas bawah dan yang ketiga mengisi warna. untuk contoh penulisanya 2px 5px blue.
10. *vertical-align*: untuk mengatur align dalam bentuk vertikal pada text value yang digunakan adalah left untuk membuat text rata kiri, right untuk rata atas dan center untuk rata tengah.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi linkCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama styleText.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam styleText.css. |
| 7 |  |
| 8 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas. (soal no 13) |
| 10 | Tambahkan kode pada styleText.css untuk mengubah posisi text menjadi di tengah dengan menambahkan text-align:center; dan menambahkan dekorasi berupa garis bawah. |
| 11 | Tulis kode program yang anda tambahkan di sini (soal no 14)  .tulisan\_warna{      color: blue;      text-align: center;  }  .par1{      text-align: center;  } |
| 12 | Tambahkan kode pada styleText.css untuk menambahkan garis bawah pada text. |
| 13 | Tulis kode program yang anda tambahkan di sini (soal no 15)  .tulisan\_warna{      color: blue;      text-align: center;      text-decoration: underline;  }  .par1{      text-align: center;  } |
| 14 | Tambahkan kode program pada styleText.css untuk memberikan jarak antar karakter pada paragraf yang ada menjadi 5px dengan letter-spacing. |
| 15 | Tulis kode program yang anda tambahkan di sini (soal no 16)  .tulisan\_warna{      color: blue;      text-align: center;      text-decoration: underline;  }  .par1{      text-align: center;      letter-spacing: 5px;  } |

**Praktikum Bagian 10: Mengenal Position CSS**

Position pada CSS di gunakan untuk mengatur posisi sebuah element HTML. Properti position CSS ini digunakan untuk menentukan posisi sebuah element HTML sesuai dengan yang diinginkan.

Secara umum untuk membuat posisi atau menetapkan posisi sebuah element kita pasti menggunakan properti css lainnya seperti mengatur top, left, bottom, right untuk menetapan posisi sebuah element, tetapi properti tersebut tidak akan bekerja jika position belum di atur terlebih dahulu, hal ini dikarenakan properti lain tergantung dengan position yang di tetapkan.

Beberapa property CSS yang dapat digunakan untuk menentukan posisi sebuah element HTML adalah:

* *Static*: Position static digunakan untuk mengatur element menjadi statis secara default. Elemen akan mengikuti posisi normal secara default, elemen tidak dipengaruhi oleh properti seperti top, bottom, left dan right.
* *Relative*: Sebuah element HTML yang menggunakan position relative akan terletak pada posisi normal. Mengatur properti atas, kanan, bawah, dan kiri dari elemen yang diposisikan dengan “relatif” akan membuatnya jauh dari posisi normalnya. Konten lain tidak akan disesuaikan agar sesuai dengan celah yang ditinggalkan oleh elemen tersebut.
* *Fixed*: Sebuah element HTML yang di setting dengan position fixed akan memiliki sifat tetap. tanpa ada perubahan bahkan jika halaman website di scroll. Berlaku pengaturan left, bottom, top dan right pada position fixed.
* *Absolute*: Element HTML yang menggunakan position absolute akan diposisikan relatif dengan element lain yang mendahuluinya yang terdekat, bukan relatif terhadap layar secara normal.
* *Sticky*: Unsur dengan posisi: sticky; diposisikan berdasarkan posisi scroll pengguna. Elemen sticky berganti-ganti antara relatif dan tetap, tergantung pada posisi scroll. Ini diposisikan relatif sampai posisi offset tertentu bertemu di viewport - lalu "menempel" di tempatnya (seperti posisi: fixed).

