**JOBSHEET 2**

**PEMROGRAMAN WEB**

**Nama: Rasendriya Dafa Setiadi**

**Kelas : TI-2E**

**NIM : 2341720125**

**Tujuan**

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat situs 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 dilakukan secara bertahap sesuai dengan langkah-langkah praktikum yang telah diberikan.

**Apa itu CSS?**

CSS adalah singkatan dari "*Cascading Style Sheets*". Sesuai dengan namanya, CSS memiliki properti "*style sheet language*" yang berarti bahasa pemrograman yang digunakan untuk desain web. CSS adalah bahasa pemrograman yang digunakan untuk mendesain halaman situs web. Dalam mendesain halaman website, CSS menggunakan marker, yaitu **id** dan **class**. CSS dapat mengubah font, ukuran font, warna dan format font, mengatur ukuran tata letak, lebar, tinggi dan elemen warna, mengubah tampilan formulir, membuat halaman situs web responsif, dan banyak lagi.

Untuk mendesain font dapat dilakukan dengan mendefinisikan font, untuk mengatur warna dapat menggunakan warna, margin digunakan untuk mengatur jarak di luar elemen tertentu. Atur ukuran font menggunakan "*ukuran font*". Jenis *font* menggunakan "*font-family*" dan banyak lainnya.

# Cara Menggunakan CSS

File css disimpan dengan **ekstensi .css**. kemudian diimpor atau ditautkan ke dalam file HTML atau PHP yang ingin kita desain dengan CSS menggunakan sintaks berikut:

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

Tag top digunakan untuk menghubungkan file HTML dengan file CSS. Sintaks ditempatkan pada file html. Atribut rel dan type dalam tag link digunakan untuk mendefinisikan bahwa apa yang disebut atau ditautkan adalah stylesheet atau file CSS, kemudian atribut href digunakan untuk menempatkan lokasi file CSS. Pada contoh di atas, file style.css terletak di folder atau direktori dengan file html. jika file CSS terletak di luar folder, itu dapat dikaitkan dengan cara:

href=".. /style.css"

Jika file css terletak di folder, katakanlah nama foldernya adalah "assets", maka untuk mengaitkannya dengan cara:

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

# Praktikum Bagian 1. Menghubungkan HTML dengan CSS

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat folder baru bernama minggu2. Buat file di dalamnya dan beri nama sebagai  index.html. |
| 2 | Ketik ke dalam file index.html kode di bawah ini. |
| 3 |  |
| 4 | Buat satu file baru di folder minggu2, beri nama style.css. Ketik kode di bawah ini di dalam file style.css. |
| 5 |  |
| 6 | Simpan file, lalu buka browser dan jalankan localhost/minggu2 |
| 7 | Dalam struktur halaman web, index.html akan selalu menjadi halaman utama web. Jadi jika ada index.html di direktori maka itu akan selalu dipanggil dan ditampilkan di browser. Itulah mengapa pada langkah 6 hanya diketik halaman localhost/minggu2 tanpa nama file. |
| 8 | Dalam kode di dalam index.html ada tag <link> di kepala di mana atribut href  mengacu pada file style.css. |
| 9 | style.css berisi kode untuk mengatur tampilan di dalam halaman yang mengacu pada file style.css, dalam praktikum ini index.html. Sehingga tampilan di dalam index.html akan sesuai dengan properti yang ditetapkan di style.css. |

**Bagian yang menggunakan tag <div>**

Tag <div> mendefinisikan bagian dalam dokumen HTML. Elemen <div> sering digunakan sebagai wadah untuk elemen HTML lainnya untuk menambahkan *gaya* dengan CSS atau untuk menampilkan tugas tertentu menggunakan JavaScript.

# Praktikum Bagian 2: Penggunaan div

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru bernama div.html di dalam folder minggu2. Ketik kode pada langkah 2 di dalam div.html |

|  |  |
| --- | --- |
| 2 |  |
| 3 | Simpan file, lalu buka browser Anda dan jalankan localhost/minggu2/div.html |
| 4 | Apa yang Anda pahami dari menggunakan div pada file? Tuliskan pemahaman Anda. (Pertanyaan No. 1)    Penggunaan duv befungsi sebagai pengelompokan elemen yang berguna untuk mempermudah styling dan penataan CSS pada website. |

**Mengenal class dan id di HTML** Class dan id digunakan sebagai penanda dalam html, yang dimaksud dengan penanda di sini adalah elemen html dapat ditandai dengan class atau id. Elemen dalam HTML ditandai sehingga dapat dimanipulasi menggunakan CSS atau JavaScript. Bayangan sederhananya adalah jika Anda memiliki lima kotak, semua kotak yang Anda miliki berwarna biru, maka Anda ingin mengubah warna kotak ketiga, nah di sinilah penggunaan class dan id masuk, untuk memberikan tanda atau nama pada

kotak Anda agar dapat diubah, dan kotak lainnya tidak akan berubah.

Perbedaan antara class dan id adalah bahwa class dipanggil pada css atau javascript menggunakan "periode".", dan id dipanggil pada CSS atau javascript dengan tagar "#". Kelebihan class adalah class dapat diberikan ke banyak elemen HTML dan dapat dipanggil sekaligus, sedangkan ID hanya dapat bekerja pada satu tag, artinya satu nama ID hanya dapat diberikan ke satu elemen.

# Praktis Bagian 3: Mengenal kelas dan id dalam HTML

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Lengkapi kode di dalam index.html dengan kode berikut. |
| 2 |  |
| 3 | Lengkapi kode di dalam file style.css sehingga menjadi kode di langkah 4 |

|  |  |
| --- | --- |
| 4 |  |
| 5 | Simpan kedua file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 6 | Apa yang Anda pahami dari penggunaan class dan id di index.html? Tuliskan pemahaman Anda. (Pertanyaan No. 2)    Fungsi class aalah sebagai penanda untuk setiap elemen agar setiap elemen dapat dimanipulasi dengan CSS dan javascript secara rapi dan terstruuktur. |

**Cara Menulis CSS**

CSS menggunakan selector (id dan class) untuk menentukan elemen yang akan dimodifikasi oleh CSS, jika diibaratkan dengan HTML sebagai pilar pada bangunan rumah, maka CSS berfungsi sebagai cat dan dekorasi pada bangunan rumah. Ada tiga teknik metode penulisan CSS, yaitu:

* *Inline CSS Style*

Ini adalah CSS yang dibuat dalam tag HTML yang hanya berlaku untuk dokumen yang dilampirkannya. Biasanya teknik ini digunakan untuk pemformatan khusus elemen HTML dan tidak digunakan untuk memformat seluruh elemen dalam dokumen web.

