

Modul 1: Desain UI/UX



Gambar 1 User Interface Mesin ATM

UI/UX Design

UI atau **User Interface** dan UX atau **User eXperience** adalah istilah yang baru muncul setelah aplikasi gengsar dikembangkan dalam format Smartphone dengan basis Sistem Operasi Android maupun iOS. Proses perancangan aplikasi yang semakin hari semakin rumit memaksa kita untuk memisahkan bagian desain dengan bagian programming. Perkembangan peralatan yang harus menggunakan antarmuka juga semakin memperlebar jurang antara desain dengan programming tersebut.

Maka, muncullah istilah User Interface dan User eXperience yang kemudian menjadi satu kesatuan dengan istilah populer UI/UX Design. Meskipun sebenarnya kedua istilah tersebut menimbulkan makna yang ambigu, sehingga membuat banyak orang bertanya-tanya tentang apa yang membedakan antara UI (User Interface) dan UX (User eXperience). Berikut ini gambaran umum mengenai kedua istilah tersebut.

Apa yang dimaksud dengan **User Interface**?

User Interface adalah antar muka yang dirancang untuk digunakan oleh user agar user dapat memanfaatkan aplikasi. Jadi, UI atau User Interface lebih bersifat teknis yakni tentang bagaimana sebuah aplikasi ditampilkan ke dalam bentuk antarmuka yang dapat dilihat oleh user. Tidak peduli apakah penggunaannya akan merasa nyaman dengan tampilan yang dibuat atau merasa tidak nyaman. Bukan itu yang menjadi tujuan User Interface. Yang penting pengguna aplikasi dapat menggunakan aplikasi melalui antarmuka yang disediakan oleh pengembang software.

Bagaimana **User Interface** ini secara sederhana dapat menjembatani aplikasi dengan penggunanya bisa kita lihat dari contoh mesin ATM. Ketika kita menggunakan mesin ATM, maka kita akan disuguhkan dengan **User Interface** sederhana, namun dengan antarmuka tersebut kita dapat berinteraksi dengan mesin ATM untuk mengambil uang, transfer ke rekening lain, melakukan pembayaran uang kuliah, membayar pesanan toko online, dan sebagainya. Pendeknya, **User Interface** adalah antarmuka pada aplikasi sebagai alat bantu bagi user agar dapat menggunakan aplikasi.

Apa yang dimaksud dengan **User eXperience**?

User eXperience secara harfiah dapat diterjemahkan sebagai “Pengalaman User”. Mungkin istilah ini sedikit rancu, namun yang ingin disampaikan dengan istilah ini adalah mengenai bagaimana user berinteraksi dengan aplikasi sehingga user berhasil mencapai tujuan yang ingin dilakukannya dengan perantaraan aplikasi tersebut. Adapun yang dilakukannya bisa berbeda-beda tergantung pada apa yang ingin dicapai oleh user tersebut.

Pada contoh mesin ATM tadi, user bisa saja melakukan sesuatu yang pilihannya telah disediakan oleh ATM. Contohnya: user menarik sejumlah uang, user melakukan transfer ke nomor rekening lain, user melakukan pembayaran tertentu, dan lain-lain.

Pada prakteknya, tidak semua kegiatan tersebut dilakukan dengan menggunakan antarmuka pada ATM, namun semua kegiatan “dieksekusi” dengan menggunakan antarmuka ATM. Contohnya, membayar orderan online shop atau membayar uang kuliah menggunakan Virtual Account yang dibuat dengan menggunakan aplikasi di website atau smartphone.

Dengan demikian, istilah User Interface berbeda dengan User eXperience. **User Interface** adalah antarmuka yang disediakan oleh aplikasi, sedangkan **User eXperience** adalah langkah-langkah yang dilakukan oleh user untuk mencapai tujuannya dengan menggunakan aplikasi.

Pada prakteknya, pengalaman tersebut menjadi sebuah hal yang diingat oleh user, sehingga apabila kita sering melakukan hal yang sama berulang-ulang maka kita akan mengingatnya dengan baik. Contohnya, bagaimana cara transfer uang dengan menggunakan mesin ATM? Maka sudah terbayang langkah demi langkah yang harus dilakukan ketika kita berada di depan mesin ATM, yakni: masukkan kartu ATM – masukkan kode PIN – pilih TRANSFER - masukkan nomor rekening – ATAU cari nomor kode Bank terlebih dahulu – masukkan nominal – masukkan kode – pilih BENAR – selesai.