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi textCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama stylePosition.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam stylePosition.css. |
| 7 |  |
| 8 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas dengan bahasamu. (soal no 17) |

**Praktikum Bagian 11: Menggunakan Float**

Teknik Floating pada bagian web design merupakan sebuah kebutuhan yang paling banyak di perlukan. Properti float digunakan untuk memposisikan dan memformat konten, misal. membuat gambar melayang ke sebelah kiri teks dalam kontainer. Salah satu contoh penggunaan float yang paling sering di temukan adalah ketika kita ingin mebuat gaambar postigan website yang terletak di bagian samping tulisan konten. Properti float dapat memiliki salah satu dari nilai berikut:

* kiri - Elemen mengapung di sebelah kiri wadahnya
* kanan- Elemen mengapung di sebelah kanan wadahnya
* tidak ada - Elemen tidak mengambang (akan ditampilkan tepat di tempat teks tersebut muncul). Ini standar
* *inherit* - Elemen ini mewarisi nilai float dari induknya

Dalam penggunaannya yang paling sederhana, properti float dapat digunakan untuk membungkus teks di sekitar gambar.

Langkah-langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Rename index.html menjadi positionCSS.html |
| 2 | Buat sebuah file baru di dalam folder dasarWeb, beri nama index.html. Ini berarti anda mempunyai satu file index.html yang baru. |
| 3 | Ketik kode program di bawah ini ke dalam index.html |
| 4 |  |
| 5 | Buat satu file baru bernama styleFloat.css di dalam folder dasarWeb |
| 6 | Ketikkan kode program pada langkah 7 berikut ke dalam styleFloat.css. |
| 7 |  |
| 8 | Simpan file, kemudian buka browser dan jalankan/refresh localhost/dasarWeb |
| 9 | Catat di sini apa yang anda amati dari kode program di atas dengan bahasamu. (soal no 18) |

**Apa itu CSS *Layouting* ?**

CSS *layouting* adalah tehnik untuk mengatur tata letak sebuah halaman web dengan menggunakan kode CSS. Tujuan dari CSS Layouting adalah agar halaman yang dibuat terlihat rapih, menarik sesuai dengan yang diharapkan oleh si pembuat. CSS *layouting* terdiri dari beberapa bagian yaitu; dimensi, *overflow*, Box model, float dan Position

**CSS *Display***

Tag pada HTML digunakan untuk memberikan ‘maksud’ / ‘arti’ pada sebuah konten (contohnya adalah p untuk paragraph, h1 untuk *heading* utama dan lain-lain). Tag <div> dan tag <span> tidak memiliki arti apapun, keduanya digunakan untuk mengelompokkan tag-tag HTML dan memberikan informasi terhadap tag-tag tersebut.

**Praktikum Bagian 12. CSS *Layouting***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah satu file baru di dalam direktori dasarWeb, beri nama index.html. |
| 2 | Ketikkan ke dalam file index.html tersebut kode di bawah ini. |
| 3 |  |
| 4 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb |
| 5 | Kemudian buatlah <div> pada halaman html yang sama seperti pada kode program di bawah ini |
| 6 | A screenshot of a computer  Description automatically generated |
| 7 | Amati hasil dari kedua program tersebut sama atau berbeda., jelaskan alasanya **(soal 19)**  Kode program pertama tanpa menggunakan div, program kedua menggunakan div.  Div digunakan untuk mengelompokan beberapa kumpulan elemen |
| 8 | Tambahkan *style* pada *class* **navigasi** dan *class* **main** seperti pada kode program di bawah ini |
| 9 |  |
| 10 | Jalankan program *capture* hasilnya dan jelaskan apa yang terjadi **(soal 20)** |

**Value dari *display***

* Display *inline*

Elemen HTML yang secara *default* tidak menambahkan baris baru Ketika dibuat. Karakteristik dari *dispay inline* adalah

1. lebar dan tinggi elemenya besarnya sesuai dengan konten yang ada di dalamnya
2. Tidak dapat mengatur tinggi dan lebar dari elemen *inline*
3. *Margin* dan *padding* hanya mempengaruhi elemen secara horizontal, tidak vertical

Adapun elemen-elemen *inline* adalah sebagai berikut; **b, strong, i, em, a, span, sub, sub, button, input, label, select, textarea**

**Praktikum Bagian – 13. Display *Inline***

|  |  |  |
| --- | --- | --- |
| **Langkah** | | **Keterangan** |
| 1 | Silakan hapus *style* pada soal no.2. kemudian tambahakan link 5 pada *class* navigasi seperti kode program di bawah ini | |
| 2 | A screenshot of a computer  Description automatically generated | |
| 3 | Jalankan program *capture* hasil dan amati apa yang terjadi. **(Soal 21)** | |