Contoh:

<h1 style="color:biru; margin-left:30px;" >Ini adalah judul.</h1>

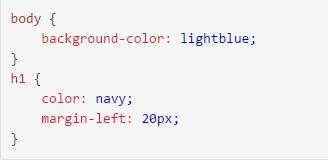
* *Internal CSS Style*

*Internal CSS Style* untuk halaman web hanya berlaku untuk halaman web tersebut. *Internal CSS Style* didefinisikan di bagian HEAD halaman HTML, dalam tag <style> sebagai berikut:



Gambar 1. Internal CSS

* *Eksternal CSS Style*
* Ini sangat ideal untuk digunakan di web dengan banyak halaman. Dengan menggunakan External Style Sheet, tampilan seluruh konten website dapat diubah hanya dengan mengubah satu file. *Eksternal CSS Style* tidak boleh berisi tag html, dan disimpan dalam file dengan ekstensi \*.css Contoh:



Gambar 2. CSS eksternal

Cara memanggil *External Style Sheet* di halaman web adalah dengan menggunakan tag <link> dengan atribut rel ditempatkan di bagian *head*. Contoh:



Gambar 3. Menggunakan file CSS eksternal ke dalam HTML

**Praktikum Bagian 4: Mengubah *Latar Belakang* Halaman Web dengan CSS**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Tambahkan cuplikan kode pada langkah 2 di bawah ini ke file style.css |
| 2 |  |
| 3 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 4 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 3)    Background berubah menjadi warna biru muda karena telah terjadi manipulasi class body menggunakan CSS. |

**Praktikum Bagian 5: Menggunakan *gambar* untuk *latar belakang*** Properti *background-image* menetapkan satu atau beberapa gambar latar belakang ke elemen. Secara default, gambar latar belakang ditempatkan di sudut kiri atas elemen, dan berulang secara vertikal dan

horizontal.

Tips: Elemen background adalah ukuran total elemen, termasuk *padding* dan *batas* (tetapi bukan margin). Tips: Selalu atur warna background yang akan digunakan jika gambar tidak tersedia.

Langkah Praktikum:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Di file style.css, ubah nilai dari properti background menjadi bunga2.jpg (atau nama file gambar sebagai background) seperti pada cuplikan kode pada langkah 2. |
| 2 |  |
| 3 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 4 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 4)    Background bisa berupa warna polos atau menggunakan gambar dengan ditambahkan url gambar yang ingin ditambahkan. |

|  |  |
| --- | --- |
| 5 | Tambahkan file gambar bernama field1.jpg dan tree1.jpg (atau nama file gambar sebagai latar belakang dan objek) ke folder img. |
| 6 | Ketik kode berikut di style.css. |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Kode Anda berjalan dengan baik jika terlihat seperti di bawah ini: |
| 10 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 5)    Gambar 2 akan melakukan perulangan karena perintah repeat pada background-repeat sedangkan gambar 1 tidak. |

# Praktikum Bagian 6: Margin dan Padding pada CSS

**Mengenal Margin di CSS**

Margin adalah sisi luar dari suatu elemen. Misalnya, Anda ingin mengatur jarak antar elemen. Anda dapat menggunakan *sintaks margin* untuk mengaturnya. Ada beberapa sisi luar margin yaitu,

* Margin atas ditulis di CSS dengan 'margin-top',
* margin bawah atau spasi luar bagian bawah ditulis dalam CSS dengan 'margin-bottom',
* 'margin-left' sebagai spasi luar di sebelah kiri elemen, dan
* 'margin-right' adalah sisi luar di sisi kanan

Tetapi jika Anda hanya menggunakan sintaks 'margin', itu akan secara otomatis mengatur spasi atas, bawah, kiri dan kanan elemen.

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi backgroundCSS.html |
| 2 | Buat file baru di dalam folder minggu2 bernama index.html. Ini berarti Anda memiliki file  index.html baru . |
| 3 | Ketik kode berikut di index.html |

|  |  |
| --- | --- |
| 4 |  |
| 5 | Buat file baru bernama styleMargin.css di folder minggu2. |
| 6 | Ketik kode berikut ke styleMargin.css |
| 7 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 8 | Tuliskan apa yang Anda amati dari kode di atas. (Pertanyaan No. 6)    Terdapat 2 kotak dengan ukuran yang berbeda, warna yang berbeda, dan pengaturan jarak yang berbeda. Margin untuk memberi jarak border disetiap sisinya sedangkan margin-left hanya memberi jarak pada sisi kiri saja. |
| 9 | Ubah nilai margin di pengaturan .box menjadi margin: 300px;. Amati perbedaannya. |
| 10 | Tuliskan apa yang Anda amati dari kode di atas. (Pertanyaan No. 7)    Setelah diubah maka akan menghasilkan output seperti gambar, terjaadi karena box mengambil jarak sebanyak 300 pixel disetiap sisinya. |

* Mengenal *Padding* di CSS

Padding adalah sisi dalam suatu elemen. Kita dapat menggunakan padding sintaks untuk mengatur spasi pada sisi dalam elemen yang kita tentukan. Sama seperti margin memiliki sisi. seperti atas, kiri, kanan, bawah. Jenis padding adalah padding atas yang ditulis dalam CSS dengan 'padding-top' yang berarti mengatur sisi dalam bagian atas elemen, pad bawah atau spasi di bagian bawah ditulis dalam CSS dengan 'padding-bottom', 'padding-left' sebagai spasi dalam di sisi kiri elemen, dan 'padding-right' adalah sisi luar di sebelah kanan. Jika Anda hanya menggunakan sintaks 'padding', itu akan secara otomatis mengatur spasi atas, bawah, kiri dan kanan dari elemen dalam.

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |

|  |  |
| --- | --- |
| 1 | Ganti nama index.html menjadi marginCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file  index.html baru. |
| 3 | Ketik kode berikut untuk index.html |
| 4 |  |
| 5 | Buat file baru bernama stylePadding.css di folder minggu2 |
| 6 | Ketik kode berikut untuk stylePadding.css |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |

|  |  |
| --- | --- |
| 9 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 8)    Adalah contoh penerapan padding pada 2 box. |
| 10 | Ubah nilai padding dalam pengaturan .box menjadi padding=200px; |
| 11 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 9)    Box pertama menjadi besar karena padding pada box pertama bertambah sebanyk 10 kali lipat. |

# Praktikum Bagian 7: Pengaturan Font di CSS

Beberapa sintaks CSS yang digunakan untuk mengatur font:

* + *font-size* digunakan untuk mengatur ukuran font
  + *font-weight* digunakan untuk menyesuaikan ketebalan font
  + *font-family* untuk mengubah jenis font
  + *font-style* digunakan untuk mengubah gaya font.
  + *color* digunakan untuk mengubah warna font

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi paddingCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file  index.html baru . |
| 3 | Ketik kode berikut pada index.html |
| 4 |  |
| 5 | Buat file baru bernama styleFont.css di folder minggu2 |
| 6 | Ketik kode berikut ke styleFont.css |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini (Pertanyaan No. 10)    Berisi styling warna text dengan berbagai gaya penulisan. |

**Praktis Bagian 8: Menyiapkan *Hyperlink* dengan CSS** *Hyperlink* adalah tautan yang dibuat untuk mengalihkan halaman saat diklik. *Hyperlink* atau tautan dibuat menggunakan tag dan diakhiri dengan tag dalam HTML. Ada 4 status yang dimiliki oleh hyperlink

html dan dapat dimanipulasi menggunakan css. Yaitu:

* + *link*. adalah tautan aktif biasa.
  + *visited*. adalah status tautan yang telah dikunjungi.
  + *hover*. adalah status tautan saat kursor mouse ditempatkan di atasnya.
  + *active*. adalah status tautan atau *hyperlink* saat diklik.

*Sintaksnya* adalah sebagai berikut:

* + *a:link* = untuk tautan biasa
  + *a:visited* = adalah status tautan yang telah dikunjungi.
  + *a:hover* = adalah status tautan saat kursor mouse ditempatkan di atasnya.
  + *a:active* = adalah status tautan atau hyperlink ketika telah diklik.

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi fontCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file  index.html baru. |
| 3 | Ketik kode berikut untuk index.html |
| 4 |  |
| 5 | Buat file baru bernama styleLink.css di folder minggu2 |
| 6 | Ketik kode berikut untuk styleLink.css |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini (Pertanyaan No. 11)    Pada gambar jika kita mengarahkan kursor pada link, link akan berubah warna menjadi merah. |
| 10 | Tambahkan kode ke styleLink.css untuk mengubah warna tautan menjadi greenyellow  setelah tautan dikunjungi. |
| 11 | Tulis kode yang Anda tambahkan di sini (Pertanyaan No. 12)    Ketika link tersebut sudah dikunjungi maka warna teks akan berganti warna menjadi greenyellow. |

# Praktikum Bagian 9: Memformat Teks dengan CSS

Pengaturan pemformatan teks di CSS adalah sebagai berikut:

1. *color*: Untuk mengatur warna teks, nilainya dapat diisi sebagai kode warna atau warna
2. *text-align:* Untuk mengatur posisi perataan pada teks atau teks datar, nilai yang dapat diisi termasuk center untuk membuat teks sejajar di tengah, left untuk membuat teks sejajar kiri, right untuk membuat teks sejajar kanan dan justify untuk membuat teks sejajar kanan dan kiri rata.
3. *text-decoration:* untuk mengatur dekorasi teks, nilainya none untuk membuat teks tidak memiliki dekorasi, overline untuk membuat teks memiliki baris di bagian atas teks, line-through untuk membuat garis coretan pada teks, dan underline untuk membuat baris di bagian bawah teks.
4. *text-transform*: Untuk mengatur huruf kapital dalam teks, nilai yang dapat digunakan meliputi uppercase untuk membuat teks huruf besar, lowercase untuk membuat teks huruf kecil, dan capitalize untuk membuat huruf awal dari setiap kata huruf besar.
5. *text-indent:* Untuk mengatur spasi paragraf dalam teks, nilai yang dapat digunakan adalah berupa nilai piksel dan lainnya sesuai kebutuhan.
6. *letter-spacing*: untuk menyesuaikan jarak antar karakter dalam teks, nilai yang diisi dalam bentuk nilai piksel dan lain-lain.
7. *word-spacing:* Untuk mengatur spasi antar kata dalam teks, nilai yang diisi juga merupakan nilai piksel.
8. *line-height*: untuk mengatur jarak antar baris dalam nilai teks yang diisi dalam bentuk nilai.
9. *text-shadow*: Untuk mengatur efek bayangan pada teks, nilai yang diisi pertama mengisi nilai untuk jarak kiri dan kanan, dan yang kedua mengisi spasi atas dan bawah dan yang ketiga mengisi warna. Untuk contoh penulisannya adalah 2px 5px blue.
10. *vertical-align*: untuk mengatur perataan dalam bentuk vertikal dalam nilai teks yang digunakan adalah kiri untuk membuat teks sejajar kiri, kanan ke atas dan tengah ke tengah.

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi linkCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file  index.html baru . |
| 3 | Ketik kode berikut untuk index.html. |
| 4 |  |
| 5 | Buat file baru bernama styleText.css di folder minggu2 |
| 6 | Ketik kode pada langkah 7 di bawah ini ke styleText.css |

|  |  |
| --- | --- |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Catat pengamatan Anda dan tuliskan jawaban Anda di bawah ini (Pertanyaan No. 13)    Berisi teks judul dengna warna biru dengan 1 paragraf. |
| 10 | Tambahkan kode ke styleText.css untuk mengubah posisi teks ke tengah dengan menambahkan text-align: center; dan menambahkan dekorasi dalam bentuk garis bawah. |
| 11 | Tulis kode yang Anda tambahkan di sini (Pertanyaan No. 14) |
| 12 | Tambahkan kode ke styleText.css untuk menambahkan garis bawah ke teks. |
| 13 | Tulis kode yang Anda tambahkan di sini (Pertanyaan No. 15) |
| 14 | Tambahkan kode ke styleText.css untuk memberikan spasi antar karakter dalam paragraf yang ada menjadi 5px dengan letter-spacing. |
| 15 | Tulis kode yang Anda tambahkan di sini (Pertanyaan No. 16) |

# Praktikum Bagian 10: Mengenal Posisi CSS

Posisi di CSS digunakan untuk mengatur posisi elemen HTML. Properti posisi CSS ini digunakan untuk menentukan posisi elemen HTML sesuai keinginan.

Secara umum, untuk membuat posisi atau mengatur posisi suatu elemen kita harus menggunakan properti CSS lainnya seperti setting top, left, bottom, right untuk mengatur posisi suatu elemen, namun properti tersebut tidak akan berfungsi jika posisi belum diatur terlebih dahulu, hal ini dikarenakan properti lain bergantung pada posisi yang diatur.

Beberapa properti CSS yang dapat digunakan untuk memposisikan elemen HTML adalah:

* + *Static*: Posisi statis digunakan untuk mengatur elemen ke statis secara default. Elemen akan mengikuti posisi normal secara default, elemen tidak terpengaruh oleh properti seperti atas, bawah, kiri dan kanan.
  + *Relative*: Elemen HTML yang menggunakan posisi relatif akan ditempatkan pada posisi normal. Mengatur properti atas, kanan, bawah, dan kiri dari elemen yang diposisikan "relatif" akan membuatnya jauh dari posisi normalnya. Konten lain tidak akan disesuaikan agar sesuai dengan celah yang ditinggalkan oleh elemen tersebut.
  + *Fixes:* Elemen HTML yang diatur ke posisi tetap akan memiliki properti tetap tanpa perubahan apa pun meskipun halaman situs web digulir. Pengaturan kiri, bawah, atas, dan kanan berlaku untuk posisi tetap.
  + *Absolute:* Elemen HTML yang menggunakan posisi absolut akan diposisikan relatif terhadap elemen lain yang mendahuluinya paling dekat dengannya, bukan relatif terhadap layar secara normal.
  + *Sticky*: Elemen dengan posisi: sticky; diposisikan berdasarkan posisi scroll pengguna. Elemen lengket bergantian antara relatif dan tetap, tergantung pada posisi gulir. Dan diposisikan relatif

sampai posisi offset tertentu bertemu di viewport - kemudian "stick" di tempatnya (seperti position: fixed).

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi textCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file index.html  baru. |
| 3 | Ketik kode berikut untuk index.html |
| 4 |  |

|  |  |
| --- | --- |
|  |  |
| 5 | Buat file baru bernama stylePosition.css di folder minggu2 |
| 6 | Ketik kode berikut untuk stylePosition.css |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini (Pertanyaan No. 17)    Percobaan pada gambar merupakan beberapa contoh elemen diposisikan dengan berbagai cara. |

**Praktikum Bagian 11: Menggunakan Float** Teknik Float di bagian desain web adalah kebutuhan yang paling dibutuhkan. Properti float bisa digunakan untuk memposisikan dan memformat konten. Salah satu contoh paling umum dari penggunaan float adalah ketika kita ingin membuat postingan situs web yang terletak di sisi teks konten. Properti float

dapat memiliki salah satu nilai berikut:

* + left - Elemen mengambang di sebelah kiri container
  + right- Elemen mengambang di sebelah kanan container
  + none - Elemen tidak mengambang (akan ditampilkan tepat di mana teks muncul) – merupakan setting default
  + *inherit* - Elemen ini mewarisi nilai float dari induknya

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

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ganti nama index.html menjadi positionCSS.html |
| 2 | Buat file baru di folder minggu2 bernama index.html. Ini berarti Anda memiliki file index.html  baru. |
| 3 | Ketik kode berikut untuk index.html |
| 4 |  |
| 5 | Buat file baru bernama styleFloat.css di folder minggu2 |
| 6 | Ketik kode berikut untuk styleFloat.css |

|  |  |
| --- | --- |
| 7 |  |
| 8 | Simpan file, lalu buka browser dan jalankan /refresh localhost/minggu2 |
| 9 | Catat pengamatan Anda dan tulis jawaban Anda di bawah ini. (Pertanyaan No. 18)    Gambar telah di setting float left yang berfungsi untuk meletakkan gambar ke sisi kiri website. |

**Apa itu Tata Letak CSS?**

Tata letak CSS (CSS Layouting) adalah teknik untuk mengatur tata letak halaman web menggunakan kode CSS. Tujuan dari CSS Layouting adalah untuk membuat halaman terlihat rapi dan menarik seperti yang diharapkan oleh pembuatnya. CSS Layouting terdiri dari beberapa bagian yaitu; dimension, overflow, box model, float dan position.

**Tampilan CSS */ CSS Display***

Tag HTML digunakan untuk memberikan 'makna' pada sepotong konten (misalnya p untuk paragraf, h1 untuk *heading* dan seterusnya). Tag <div> dan <span> tidak memiliki arti, keduanya digunakan untuk mengelompokkan tag HTML dan memberikan Keterangan kepada mereka.

**Praktikum Bagian 12. *Tata Letak CSS***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru di folder minggu2 dan beri nama csslayouting.html |
| 2 | Ketik kode berikut untuk csslayouting.html |
| 3 |  |
| 4 | Simpan file, lalu buka browser dan jalankan /refresh  localhost/minggu2/csslayouting.html |

|  |  |
| --- | --- |
| 5 | Kemudian buat <div> pada halaman html yang sama seperti pada kode di bawah ini |
| 6 |  |
| 7 | Amati apakah hasil dari kedua program tersebut sama atau berbeda, jelaskan (Pertanyaan No. 19)  sama saja akrena masih hanya diberikan layouting div tanpa styling. |
| 8 | Tambahkan *style* ke kelas ***navigasi*** dan kelas ***main*** seperti pada kode di bawah ini |
| 9 |  |
| 10 | Jalankan kode dan screen shoot hasilnya. Jelaskan apa yang terjadi (Pertanyaan No. 20)    Yang terjadi adalah warna background elemen navigasi dan main berbeda karena sudah diberikan warna yang berbeda dalam styling css. |

# Value of Display

* Inline Display

HTML secara *default* tidak menambahkan baris baru saat dibuat. Ciri-ciri tampilan *inline* adalah:

1. Lebar dan tinggi elemen sesuai dengan konten di dalamnya
2. Tidak dapat menyesuaikan tinggi dan lebar elemen sebaris
3. *Margin* dan *padding* hanya memengaruhi elemen secara horizontal, bukan vertikal

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

# Praktikum Bagian 13. Tampilan Sebaris

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Hapus *style* di **langkah 9 Praktikum Bagian 12**. Tambahkan 5 link ke kelas ***navigasi*** seperti yang ditunjukkan pada kode di bawah ini. |

|  |  |
| --- | --- |
| 2 | Tangkapan layar komputer  Deskripsi dibuat secara otomatis |
| 3 | Jalankan kode dan screen shoot hasilnya. Jelaskan apa yang terjadi (Pertanyaan No. 21)    Web akan kembali seperti langkah ke 7 karena styling css yang sebelumnya ada dihapus. |

* *Display inline-block*

Jika dalam elemen *inline* kita tidak dapat mengatur tinggi dan lebar suatu elemen, maka kita dapat mengaturnya menggunakan *inline-block*. Elemen tidak memiliki properti *secara default***.**

|  |  |
| --- | --- |
| **Praktikum Bagian 14. Tampilkan *blok sebaris*** | |
| **Langkah** | **Keterangan** |
| 1 | Tambahkan *weight*, *height* dan *display* ke *style* dalam elemen **a** seperti yang ditunjukkan pada kode di bawah ini |
| 2 | Tangkapan layar komputer  Deskripsi dibuat secara otomatis |
| 3 | Jalankan kode dan screen shoot hasilnya. Jelaskan apa yang terjadi. (Pertanyaan No. 22)    Yang terjadi pada gamba adalah menjadikan background setiap elemen dengan tag “a” berwarna pink,dengan lebar dan panjang 200px dengan display inline-block.la |

* *Display Block*

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Praktikum Bagian 15. Display Block** | | |  |
|  | **Langkah** | **Keterangan** | |
| 1 | Style elemen h1, h2 dan p dengan *background color* **light-green** seperti yang ditunjukkan pada kode di bawah ini | |
| 2 | Tangkapan layar program komputer  Deskripsi dibuat secara otomatis | |

|  |  |
| --- | --- |
| 3 | Jalankan program dan coba ubah ukuran halaman web. *Screen shoot* hasilnya, amati dan jelaskan apa yang terjadi. (Pertanyaan No. 23)    Yang terjadi pada gambar adalah setiap elemen dengan tag h1,h2,dan p diberikan background dengan warna lightgreen. |
| 4 | Kemudian tambahkan gaya ke kelas **main** seperti pada kode di bawah ini |
| 5 | Tangkapan layar program komputer  Deskripsi dibuat secara otomatis |
| 6 | Jalankan program. *Screen shoot* hasilnya, amati dan jelaskan apa yang terjadi. (Pertanyaan No. 24)    Fungsi .main adalah untuk memanipulasi elemen dengan tag main beserta tag yang ada didalam main. Untuk main h2 untuk mengatur teks h2 yang ada pada tag main saja. |

* *Display none*

Nilai tampilan terakhir adalah *tidak ada*, *tidak ada yang* dapat digunakan untuk menghapus elemen

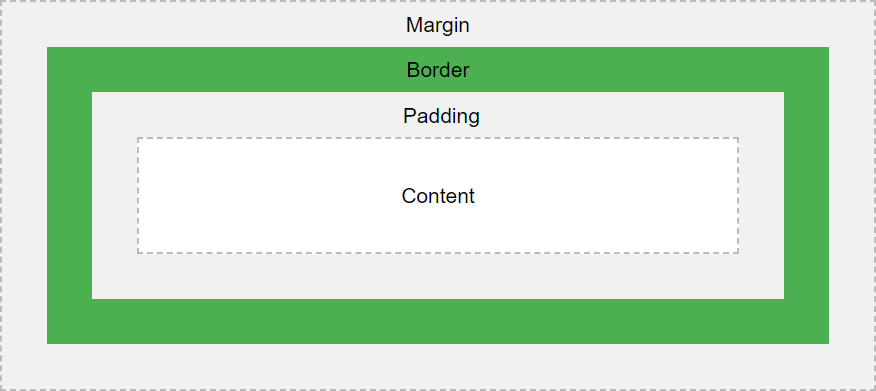
**Dimensi dan *Overflow* pada CSS**

Dimensi memiliki dua *properti* dalam CSS, yaitu *width* untuk lebar dan *height* untuk tinggi. Satuan dimensi beragam termasuk **px, %, in, cm, mm, pc**.

*Overflow* adalah *properti* CSS yang digunakan untuk mengatur perilaku elemen yang tidak cukup pada *induk.* Ada empat nilai property *overflow*:

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

***Box model* pada CSS**

Setiap elemen di halaman *situs web* ada di dalam kotak. Kita dapat mengatur ukuran dan posisi kotak. Kita bisa memberikan warna/gambar sebagai *background* kotak. *Box model* di CSS mendefinisikan 'box' yang dihasilkan oleh suatu elemen, dan kemudian menampilkannya sesuai dengan format visualnya. *Box model* CSS terdiri dari 4 komponen yaitu; margin, border, padding, dan content seperti yang ditunjukkan pada gambar di bawah ini.

Gambar 4. Komponen Model Kotak

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

Tabel 1. Cara menyiapkan properti box model

|  |  |  |
| --- | --- | --- |
| ***Margin*** | ***Padding*** | ***Batas*** |
| *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 margin. Kiri dan kanan atau atas dengan bawah. Secara teori, jika itu terjadi, nilai terbesar akan diambil.
  + *Negative* margin akan membuat kotak bergerak ke arah yang berlawanan. Ini dapat digunakan jika kita ingin menyembunyikan elemen.
  + *Auto* margin adalah nilai yang dapat kita tetapkan ke margin khusus untuk margin kiri dan margin kanan. Otomatis ini akan membuat elemen di tengah halaman web browser
  + *Shorthand* margin adalah cara menyingkat penulisan margin.

**Praktikum Bagian 16. *Model Kotak: Margin***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat 2 file seperti yang ditunjukkan pada gambar di bawah ini. File pertama dengan nama margin.html dan file kedua marginstyle.css |
| 2 | Tangkapan layar program komputer  Deskripsi dibuat secara otomatis |
| 3 | Jalankan program, amati hasilnya dan jelaskan ( Pertanyaan No. 25)    Percobaan ini membuat 3 elemen div dengan styling yang berbeda disetiap elemennya. Dengan ukuran panjang dan lebar yang berbeda dan warna background yang berbeda dari yang terbesar dengan ukuran 200px sampai yang terkecil dengan ukuran 50px. |
| 4 | Selanjutnya adalah memberikan *margin*, di mana *margin adalah area transparan* di sekitar kotak. Tambahkan style ke **marginstyle.css** dengan margin-top 100px pada class **.satu**, simpan dan kemudian jalankan di browser web. Amati dan jelaskan pengamatan Anda (Pertanyaan No. 26)    Posisi kotak pertama berada lebih kebawah dibanding percobaan sebelumnya karena telah diberikan tambahan kode berupa margin-top: 100px yaitu memberikan jarak transparan pada kotak dengan batas atas web sebanyak 100px. |
| 5 | Kemudian tambahkan ***ukuran margin*** lain seperti pada kode berikut |

|  |  |
| --- | --- |
| 6 |  |
| 7 | Jalankan program. *Screen shoot* hasilnya, amati dan jelaskan apa yang terjadi. (Pertanyaan No. 27)    Perbedaan dari percoobaan sebelumnya adalah pada percobaan ini kotak satu diberikan margin yang berbeda pada setiap sisinya dari yang sebelumnya hanya diberikan margin pada sisi atas saja. |

* *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*

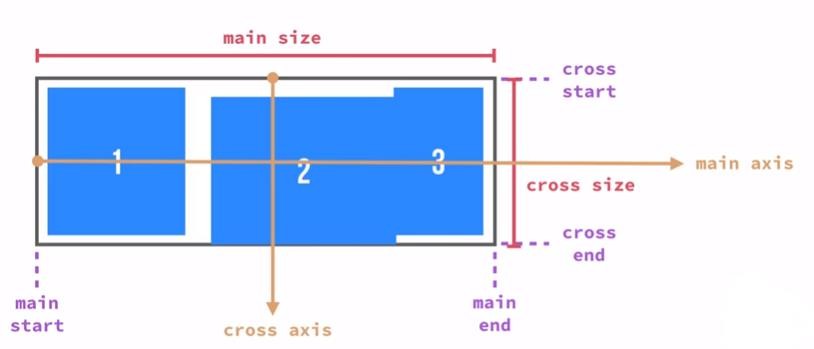
*Box sizing* adalah properti yang menerima nilai *padding* dan *border* pada suatu elemen termasuk sebagai nilai total dari *width* dan *height* suatu elemen.

*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.



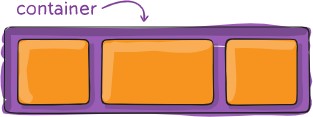
Gambar 5. Ketentuan pada Flex Box

(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

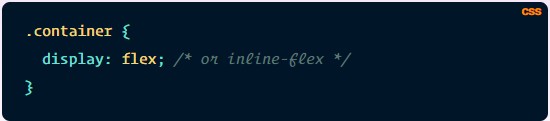
***Property* pada *container***

*Container* adalah pembungkus dari elemen



Gambar 6. Wadah

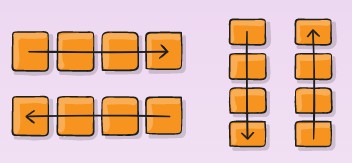
## Display



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

## Flex- direction

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

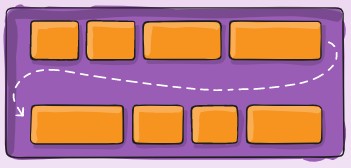


Gambar 7. Arah fleksibel

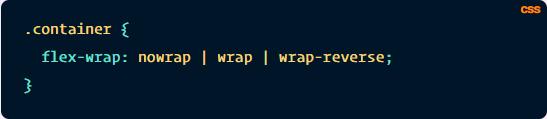


## Flex-wrap

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



Gambar 8. Flex warp



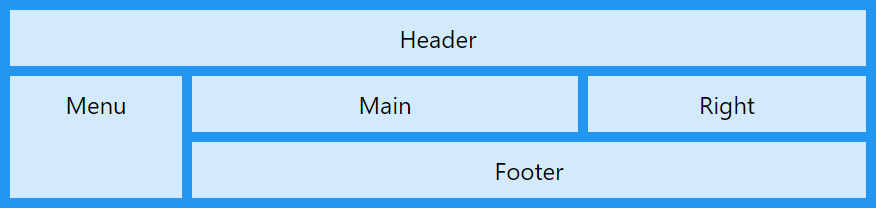
Untuk properti lainnya, Anda dapat mengunjungi situs web https://css-tricks.com/snippets/css/a-guide-to- flexbox/

**Praktikum Bagian 17. *Flex Box***

|  |  |  |
| --- | --- | --- |
|  | **Langkah** | **Keterangan** |
| 1 | Ketik kode di bawah ini lalu simpan file dengan nama flexbox.html |
| 2 |  |
| 3 | Dan untuk file style1.css sebagai berikut |
| 4 | Tangkapan layar komputer dari layar hitam  Deskripsi dibuat secara otomatis |
| 5 | Jalankan program, amati hasilnya dan jelaskan ( Pertanyaan No. 28) |
| 6 | Tambahkan *gaya* ke kelas **container-satu** pada file style1.css seperti pada kode di  bawah ini |
| 7 | Tangkapan layar program komputer  Deskripsi dibuat secara otomatis |

|  |  |
| --- | --- |
| 8 | Jalankan program. *Screen shoot* hasilnya, amati dan jelaskan apa yang terjadi. (Pertanyaan No. 29) |
| 9 | Tambahkan property **display** pada selector container-satu dengan value **flex**. Capture dan jelaskan hasilnya (Pertanyaan No. 30) |

**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)

Gambar 9 Modul Tata Letak Grid

*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 detail lebih lanjut tentang kisi, Anda dapat mempelajari di situs web berikut <https://www.w3schools.com/css/css_grid.asp>

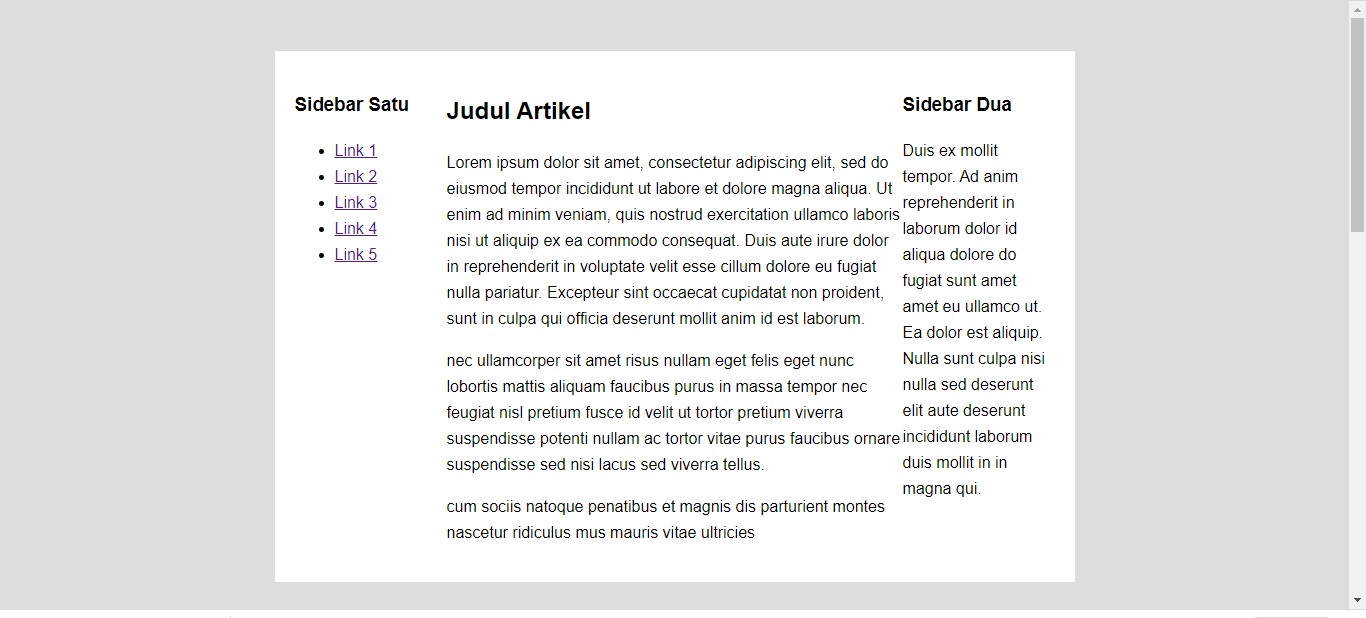
**Praktikum Bagian 18. *Grid CSS***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ketik kode di bawah ini lalu beri nama dengan grid.html |

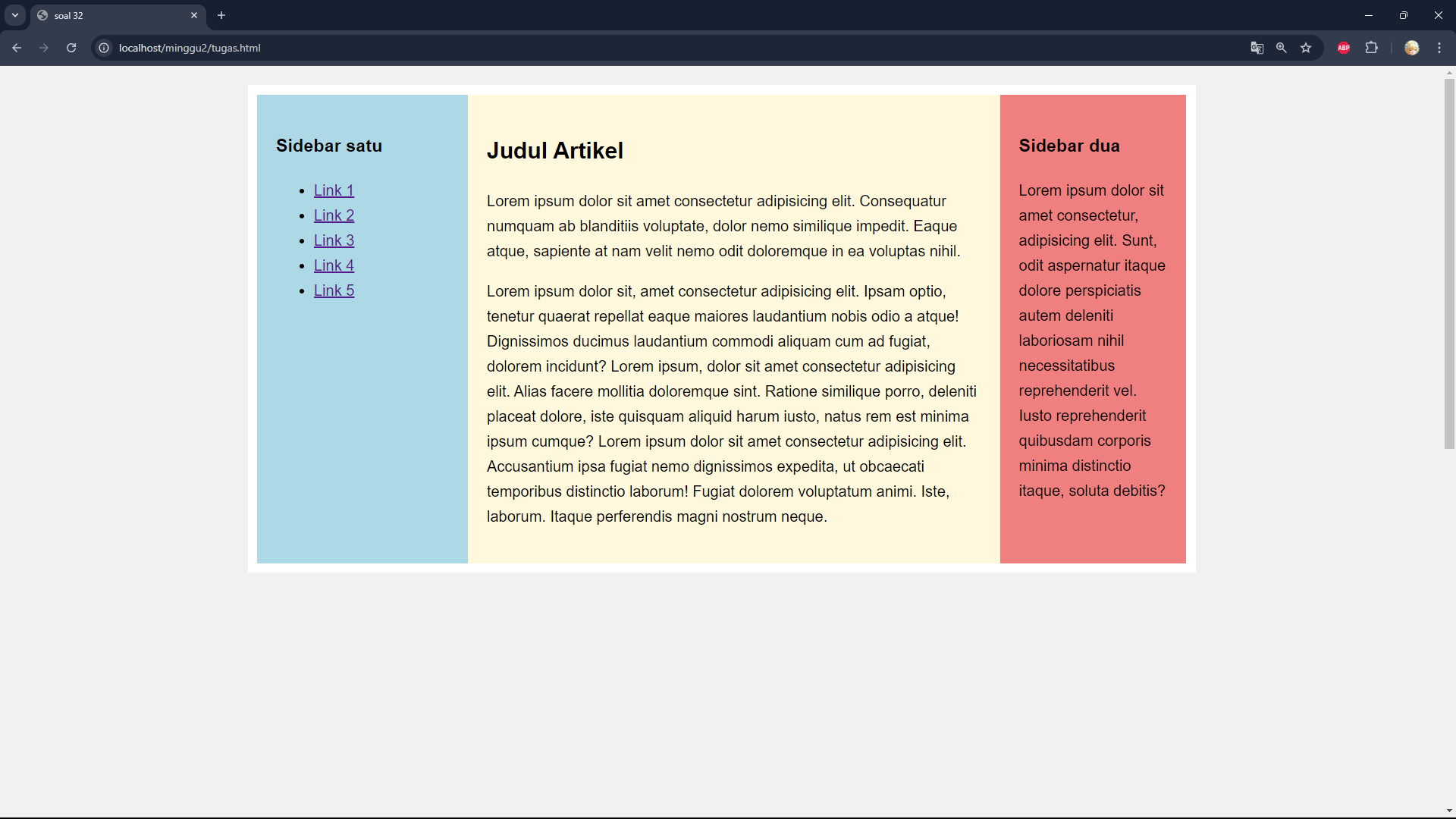
|  |  |
| --- | --- |
| 2 |  |
| 5 | Jalankan program, amati hasilnya dan jelaskan. (Pertanyaan No. 31) |

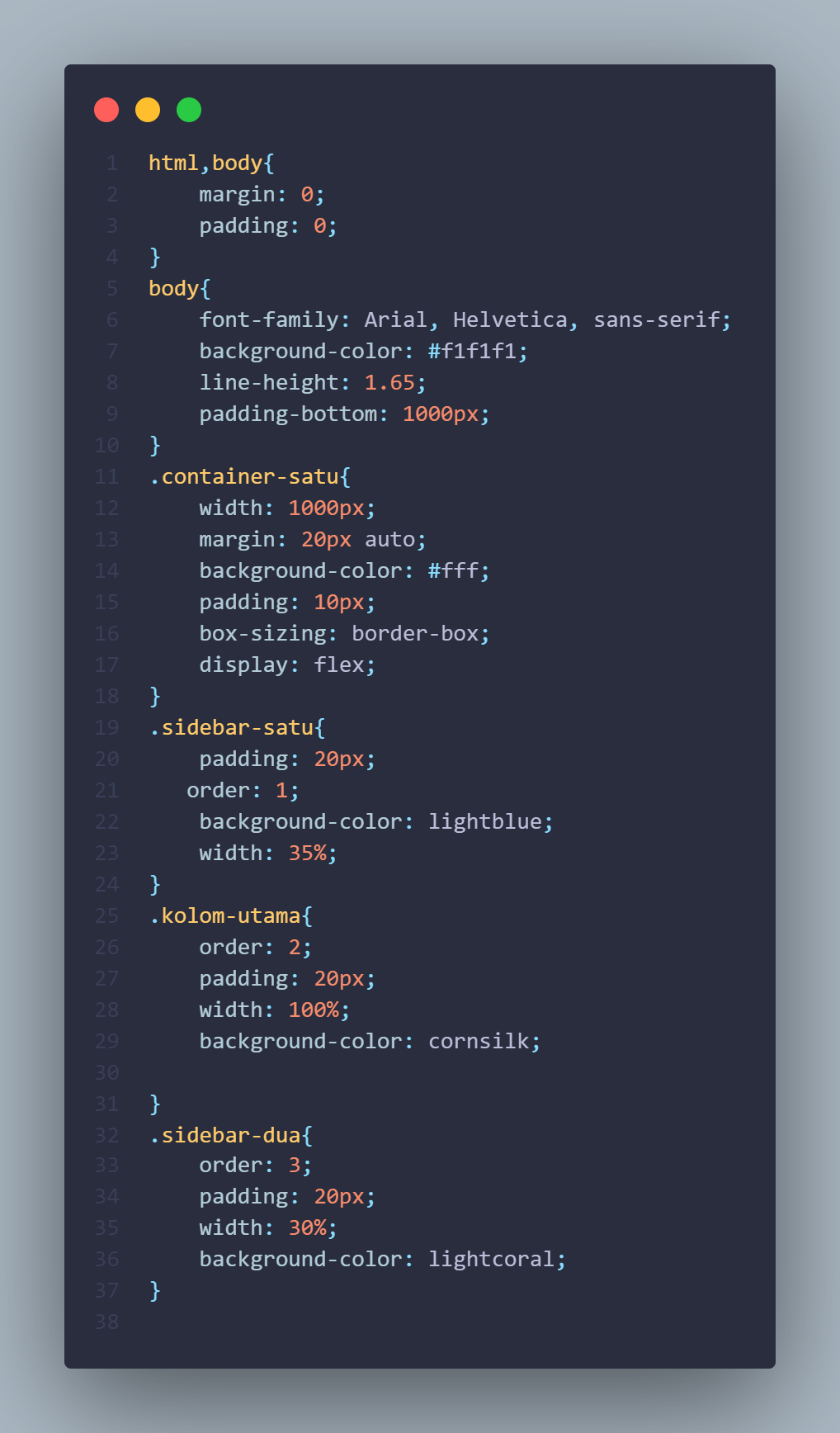
**LEMBAR TUGAS 2** (Pertanyaan No. 32)

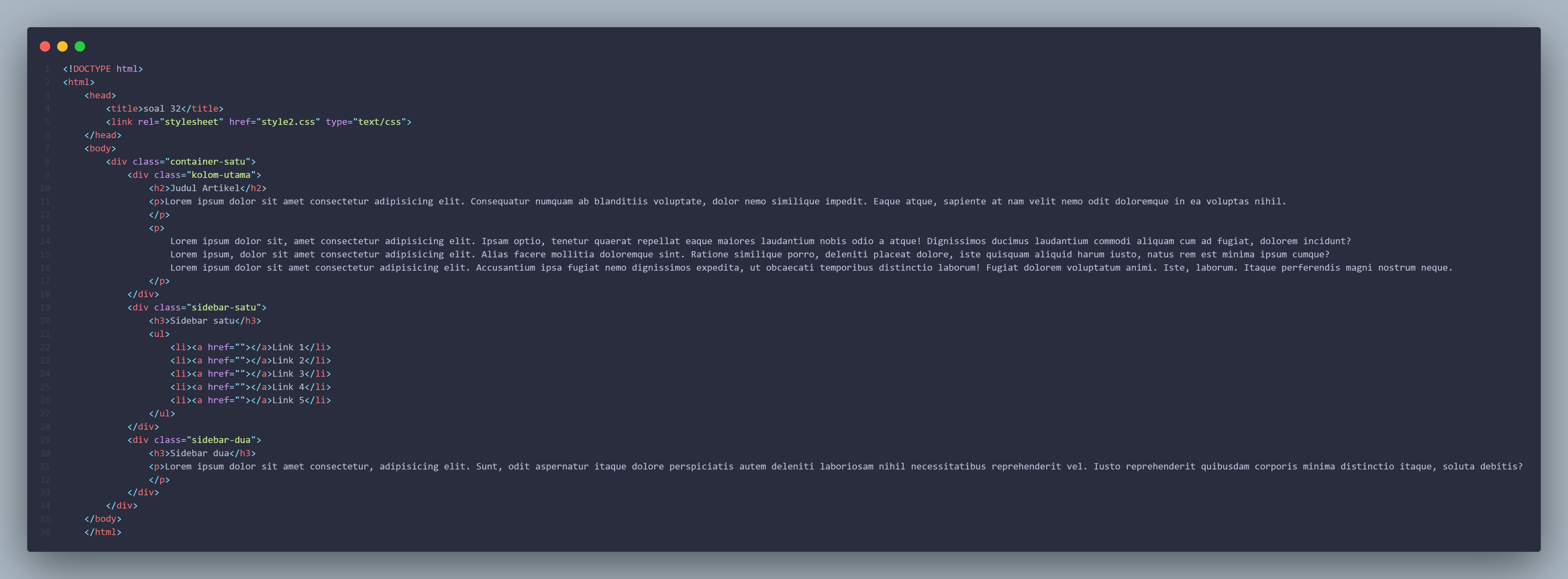
1. Buatlah tampilan dari hasil *flex-box* sebelumnya menjadi seperti ini.



1. Tambahkan gambar dan terapkan permainan warna ke latar belakang agar lebih menarik. **Kata kunci**; Gunakan *properti Flex* dan *Order* pada masing-masing *pemilih itemnya*. Kemudian *screen shoot* kode dan hasilnya.







Link GitHub : https://github.com/lostboiii/pemrograman-web/tree/main/minggu2

Referensi:

1. Jason Beaird, Prinsip-prinsip Desain Web yang Indah
2. Rian Ariona, Belajar HTML dan CSS (Tutorial Dasar belajar HTML dan CSS)
3. Adi Hadisaputra, Dasar-dasar HTML dan CSS dari Akar hingga Daun John Duckett, HTML dan CSS merancang dan membangun situs web
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>)