Menjadi UI/UX Desainer

UI/UX Design adalah sebuah tantangan yang menarik. Sebuah pekerjaan yang memadukan keterampilan merancang halaman dengan kemampuan pemrograman. Dua hal yang “wajib” dimiliki oleh para pembuat aplikasi.

Kemampuan merancang dan kemampuan pemrograman ini tentu membutuhkan penguasaan yang mumpuni untuk dapat menghasilkan aplikasi yang tampil secara menarik, sekaligus mudah diakses. Ibarat menyatukan dua kutub yang berbeda. Dua kemampuan tadi tentu menyuguhkan problematika yang sangat bertentangan. Merancang halaman membutuhkan seni dan kreativitas yang tinggi, yang lebih mengedepankan “citarasa” dengan pendekatan desain, seperti warna, nuansa, keseimbangan, fokus, irama, dll. Sedangkan kemampuan programming sangat kental dengan disiplin ilmu informatika seperti syntax dan algoritma yang harus taat azas.

Memiliki kedua kemampuan tersebut adalah sebuah anugerah yang sangat tak ternilai. Hal ini jarang dimiliki oleh kebanyakan orang. Seorang desainer grafis biasanya sangat menyukai pekerjaan-pekerjaan yang berhubungan dengan gambar, warna, tata letak halaman, dan sebagainya. Mereka

akan bekerja dengan cepat dan mengasyikkan dalam hal merancang, namun sulit untuk diajak bergelut dan berkonsentrasi dengan syntax pemrograman yang biasanya dianggap rumit. Sedangkan seorang programmer justru sebaliknya, biasanya sangat terampil dalam menyelesaikan masalah-masalah pemrograman, akan tetapi sangat lamban untuk diajak merancang halaman-halaman yang menarik. Tampilan halaman yang mereka buat biasanya sangat kaku, tidak berselera, dan sama sekali tidak berkelas.

Oleh karena itu, berbahagialah Anda yang dapat memadukan seni dan kreativitas desain sekaligus mengerti bahasa pemrograman. Dengan perpaduan keduanya, maka pintu untuk menjadi UI/UX Designer sudah terbuka lebar. Setidaknya, dengan memahami bagaimana cara membuat layout halaman aplikasi, dan kemudian mengetahui struktur format aplikasi standar, maka sudah cukup untuk dapat menciptakan aplikasi yang baik dan menarik, sekaligus siap untuk “menjual” kemampuan Anda.

Namun, memiliki dua hal itu saja ternyata belumlah cukup. Dibutuhkan kemampuan-kemampuan lain yang mengharuskan seorang UI/UX Designer bisa berakrobat. Software-software alat bantu merancang halaman sudah barang tentu harus menjadi teman setia dalam melakukan tugas sehari-hari. Hal-hal lain tersebut tentu cukup banyak untuk bisa disebutkan. Dari mulai keterampilan membuat animasi (gif dan canvas), script programming (ASP, PHP, Javascript, Python, dkk), database (SQL, MySQL), pengelolaan situs atau CMS, sampai pada hal-hal yang berhubungan dengan SEO (Search Engine Optimization) atau mengatur bagaimana caranya agar aplikasi yang sudah dibuat dapat tampil di halaman-halaman awal Search Engine terkenal.

Nah, hal-hal yang telah disebutkan di atas adalah pintu untuk membuka jendela kemampuan kita dalam “menjual diri” sebagai UI/UX Designer. Dengan pengetahuan yang lengkap, maka aplikasi serumit apapun dapat kita ciptakan. Dengan pengetahuan yang lengkap pula, maka kita tidak perlu khawatir dengan berbagai keinginan klien yang sangat mungkin akan sering muncul seiring perjalanan kita menjadi UI/UX Designer.

Namun, ternyata itu pun masih belum seberapa! Dengan segala macam pengetahuan yang telah kita miliki tentu kita tidak ingin berakhir di belakang meja komputer di bawah tangga kantor. Untuk itu dibutuhkan seni akrobat yang lain yang tak kalah seru, yakni bagaimana “memasarkan” semua keterampilan itu dan kemudian bernegosiasi dengan klien atau calon klien.