* *Display inline-block*

Jika pada elemen *inline* kita tidak bisa mengatur tinggi dan lebar dari suatu elemen maka kita bisa mengaturnya dengan menggunaka *inline-block*. tidak ada elemen yang secara *default* memiliki *property***.**

**Praktikum Bagian – 14. Display *inline-block***

|  |  |  |
| --- | --- | --- |
| **Langkah** | | **Keterangan** |
| 2 | ambahkan *weight*, *hight* dan *display* pada *style* di elemen **a** seperti pada kode program dibawah ini | |
| 2 | A screen shot of a computer  Description automatically generated | |
| 3 | Jalankan program *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 22)** | |

* *Display Block*

*Block* adalah elemen HTML yang secara *default* menambahkan baris baru Ketika dibuat Jika tidak diatur lebar-nya, maka lebar *default* dari elemen *block* akan memenuhi lebar dari *browser* / *parent*-nya sehingga kita dapat mengatur tinggi dan lebar dari elemen *block.* Di dalam elemen *block*, kita dapat menyimpan tag dengan elemen *inline*, *inline-block*, atau bahkan elemen *block* lagi. Adapun contoh-contoh elemen *block* adalah ; h1-6, p, ol, ul, li, form, hr, div

**Praktikum Bagian – 15. Display *block***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Beri *style* pada elemen h1, h2 dan p dengan *background-color* warna *lightgreen* seperti pada kode program di bawah ini |
| 2 | A screenshot of a computer program  Description automatically generated |
| 3 | Jalankan program dan coba resize halaman web tersebut. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 23)**    Pada kode program tersebut semua class h1, h2, dan p akan mempunyai background color lightgreen dan semua class a akan mempunyai background color pink. |
| 4 | Kemudian selanjutnya silakan tambahkan style pada class main seperti pada kode program dibawah ini |
| 5 | A screenshot of a computer program  Description automatically generated |
| 6 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 24)**    Pada program tersebut semua class main akan memiliki desain ukuran 600px dan mempunyai background lightgreen kecuali class h2 mempunyai background pink. |

* *Display none*

Untuk *value display* yang terakhir adalah *none*, none ini dapat digunakan untuk menghilangkan sebuah elemen

**Dimensi dan *Overflow* pada CSS**

Dimensi memiliki dua *property* di CSS yaitu *width* untuk lebar dan *height* untuk tinggi. Satuan dari dimensi ada macam *macam* diantaranya adalah **px, %, in, cm, mm pc pc**.

*Overflow* adalah *property* CSS yang digunakan untuk mengatur perilaku elemen yang tidak cukup pada suatu *parent.* Ada empat *value* dari *property overflow* diantaranya yaitu;

1. *Visible* : *Value default*
2. *Auto* : CSS akan secara otomatis akan menambahkan *scroll* jika konten tidak cukup
3. *Hidden* : konten akan disembunyikan atau tidak kelihatan
4. *Scroll* : seperti auto, akan memunculkan *scroll*, tetapi jika *content* cukup scroll akan tetap ada

***Box model* pada CSS**

Setiap elemen di halaman *website* berada di dalam sebuah *box* (kotak). Kita bisa mengatur ukuran dan posisi kotak tersebut. Kita bisa memberi warna / gambar sebagai *background* kotak tersebut. Box model pada CSS mendefinisikan ‘kotak’ yang dihasilkan oleh sebuah elemen, lalu menampilkannya sesuai dengan format visualnya. CSS box model terdiri dari 4 komponen yaitu; margin, border, padding dan *content* seperti pada gambar di bawah ini.

A screenshot of a computer

Description automatically generated

Gambar 1. Komponen Box Model

1. *Margin*: area transparan di sekitar kotak (diluar *border*)
2. *Border*: batas disekeliling *conten* dan *padding*
3. *Padding*: area transparan di dalam kotak (antara *content* dan *border*)
4. *Content*: konten sebenarnya di dalam *box*, bisa berupa teks atau gambar

Cara mengatur propertinya seperti pada tabel 1 dibawah ini

Tabel 1. Cara mengatur property box model

|  |  |  |
| --- | --- | --- |
| ***Margin*** | ***Padding*** | ***Border*** |
| *Margin-top*  *Margin-right*  *Margin-bottom*  *Margin-left*  *margin* | *Padding -top*  *Padding -right*  *Padding -bottom*  *Padding -left*  *Padding* | *Border -top*  *Border -right*  *Border -bottom*  *Border -left*  *Border* |

* *Box Model: Margin*

*Ovelapping* margin

Terjadi Ketika kita menggabungkan dua buah margin. Kiri dan kanan atau atas dengan bawah. Secara teori jika terjadi maka akan di ambil nilai yang paling besar.

*Negative* margin akan membuat kotak berlawanan arah. Bisa digunakan jika kita ingin menyembunyikan elemen.

*Auto* margin adalah nilai yang dapat kita kasih kedalam margin khusus untuk margin kiri dan margin kanan. Auto ini akan membuat elemenya berada di tengah-tengah halaman web browser

*Shorthand* margin adalah cara menyingkat penulisan margin

**Praktikum Bagian – 16. *Box Model: Margin***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah 2 file seperti pada gambar di bawah ini. File yang pertama dengan nama index.html dan file yang kedua style.css |
| 2 | A screenshot of a computer program  Description automatically generated |
| 3 | Jalankan program, amati hasilnya dan jelaskan **(Soal 25)**    Class satu memiliki lebar dan tinggi 200px dan background lightgreen  Class dua memiliki lebar dan tinggi 100px dan background lightblue  Class tiga memiliki lebar dan tinggi 50px dan background violet |
| 4 | Selanjutnya adalah memberi *margin*, dimana *margin* adalah *area transparan* yang ada di sekitar kotak. Tambahkan *style* pada style.css **margin-top** sebesar 100px pada **class.satu** simpan dan kemudian jalankan pada web browser. Capture dan amati hasilnya  **(Soal 26)** |
| 5 | Kemudian tambahkan ukuran ***margin*** yang lain seperti pada kode program berikut ini; |
| 6 |  |
| 7 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 27)**    Posisi class satu berubah |

* *Box Model: Padding, Border & Box Sizing*

*Padding*

Cara pakai *padding* sama seperti margin yaitu, tidak bisa di pakai *negative*, tidak bisa di beri nilai *auto* dan mempengaruhi ukuran dari *box* dari suatu elemen.

*Border*

Cara menulisnya;

*Border: width style* *color*;

*Style* pada border; solid, dotted (titik titik), dashed(garis-garis), double

*Box sizing*

*Box sizing* adalah properti yang menerima nilai *padding* dan *border* pada suatu elemen termasuk sebagai nilai total dari *width* dan *height* suatu elemen.

*box-sizing*: *content-box* (*default*)| *border-box* | *unser* | *initial* | *inherit*;

***Flex Box***

Merupakan Model layout 1 dimensi yang dapat mengatur jarak dan penjajaran antar item dalam sebuah *container*.Yang dimaksud dengan satu dimensi adalah hanya dapat mengatur satu dimensi pada saat tertentu, antara baris atau kolom, tidak bisa keduanya sekaligus.

*Flex Box* atau bisa di sebut juga *Flexbox Layout Module* adalah sebuah modul yang menawarkan cara yang efektif untuk Menyusun, mensejajarkan dan mendistribusikan jarak antar item di dalam sebuah *container*, meskipun ukuranya dinamis atau bahkan kita tidak tahu.

A diagram of a blue rectangular object with white text

Description automatically generated

Gambar 2 Istilah-istilah pada flex Box

(sumber; <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>)

* ***Main axis***; Sumbu utama dari sebuah *container* yang menentukan urutan dari penempatan item secara horizontal
* ***Main start****/****main end***; Mulai dan berakhirnya items yang disimpan di dalam *container*
* ***Main size***; ukuran (*width/height*) dari *container* yang akan membuat dimensi dari items nya relative terhadap size

***Property* pada *container***

*Container* adalah pembungkus dari elemen element

A cartoon of a purple rectangular object

Description automatically generated

Gambar 3 Container

***Display***

A blue background with white text

Description automatically generated

*Display* Membuat sebuah elemen *parent* menjadi flex box, dan membuat elemen elemen di dalamnya bisa berprilaku flex juga.