Software untuk UI/UX Design

Beberapa software yang dapat digunakan untuk UI/UX Design sudah bisa diaplikasikan, baik yang berbayar maupun yang open source. Contohnya **Adobe XD** yang dirilis oleh Adobe, sebuah software developer yang sangat kuat. Adobe sangat dikenal dengan software-software dalam bidang desain, sehingga tidaklah mengherankan apabila Adobe juga merilis Adobe XD yang juga merupakan software dalam bidang desain. Selain Adobe XD, ada juga software yang dapat digunakan secara online yaitu **FIGMA** (figma.com). Software ini juga dapat digunakan untuk UI/UX Design dan menyediakan fitur untuk Colaboration sehingga sebuah proyek dapat dikerjakan oleh beberapa orang sekaligus.

Masih ada beberapa software lain yang juga dapat digunakan dalam lingkungan UI/UX Design, contohnya software **ZEPELIN** (zepelin.com) yang dapat menjembatani desain dan teknis. Pada dasarnya software-software UI/UX Design ini adalah software desain yang juga dibekali dengan fitur-fitur untuk menghubungkan satu halaman dengan halaman lainnya.

Desain Web dengan HTML

Website pada awalnya adalah sesuatu yang sederhana. Ia diciptakan untuk mempermudah penyampaian informasi dari penyedia informasi (server) kepada browser/peselancar di seluruh dunia. Pengiriman informasi ini dilakukan melalui jaringan yang sangat dibatasi oleh kecepatan MODEM dan lebar jalur data (Bandwidth) yang dipergunakan. Oleh karena itu, website harus dibuat dengan format yang sesederhana mungkin, yang memungkinkan semua informasi di dalamnya dapat dikirim dengan cepat.

Website diciptakan dengan plain text biasa. Kita dapat membuat website dengan hanya mengandalkan text editor sederhana seperti “Notepad” yang ada pada Microsoft Windows. Hal ini dimungkinkan karena semua format penulisannya disertakan pada text yang kita tulis. Pada saat-saat awal perkembangannya, website tidaklah serumit sekarang, melainkan hanya berupa text formatting biasa agar tulisan dapat ditampilkan dan dibaca dengan nyaman, ditambah dengan link yang dapat menghubungkan kita ke halaman-halaman yang disediakan.

Perkembangan teknologi yang sangat cepat dan kebutuhan penyimpanan informasi yang semakin canggih membuat website harus dapat dipergunakan untuk menyiapkan dan mengirimkan informasi secepat mungkin. Contohnya berita yang ada di suatu tempat dapat ditulis pada saat itu juga dan dalam hitungan menit langsung dapat dibaca di seluruh dunia. Untuk itu website dikembangkan dengan kemampuan untuk menyimpan dan membaca database. Informasi yang ditulis disimpan dalam database, kemudian website bekerja untuk menampilkan isi database yang berupa tulisan tadi.

Banyak format database yang dipergunakan untuk kebutuhan internet, misalnya Oracle, SQL dan MySQL. Tujuan dari penggunaan database ini adalah untuk menyimpan informasi yang nantinya akan ditampilkan melalui website. Contoh paling praktis untuk menjelaskan kebutuhan database adalah sebagai berikut:

Ketika kita akan menuliskan pesan singkat yang dapat dibaca oleh semua orang, maka pesan tersebut dapat kita ketikkan dalam form yang sudah disediakan oleh website. Tulisan tersebut disimpan dalam database, dan kemudian website akan membaca database dan meletakkan tulisan tadi dalam form yang sudah disiapkan.

Jadi, form- form ini dapat diibaratkan sebagai sebuah wadah kosong di dalam website, apapun yang kita ketikkan di dalamnya akan disimpan dalam database. Di dalam website kemudian disiapkan pula form kosong yang dipergunakan untuk menampilkan tulisan apapun yang sudah kita ketik tadi – yang dalam hal ini sudah tersimpan dalam database. Jadi, di dalam website hanya ada form-form kosong, sedangkan isinya dikirim dan disimpan dalam database.