***Flex- direcrtion***

*Flex-direction* Mengatur arah / urutan dari items di dalam *container*

A diagram of orange squares

Description automatically generated with medium confidence

Gambar 4 Flex- direcrtion

A screen shot of a computer

Description automatically generated

***Flex-wrap***

Secara *default*, semua *items* di dalam *container* akan berada pada satu baris meskipun ukuranya sudah sudah tidak cukup, *wrap* memungkinkan untuk memindahkan items ke baris di bawahnya

A purple and orange rectangular object with white dotted line

Description automatically generated

Gambar 5 Flex-Wrap

A blue background with yellow and orange text

Description automatically generated

Untuk property yang lain bisa kunjungi website beriku <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

**Praktikum Bagian – 17. *Flex Box***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ketikkan kode program di bawah ini kemudian simpan file dengan nama flexbox.html |
| 2 | A screenshot of a computer screen  Description automatically generated  A screenshot of a computer  Description automatically generated |
| 3 | Dan untuk file style.css sebagai berikut |
| 4 | A computer screen shot of a black screen  Description automatically generated |
| 5 | Jalankan program, amati hasilnya dan jelaskan **(Soal 28)** |
| 6 | Tambahkan *style* pada class **container-satu** pada file style1.css seperti pada kode program di bawah ini |
| 7 | A screen shot of a computer program  Description automatically generated |
| 8 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 29)**    Semua elemen yang beradi di div container-satu jadi berada di tengah |
| 9 | Tambahkan property **display** pada selector container-satu dengan value **flex**. Capture dan jelaskan hasil nya **(Soal 30)**    Dengan di tambahkan display flex, maka space kosong akan terisi oleh elemen lain(sesuai size boxnya) |

**CSS *Grid***

*CSS grid layout module* menawarkan *system* *layout* berbasis grid dengan baris dan kolom sehingga mempermudah melakukan desain halaman web tanpa menggunakan float dan position. Merupakan modul CSS baru untuk mendefinisikan system layout berbentuk grid dalam 2 dimensi (baris dan kolom)

A screenshot of a computer

Description automatically generated

Gambar 6 Grid Layout Module

*Grid element*

Grid layout terdiri dari satu elemen induk dengan satu atau lebih elemen anak

*CSS Grid Terminology*

* *Grid container:* Element pembungkus grid, didefinisikan dengan menuliskan: display: grid;
* *Grid item*; element element yang berada (1 level) di dalam grid container
* *Grid line*; Garis horizontal (kolom) atau vertical (baris) yang memisahkan grid menjadi beberapa bagian dan ditandai dengan angka
* *Grid cell*; Perpotongan/pertemuan antara baris dan kolom di dalam grid
* *Grid area*; Kumpulan lebih dari satu grid cell yang membentuk kotak
* *Grid track*; Ukuran/jarak antara 2 grid line, bisa horizontal (kolom) atau vertical (baris)
* *Grid gap*; jarak antar grid track/cell

Untuk lebih detail tentang grid bisa mempelajari di website berikut ini <https://www.w3schools.com/css/css_grid.asp>

**Praktikum Bagian – 18. *CSS Grid***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ketikkan kode program di bawah ini kemudian simpan file dengan nama grid.html |
| 2 | A screen shot of a computer  Description automatically generated    A screen shot of a computer  Description automatically generated |
| 5 | Jalankan program, amati hasilnya dan jelaskan **(Soal 31)** |

**Tugas *Jobsheet* 2:**

1. Buatlah tampilan dari hasil *flex-box* sebelumnya menjadi seperti ini

A screenshot of a computer

Description automatically generated

1. Tambahkan gambar dan aplikasikan permainan warna untuk backgroundnya supaya menjadi lebih menarik

**Jawaban Tugas**

Kode program :

tugas1.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Tugas 1</title>

    <link rel="stylesheet" href="styleTugas1.css">

</head>

<body>

    <div class="container-satu">

        <div class="kolom-utama">

            <h2>Judul Artikel</h2>

            <img src="https://picsum.photos/id/1060/600/250" alt="Gambar Artikel">

            <p>

                Lorem ipsum dolor sit amet consectetur adipisicing elit. Nulla magni facilis sequi, minima explicabo voluptate sit eligendi, fuga quaerat nam sint quos reiciendis officia obcaecati, saepe ipsa repellat fugiat! Quos?

                Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo magnam totam cum vero corrupti, obcaecati delectus ducimus expedita nam nostrum suscipit ab modi, pariatur dignissimos est animi, exercitationem eius. Sapiente?

            </p>

            <p>

                Lorem ipsum dolor sit, amet consectetur adipisicing elit. Necessitatibus vel ipsa alias hic ex? Tenetur, molestias veniam. Ducimus minima, quod praesentium expedita dolor assumenda, inventore aliquam provident magni quis nobis?

                Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quibusdam ab veniam molestias autem aspernatur eaque quasi praesentium laboriosam enim nam facere, id libero rem tenetur tempore. Soluta molestias aperiam beatae.

            </p>

            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores aliquam numquam corporis tempora? Quae error cumque corrupti voluptates perferendis reiciendis velit sapiente eligendi nam excepturi, ipsum repellendus eos consequatur alias.</p>

        </div>

        <div class="sidebar-satu">

            <h3>Sidebar Satu</h3>

            <ul>

                <li><a href="">Link 1</a></li>

                <li><a href="">Link 2</a></li>

                <li><a href="">Link 3</a></li>

                <li><a href="">Link 4</a></li>

                <li><a href="">Link 5</a></li>

            </ul>

        </div>

        <div class="sidebar-dua">

            <h3>Sidebar Dua</h3>

            <p>

                Lorem ipsum dolor sit, amet consectetur adipisicing elit. Error quia quidem dolores commodi ipsa, tenetur libero, provident sunt id perferendis veritatis aliquid, neque nulla. Eius ea facilis praesentium laboriosam quod.

            </p>

        </div>

    </div>

</body>

</html>

styleTugas1.css:

html, body {

    margin: 0;

    padding: 0;

}

body {

    font-family: Arial, Helvetica, sans-serif;

    background-image: linear-gradient(to top, #cfd9df 0%, #e2ebf0 100%);

    line-height: 1.65;

    padding-bottom: 50px;

}

.container-satu {

    width: 960px;

    margin: 50px auto;

    background-color: #fff;

    padding: 20px;

    box-sizing: border-box;

    display: flex;

    gap: 25px;

    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);

    border-radius: 8px;

}

.kolom-utama {

    order: 2;

    flex: 3;

}

.sidebar-satu {

    order: 1;

    flex: 1;

}

.sidebar-dua {

    order: 3;

    flex: 1;

}

.kolom-utama h2, .sidebar-satu h3, .sidebar-dua h3 {

    color: #0d47a1;

    border-bottom: 2px solid #f0f0f0;

    padding-bottom: 10px;

}

.kolom-utama img {

    max-width: 100%;

    height: auto;

    border-radius: 5px;

    margin-bottom: 15px;

}

.sidebar-satu ul {

    list-style: none;

    padding: 0;

}

.sidebar-satu li a {

    text-decoration: none;

    color: #1565c0;

    display: block;

    padding: 8px 0;

    border-bottom: 1px solid #eee;

    transition: background-color 0.2s ease;

}

.sidebar-satu li a:hover {

    color: #0d47a1;

    background-color: #f7f9fc;

}

hasil :



**Kata kunci**; gunakan *property flex* dan *order* pada masing-masing *selector* itemnya. Kemudian *capture* kode program dan hasilnya

Referensi:

1. Jason Beaird, The principles of Beautiful Web Design
2. Rian Ariona, Belajar HTML dan CSS (Tutorial Fundamental dalam mempelajari HTML dan CSS)
3. Adi Hadisaputra, HTML dan CSS Fundamental dari Akar menuju Daun John Duckett,HTML dan CSS design and build websites
4. <https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements>
5. <https://css-tricks.com/almanac/properties/d/display>
6. <http://www.w3.org/TR/CSS2/box.html>)
7. <http://www.w3schools.com/css/css_boxmodel.asp>)