Di lain pihak, agar dapat menarik banyak pengunjung, website juga harus tampil menarik. Oleh karena itu, tulisan yang ada pada website dapat diberi asesoris seperti ukuran dan tipe huruf, warna teks dan warna latar belakang, bahkan kemudian dapat diberi gambar (image). Penggunaan asesoris ini apabila dirancang dengan baik akan membuat nyaman para peselancar dan informasi yang disampaikan menjadi lebih efektif. Dalam hal ini peselancar dapat dengan mudah mencari dan membedakan berbagai bentuk informasi yang disajikan, contohnya teks, menu, image, dsb. Dengan kata lain, desain yang baik akan membuat penyampaian informasi menjadi lebih baik.

Bagaimana caranya memulai membuat sebuah website?

Semuanya berawal pada kode-kode HTML. Dengan memahami cara penulisan format HTML maka kita dapat membuat website yang baik. Dengan mengetahui kode-kode HTML maka kita dapat membuat website yang ringkas sekaligus terhindar dari kesalahan-kesalahan yang tidak perlu.

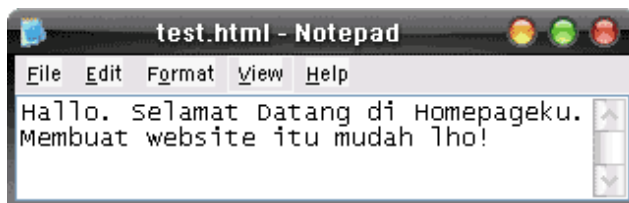
Membuat File HTML Itu Mudah

HTML (Hypertext Markup Language) bukanlah bahasa pemrograman yang rumit. Ia hanyalah sebuah “cara menulis” dalam format yang bisa dibaca oleh semua internet browser. Contohnya, dengan HTML kita dapat membuat tulisan yang dicetak tebal, miring, diberi garis bawah, dicoret, dst. Semuanya ditulis dengan format tulisan sederhana dan dapat dibaca oleh penulisnya maupun oleh internet browser. Cara penulisan dengan melampirkan kode-kode format tampilan tersebut dinamakan dengan **Tag**. HTML Tag dipergunakan untuk menuliskan perintah kepada browser agar formatnya ditampilkan sesuai dengan perintah yang ditulis di dalam Tag.

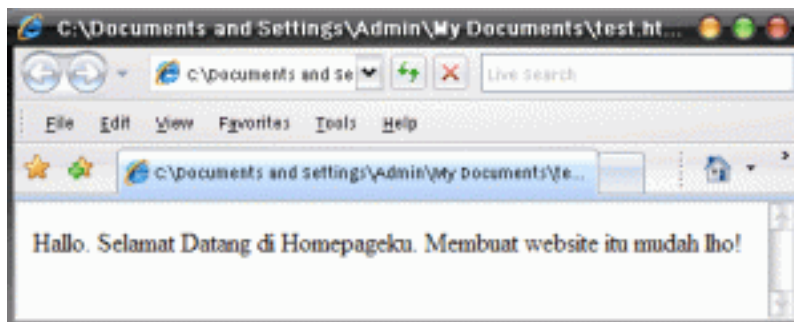
Membuat file HTML dapat dilakukan dengan mudah. Bukalah Notepad yang disediakan oleh Microsoft Windows, kemudian ketikkan sesuatu di dalamnya, contohnya:

Hallo. Selamat Datang di Homepageku. Membuat website itu mudah lho!

Simpan text tersebut dengan perintah **Save As**. Ketikkan nama filenya dengan tambahan “.html”. Contohnya: “**test.html**”, kemudian pada bagian **Save as type**: pilih **All Files**, dan simpanlah file tersebut pada folder **My Documents**.



Bukalah Explorer dan carilah file **test.html** tadi di folder **My Documents**. Klik ganda untuk membuka file tersebut. Maka program Internet Browser akan terbuka dan menampilkan tulisan yang telah kita simpan. Sederhana, bukan ?



Nah, dengan cara ini Anda dapat menuliskan sepanjang apapun sesuka Anda, tanpa harus takut ukuran filenya menjadi besar.

Text Formating

HTML (Hypertext Markup Language) bukanlah bahasa pemrograman yang rumit. Ia hanyalah sebuah “cara menulis” dalam format yang bisa dibaca oleh semua internet browser. Mirip dengan cara penulisan pada Word Processor di tahun 80-an. Saat itu kita mengenal Wordstar sebagai salah satu software khusus untuk pengetikan dengan komputer. Pada software tersebut, untuk menulis dengan huruf yang dicetak tebal kita harus memasukkan perintah pada awal tulisan yang akan dicetak tebal, dan mengetikkan perintah setelah tulisan yang dicetak tebal tersebut untuk mengembalikan ke bentuk tulisan biasa.

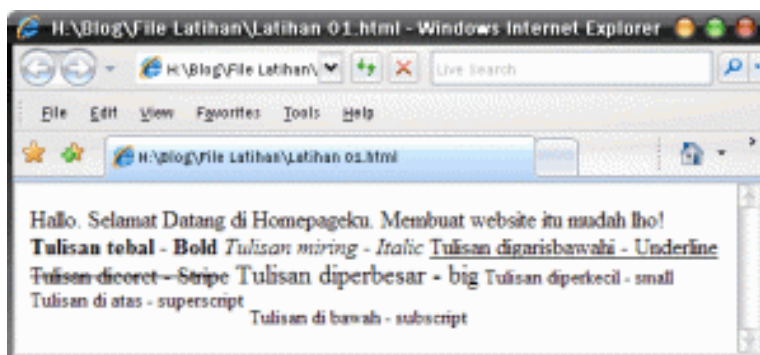
Format penulisan perintah pada HTML dikenal dengan **Tag**. Kode tag dituliskan di dalam tanda kurung siku **<Tag>**. Tag HTML ini ada yang berpasangan untuk mengapit tulisan yang diberi format penulisan yang berbeda, dan ada yang tidak berpasangan sebagai format perintah yang tidak mengatur tulisan.

Contohnya, kita bisa membuat tulisan yang dicetak tebal dengan mengetikkan **** dan ditutup dengan ****. Bukalah file **test.html** sebelumnya, kemudian tambahkan tulisan sebagai berikut:

```
Hallo. Selamat Datang di Homepageku. Membuat website itu mudah lho!
<b>Tulisan tebal - Bold</b>
<i>Tulisan miring - Italic</i>
<u>Tulisan digarisbawahi - Underline</u>
<s>Tulisan dicoret - Stripe</s>
<big>Tulisan diperbesar - big</big>
<small>Tulisan diperkecil - small</small>
<sup>Tulisan di atas - superscript</sup>
<sub>Tulisan di bawah - subscript</sub>
```

Save As file ini sebagai **latihan 01.html**, dan bukalah file tersebut dengan menggunakan Internet Browser.

Hasilnya akan terlihat seperti ini:

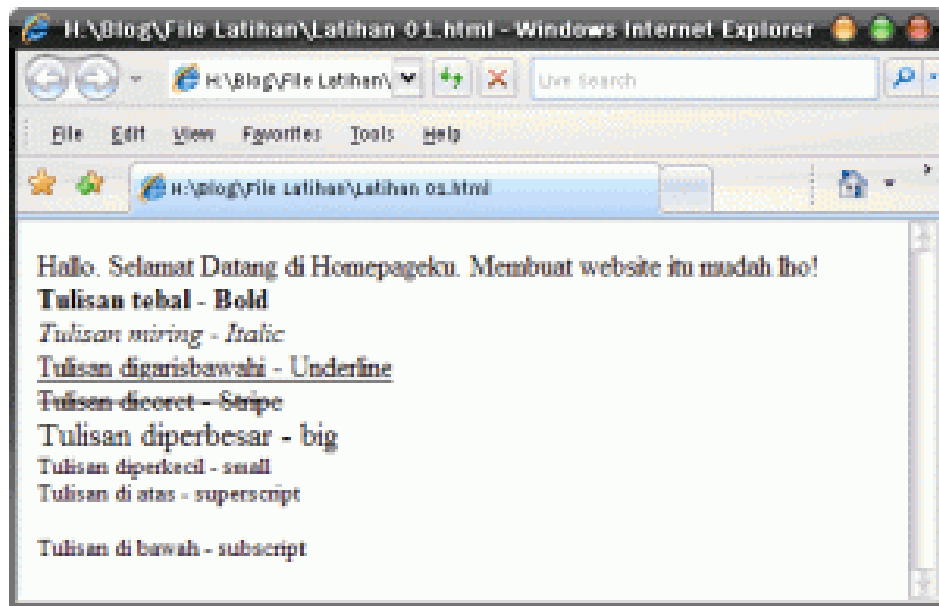


Pada contoh di atas, format tulisan ditampilkan sesuai dengan kode tag yang kita buat. Akan tetapi semua tulisan disatukan tanpa adanya pemisahan baris. Hal ini dapat diperbaiki dengan penggunaan tag break **
** sehingga tulisannya menjadi seperti ini:

```
Hallo. Selamat Datang di Homepageku. Membuat website itu mudah
lho!<br>
<b>Tulisan tebal - Bold</b><br>
<i>Tulisan miring - Italic</i><br>
<u>Tulisan digarisbawahi - Underline</u><br>
<s>Tulisan dicoret - Stripe</s><br>
<big>Tulisan diperbesar - big</big><br>
<small>Tulisan diperkecil - small</small><br>
<sup>Tulisan di atas - superscript</sup><br>
<sub>Tulisan di bawah - subscript</sub>
```

Save file ini dan bukalah dengan menggunakan Internet Browser.

Hasilnya akan terlihat seperti ini:



Tag Heading

Tag Heading dipergunakan untuk format tulisan Judul dan Subjudul. Terdapat enam tag heading yang dapat dipergunakan, dari **<H1>...</H1>** untuk format Heading paling besar sampai dengan **<H6>...</H6>** untuk format Heading paling kecil.

```
<H1>Heading Satu</H1>
<H2>Heading Dua</H2>
<H3>Heading Tiga</H3>
<H4>Heading Empat</H4>
<H5>Heading Lima</H5>
<H6>Heading Enam</H6>
```

Jika di **Save**, maka hasilnya akan tampak sebagai berikut:

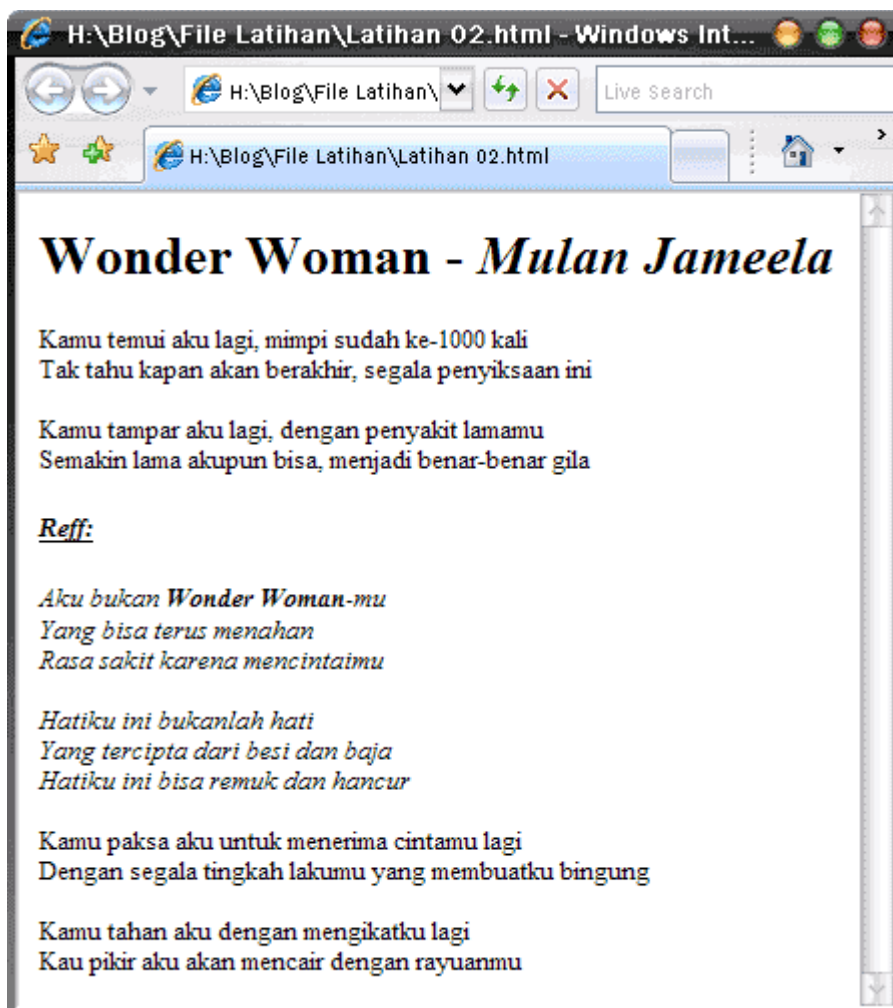


Penggunaan **Tag Heading** sangat tepat untuk penulisan Judul dan Subjudul. Perhatikan bahwa sesudah penutup **Tag Heading** tidak perlu menuliskan **Tag Line Break
** untuk pindah baris. Dengan kata lain, **Tag Heading** adalah format penulisan Paragraf.

Nah, dengan mengetahui text formatting pada HTML, kita bisa berlatih menuliskan lirik lagu kesukaan kita. Contohnya sebagai berikut:

```
<H1>Wonder Woman - <i>Mulan Jameela</i></H1>
Kamu temui aku lagi, mimpi sudah ke-1000 kali<br>
Tak tahu kapan akan berakhir, segala penyiksaan ini<br>
<br>
Kamu tampar aku lagi, dengan penyakit lamamu<br>
Semakin lama akupun bisa, menjadi benar-benar gila<br>
<br>
<H4><i><u>Reff:</u></i></H4>
<i>Aku bukan <b>Wonder Woman</b>-mu<br>
Yang bisa terus menahan<br>
Rasa sakit karena mencintaimu<br>
<br>
Hatiku ini bukanlah hati<br>
Yang tercipta dari besi dan baja<br>
Hatiku ini bisa remuk dan hancur</i><br>
<br>
Kamu paksa aku untuk menerima cintamu lagi<br>
Dengan segala tingkah lakumu yang membuatku bingung<br>
<br>
Kamu tahan aku dengan mengikatku lagi<br>
Kau pikir aku akan mencair dengan rayuanmu
```

Hasilnya akan terlihat sebagai berikut:



Perhatikan bahwa dalam contoh latihan di atas, format penulisan dapat digabung-gabungkan. Artinya, kita bisa membuat huruf miring (*Italic*) sekaligus huruf tebal (**Bold**), meskipun awal dan akhir kode huruf miring berbeda dengan dengan awal dan akhir kode huruf tebal.

Desain Web dengan HTML vs UI/UX Design

Pada contoh di atas, kita bisa melihat bahwa untuk membuat halaman website sangat diperlukan pengetahuan tentang bahasa HTML. Hal ini dilakukan karena HTML adalah struktur utama dari sebuah halaman web sehingga kita harus memahami HTML jika kita ingin membuat halaman web.

Pada masa-masa awal berkembangnya website, perancangan website biasanya dilakukan dengan format HTML. Pada masa itu, sangatlah lumrah bagi perancang web dituntut untuk mengerti bahasa HTML agar website yang dirancangnya bisa langsung menjadi sebuah halaman yang siap pakai.

Akan tetapi, pada masa sekarang untuk membuat halaman web diperlukan banyak hal lain selain HTML di antaranya ada CSS, Javascript, PHP, Bootstrap, Framework, Python, AJAX, dan lain-lain. Hal ini membuat proses merancang web menjadi pekerjaan yang sangat berat bagi desainer yang tidak mengetahui berbagai bahasa tersebut. Oleh karena itu, dipandang perlu untuk memisahkan pekerjaan merancang yang dilakukan oleh desainer dan pekerjaan pengembangan web atau aplikasi yang dilakukan oleh developer.

Dengan adanya pemisahan tersebut, maka dapat dipastikan bahwa seorang desainer interface tidak perlu mengetahui program yang ada di balik aplikasi yang sedang dikerjakannya. Desainer akan fokus terhadap pekerjaannya, tidak perlu mempelajari bahasa pemrograman yang baginya sangat sulit dipahami. Akan tetapi, tentu akan lebih baik jika desainer tersebut mengetahui dan mengerti dasar-dasar pemrograman. Hal tersebut akan membuat apa yang dirancangnya memiliki dasar yang kuat dan tidak asal desain.

Contoh desain halaman login sebuah aplikasi kesehatan dengan menggunakan figma:

