

BUKU AJAR

Interface USER EXPERIENCE

DISUSUN OLEH
HIDAYATULAH HIMAWAN
MANGARAS YANU F



BUKU AJAR

Interface
USER
EXPERIENCE

DISUSUN OLEH
HIDAYATULAH HIMAWAN
MANGARAS YANU F





Hidayatulah Himawan
Mangaras Yanu F.

Copyright © Hidayatulah Himawan, Mangaras Yanu F. 2020
Hak cipta dilindungi oleh undang-undang

Dilarang mengutip atau memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotocopy, merekam, atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penyusun.

Cetakan Pertama, 2020
ISBN: 978-623-7594-55-0

Diterbitkan oleh:
Lembaga Penelitian dan Pengabdian kepada Masyarakat
UPN Veteran Yogyakarta
Jl. SWK 104 (Lingkar Utara), Condongcatur , Yogyakarta, 55283
Telp. (0274) 486188,486733, Fax. (0274) 486400

PRAKATA

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga Buku Ajar dengan topik bahasan “User Interface dan User Experience” dengan judul “**Interface USER EXPERIENCE**” ini telah selesai disusun. Buku ajar ini merupakan dokumentasi hasil penelitian yang telah dilakukan oleh penyusun, yang diharapkan dapat dijadikan sebagai pedoman untuk mengajar mata kuliah dengan topik bahasan “User Interface dan User Experience”.

Terimakasih kami sampaikan kepada Universitas Pembangunan Nasional “Veteran” Yogyakarta khususnya pada Lembaga Penelitian dan Pengabdian Kepada Masyarakat atas dukungan pendanaan sehingga kami dapat menyelesaikan penelitian yang kami jadikan sebagai dasar dalam penyusunan buku ini. Terimakasih juga kami sampaikan kepada semua pihak yang tidak bisa kami sebutkan satu-persatu yang telah berkontribusi baik dalam penelitian yang telah kami lakukan maupun dalam penyusunan buku ini.

Kami menyadari masih terdapat kekurangan dalam buku ini, untuk itu kritik dan saran terhadap penyempurnaan buku ini sangat diharapkan. Semoga buku ini dapat memberi manfaat bagi Universitas Pembangunan Nasional “Veteran” Yogyakarta khususnya dan bagi semua pihak yang membutuhkan.

Yogyakarta, Januari 2020

Penyusun

DAFTAR ISI

PRAKATA	1
DAFTAR ISI	3
BAB I DEFINISI UI UX.....	5
1.1 Definisi User Interface.....	5
1.2 Definisi User Experience	5
1.3 Perbedaan UI dan UX.....	6
1.4 UI Design	7
1.5 Memulai Menjadi UI Designer	8
1.6 Mengapa User Experience Itu Penting?	12
BAB II Dasar-Dasar UX	14
BAB III STUDI KASUS.....	26
BAB IV VARIABEL FONT	2
BAB V AKSESIBILITAS	10
5.1 Pengantar Fokus	19
5.2. Pentingnya Urutan DOM	22
5.3 Menggunakan tabindex.....	25
5.4 Mengelola fokus pada level laman	26
5.5 Mengelola fokus di komponen.....	26
5.6 Modal dan jebakan keyboard	28
BAB VI PENGANTAR SEMANTIK	31
6.1 Teknologi pendukung	31
6.2 Kemampuan	33
6.3 Pembaca layar	34
6.4 Pohon Aksesibilitas	35
6.5 Semantik di HTML asli	37
6.6 Alternatif Berupa Teks untuk Gambar	39
6.7 Semantik dan Menyusuri Materi	41
6.8 Menggunakan heading secara efektif	42
BAB VII Pengantar ARIA	45
7.1 Hubungan dan Label ARIA	49
7.2 Menyembunyikan dan Memperbarui Materi	53

7.3 Gaya yang Dapat Diakses	56
7.4 Cara Melakukan Tinjauan Aksesibilitas	65
7.5 Aksesibilitas untuk tim	70
BAB VIII ANIMASI	79
8.1 Animasi CSS Versus JavaScript.....	80
8.2 Dasar-Dasar Easing	84
8.3 Easing Khusus	88
8.4 Menganimasikan Antar Tampilan.....	91
8.5 Memilih Easing yang Tepat	94
8.6 Menganimasikan Tampilan Modal.....	95
8.7 Pengaturan waktu animasi asimetris.....	98
8.8 Animasi dan Kinerja.....	99
BAB IX Dasar-Dasar Desain Web Responsif.....	102
9.1 Responsive Web Design	103
9.2 Pola Desain Web Responsif.....	119
9.3. Gambar.....	128
9.4 Materi Multi-Perangkat.....	146

BAB I DEFINISI UI UX

Sasaran Pembelajaran

Mahasiswa mampu menguraikan dan menjelaskan definisi User Interface dan User Experience (UI/UX)

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer dan menguasai tools untuk mendesain

Keterkaitan bahan pembelajaran dengan pokok bahasan lainnya

Materi ini sebagai dasar untuk mempelajari pokok bahasan lainnya

Manfaat atau pentingnya bahan pembelajaran ini

Memberi pemahaman pengertian UI UX

1.1 Definisi User Interface

UI atau User Interface merupakan mekanisme komunikasi antara pengguna (user) dengan sistem pada sebuah program, baik itu aplikasi website, mobile, ataupun software. Mekanisme itu disesuaikan dengan kebutuhan pengguna terhadap program yang tengah dikembangkan. Cakupan UI itu meliputi tampilan fisik, penggunaan warna, tampilan animasi, hingga pola komunikasi suatu program dengan penggunanya.

Biasanya, seorang desainer UI akan membuat desain yang kiranya memudahkan pengguna programnya. Adapun, desain itu disesuaikan dengan tingkat kebutuhan dasar pengguna terhadap program aplikasi web ataupun mobile tersebut. Output dari hasil desainer UI ialah program dengan segala fitur yang kiranya sesuai dengan kebutuhan pengguna dalam menggunakan program tersebut.

1.2 Definisi User Experience

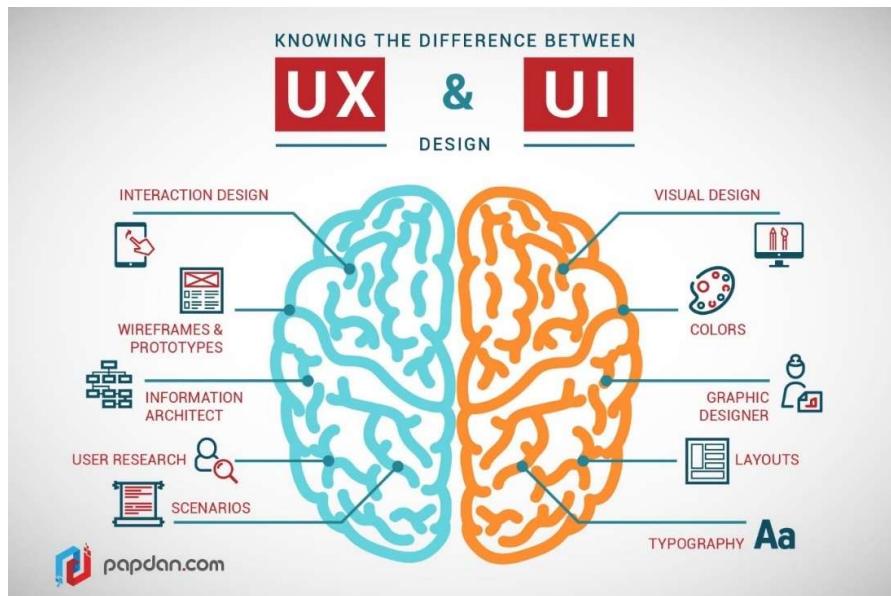
Pengertian UX atau User Experience memang tidak terlalu jauh berbeda dengan UI. Perbedaannya terletak pada fokus utama hubungan komunikasi antara pengguna dengan programnya, yakni berfokus pada pengalaman penggunanya.

Seorang desainer UX akan merancang program aplikasi web atau mobile berdasarkan pengalaman dari pengguna atau user setelah menggunakan aplikasi web atau mobile tersebut. Dengan begitu, program yang dirancang menjadi lebih mudah digunakan oleh penggunanya.

User experience (UX) sesuai artinya dalam bahasa indonesia “pengalaman pengguna” adalah pengalaman yang diberikan website atau software kepada penggunanya agar interaksi yang dilakukan menarik dan menyenangkan. Kalau dulu aplikasi mempunyai usability yang bagus saja sudah cukup. Sekarang sebuah aplikasi juga harus memiliki user experience yang bagus.

Seperti apa user experience itu? Saat membuka Instagram sampai berjam-jam tanpa bosan, saat chatting menggunakan WhatsApp tanpa henti, saat berlama-lama mencari barang-barang jualan di toko online, berarti sudah menikmati user experience yang sudah diberikan oleh Instagram, WhatsApp dan juga toko online. Kenapa juga bisa berjam-jam sibuk dengan smartphone? Itu semua karena penerapan user experience dalam smartphone sudah sangat baik.

1.3 Perbedaan UI dan UX



Sebelumnya sudah disinggung bahwa perbedaan antara UI dan UX berada pada fokus utamanya. Bila UI fokus pada interaksi pengguna dengan programnya, maka UX fokusnya pada pengalaman pengguna dalam menggunakan suatu aplikasi web atau mobile.

Seorang desainer UI akan mendesain program aplikasi web atau mobile sesuai dengan kebutuhan pengguna. Sehingga, ketika menggunakan program tersebut pengguna lebih mudah dan tidak kesulitan.

Sedangkan, desainer UX membuat program berdasarkan pengalaman dari penggunanya. Apa saja yang dirasakan dan kesulitan apa saja yang dihadapi ketika menggunakan program tersebut.

Sebenarnya, keduanya memiliki tujuan yang sama dalam mendesain program aplikasi web ataupun mobile, yakni memudahkan penggunannya. Oleh sebab itu, seringkali dalam proses perancangan sebuah program, desainer UI dan UX selalu berada dalam satu tim. Sebab, dengan

perpaduan keduanya, sebuah program aplikasi web ataupun mobile menjadi sangat mudah digunakan oleh pengguna tanpa harus membaca panduan. Seringkali, desainer UI dan UX bertukar data analisis untuk menyempurnakan program yang tengah dibuatnya.

Jadi, UI dan UX sebenarnya berbeda. Perbedaannya pada fokus utama. UI fokus pada kebutuhan pengguna terhadap program aplikasi aplikasi web atau mobile, sedangkan UX fokus pada pengalaman pengguna.

Pada dasarnya, User Experience adalah tentang “memahami pengguna”. Tujuan UX adalah mencari tahu siapa mereka, apa yang mereka capai dan apa cara terbaik bagi mereka untuk melakukan “sesuatu”.

UX berkonsentrasi pada bagaimana sebuah produk terasa dan apakah itu memecahkan masalah bagi pengguna.

Sedangkan User Interface adalah bagaimana suatu website atau aplikasi yang dibuat terlihat dan berbentuk seperti apa. Hal tersebut mencakup Layout (tata letak), Visual Design (desain visual) dan Branding.

Mengerti perbedaan antara UI Design dan UX Design, bukan sekedar untuk teori, tapi akan berpengaruh pada proses design. Beberapa orang (kalau bukan kebanyakan) menganggap design itu hanya terkait warna, pemilihan font, gambar/foto, dan icon. Padahal UX Design itu jauh melebihi warna dan sebagainya.

1.4 UI Design

Seperti namanya, (UI) User Interface. Maka ada 2 hal yang harus diperhatikan oleh seorang designer saat membuat UI, yaitu User dan Interface. Berikut gambarannya:



Untuk menghasilkan UI (User Interface) yang baik, Anda harus memperhatikan 2 hal diatas. Kata User ditempatkan di depan kata Interface, karena UI yang baik selalu memperhatikan dan mengutamakan user. UI yang baik akan membantu user. Dan, UI yang baik akan membuat user nyaman menggunakan.

1.5 Memulai Menjadi UI Designer

Untuk memulai menjadi UI Designer Anda boleh men-skip bagian User-nya dan fokus di Interface-nya dulu. Dalam pembuatan UI, Anda harus memikirkan tentang bagaimana nanti user memakainya. Contohnya: Sebuah Tombol.

Saat men-desain sebuah tombol, Anda harus memikirkan “Apakah user tahu kalau ini tombol yang bisa di-klik?”. Tapi, hal yang tidak kalah penting adalah Anda harus tahu bagaimana bentuk tombol itu. Maka dari itu, sebagai permulaan UI untuk fokus di bagian Interface-nya terlebih dahulu agar Anda familiar dengan bentuk-bentuk UI.

Apakah Anda familiar dengan Photoshop? GIMP? atau software sejenis lainnya?. Jika belum, maka sebaiknya Anda mulai belajar menggunakan software tersebut.

Langkah pertama untuk menjadi UI Designer adalah mencoba untuk membuat UI. Tidak perlu bingung mau mulai dari mana atau mau mendesain apa. Anda cukup memilih desain-desain interface yang ada di internet. Pilih yang Anda suka lalu jiplaklah, buat semirip mungkin.

Anda bisa browsing di situs seperti dribbble.com, behance.net atau kreavi.com. Banyak sekali desain UI disana.



Kenapa harus menjiplak desain?

Hal ini untuk membuat Anda familiar dengan UI. Bagaimana bentuk tombol, ukuran teks, jarak antar baris kalimat, jarak tiap elemennya.

Jadi, Semakin sering Anda menjiplak sebuah desain, maka Anda akan semakin familiar dengan ukuran setiap elemen UI. Tapi ingat, jiplak menjiplak ini hanya untuk kepentingan belajar. BUKAN untuk diupload di sosial media ataupun situs portfolio.

Jika Anda telah menjiplak UI berkali-kali, maka Anda sudah terbiasa membuat tombol, terbiasa membuat dropdown, input text, dan elemen-elemen UI lainnya. Kini, saatnya Anda untuk membuat UI Anda sendiri.

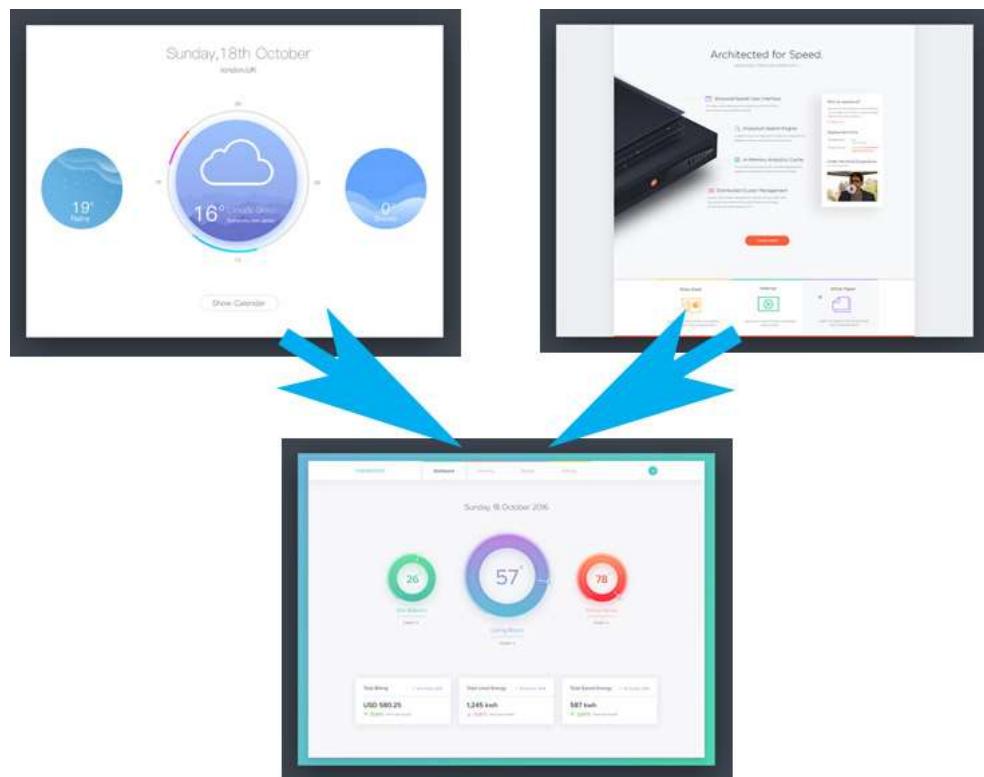
Caranya gampang, Pertama, tentukan platform terlebih dahulu. Anda ingin membuat apa? design untuk mobile app (Android/iOS), design untuk web, atau bahkan design untuk smartwatch app?

Setelah itu, tentukan tema design-nya. Anda ingin membuat tampilan untuk aplikasi kesehatan? website artikel teknologi? atau tampilan untuk aplikasi pemesanan makanan?

Setelah platform dan tema sudah dipilih, kini saatnya Anda browsing lagi design-design yang Anda sukai. Tapi kali ini, carilah design yang sesuai dengan platform dan tema yang Anda pilih tadi.

Lalu pilih 2 design yang paling Anda sukai, dan coba buat sesuatu yang baru dari hasil kombinasi dua design tersebut.

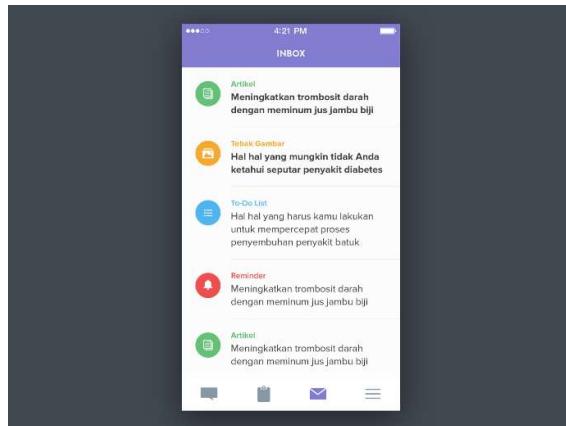
Contohnya seperti ini:



Saat mendesain sebuah UI, pikirkanlah bagaimana nanti user memakainya. Untuk melatih hal itu, biasakanlah mendesain dengan sebuah alasan. Apa artinya? Design yang Anda buat,

komponen yang Anda buat, sebaiknya memiliki alasan yang kuat kenapa diletakkan seperti itu atau berbentuk seperti itu.

Contoh 1:



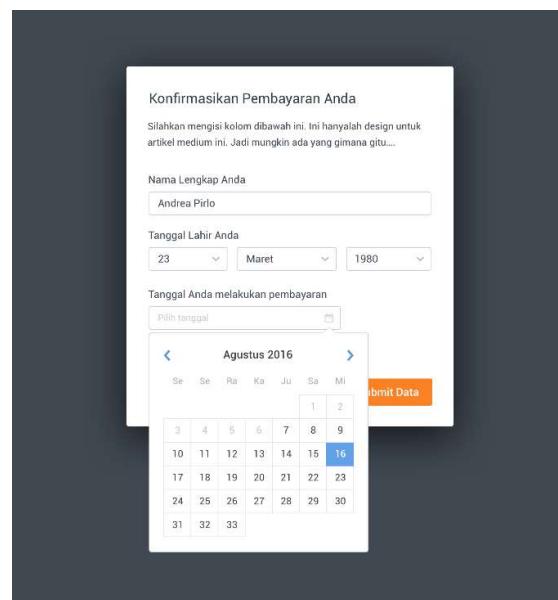
Kenapa menu nya ada dibawah?

Karena app ini memiliki 3 menu utama. Dan dalam penggunaan app ini, user cenderung berpindah-pindah dari menu satu ke menu utama lainnya dalam selang waktu relatif singkat. Maka dari itu, untuk mempermudah user, 3 menu utama ditaruh di bawah dan sisanya ditaruh di menu more yang terletak di paling kanan.

Kenapa ada tulisan yang tebal dan ada yang tidak tebal?

Ini adalah menu Inbox, tulisan yang tebal sebagai penanda belum dibaca. Dan yang tidak tebal sebagai penanda sudah pernah dibaca. Sehingga memudahkan user untuk membedakannya.

Contoh 2:



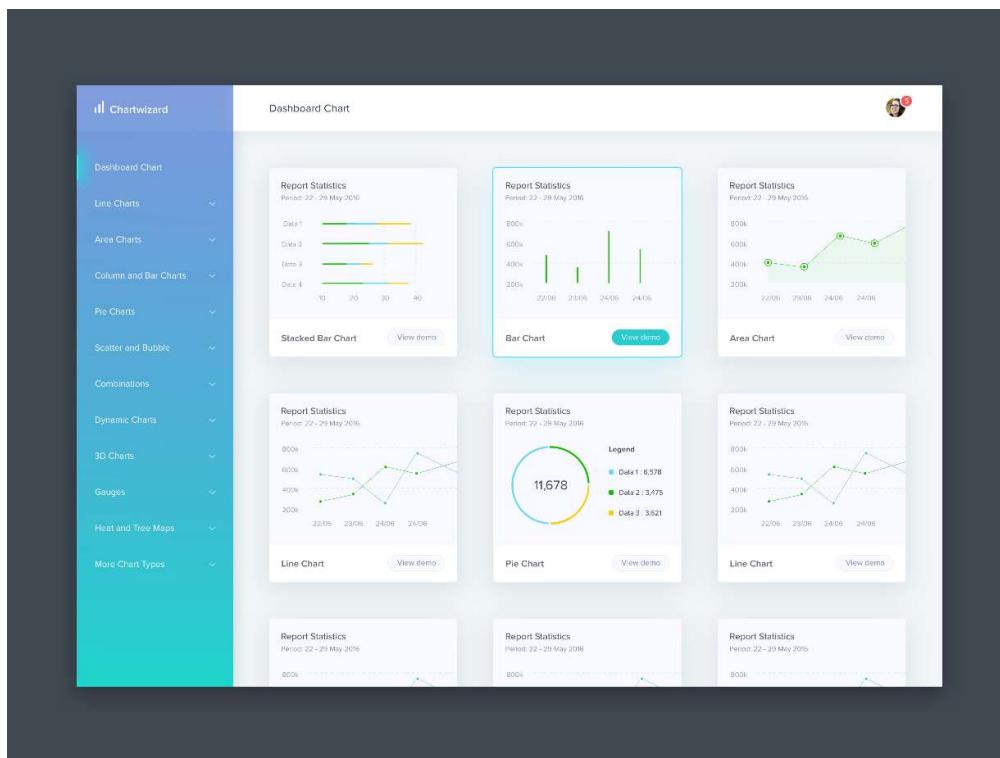
Ada dua inputan tanggal, tapi kenapa beda? Padahal kan sama-sama memilih tanggal.

Coba bayangkan jika Anda menjadi user. Untuk mengisi tanggal lahir, lebih mudah memakai format pengisian yang atas atau yang bawah? Tentu lebih mudah memakai format yang atas. Karena, Anda ingat tanggal lahir Anda. Dan dengan cepat akan mengisi hari, bulan dan tahunnya. Tapi, coba bayangkan jika menggunakan format pengisian yang bawah, berapa kali Anda harus mengklik tombol back untuk menuju bulan Maret 1980.

Lalu untuk inputan tanggal pembayaran, kenapa format pengisian-nya harus seperti itu?

Ini adalah form konfirmasi pembayaran, setelah user melakukan pembayaran, biasanya user akan langsung mengkonfirmasikan pembayaran-nya. Coba bayangkan, jika user harus mengisi dengan format pengisian seperti tanggal lahir. Cukup merepotkan user, karena dia harus mengingat hari ini tanggal berapa. Tapi dengan format pengisian seperti di design, ketika user memilih tanggal, maka akan terbuka kalender bulan ini, dengan tanggal hari ini ditandai biru. Sehingga user dengan mudah memilih tanggal pembayarannya.

Contoh 3:



Kenapa menu nya ada di kiri dan ke bawah? apakah karena ini dashboard yang tren-nya selalu menaruh menu di sebelah kiri?

Menaruh menu di kiri dan vertical ke bawah, pertimbangannya adalah skalabilitas. Sebelum merancang design ini, kami menganalisa bahwa kedepannya, menu menu di dashboard ini akan bertambah dan pertambahannya bisa cukup banyak. Jika membuat design menu-nya secara horizontal, maka kelak jika menunya sangat banyak, akan terjadi masalah pada layout-nya.

Tentu Anda tidak ingin melihat 2 baris menu terletak di bagian atas. Maka dari itu, diputuskan untuk membuat design menu-nya secara vertical ke bawah. Jadi, jika ada pertambahan menu, tidak akan terjadi masalah pada layout-nya.

1.6 Mengapa User Experience Itu Penting?

Memudahkan pengguna

Penerapan user experience akan memudahkan pengguna dalam menggunakan aplikasi. Karena didalamnya sudah ada penilaian aspek usability. Setiap aplikasi pastilah dibuat agar para pengguna mudah untuk menggunakannya.

Menarik minat pengguna

Selain faktor kemudahan penggunaan, penerapan user experience juga untuk menarik minat pengguna. Tujuan aplikasi dibuat sudah pasti ingin penggunanya selalu menggunakan aplikasinya. Jika aplikasi tidak menarik untuk pengguna, sudah pasti bisa dengan mudah akan ditinggalkan.

Berdampak pada faktor kesuksesan

Seperti pada poin diatas, pengguna akan mudah meninggalkan aplikasi yang tidak memberikan pengalaman menarik. Contoh aplikasi chatting WhatsApp. Aplikasi ini makin hari makin banyak penggunanya karena memberikan pengalaman pengguna yang menarik. Kita bisa bandingan dengan aplikasi sejenis yang penggunanya semakin menurun setiap hari. Oleh karena itu, user experience penting diterapkan pada sebuah aplikasi untuk meningkatkan kesuksesan atau minimal mempertahankan kesuksesan.

Menghasilkan UI yang bagus

User interface itu merupakan keluaran dari penerapan user experience. Jika user experience pada sebuah aplikasi benar-benar diperhatikan penerpannya, maka akan menghasilkan desain UI yang bagus. Bagus disini tidak harus warna-warni dan bling-bling, namun secara tampilan akan elegan dan menarik.

Untuk memenangkan persaingan

Apakah Anda mengetahui mengapa toko ritel seperti indomaret dan alfamart bisa memenangkan persaingan dari toko di sekelilingnya? Itu karena mereka menerapkan user experience yang bagus. Padahal barang yang dijual sama, secara harga bisa lebih mahal, tapi secara pengalaman yang mereka berikan itu tidak ada di toko biasa.

Contoh lain lagi yaitu antara Android dan IOS. Mengapa orang mau bayar mahal smartphone ini? Karena user experience yang mereka berikan memang terbaik. Berikut beberapa hasil penelitian mengenai pentingnya UX:

1. Menurut penelitian dari Imaginovation, sebuah lembaga penelitian berbasis di Amerika: Jika konten Anda tidak dioptimalkan dengan baik, sebanyak 79% pengunjung akan keluar dari website Anda dan mencari konten/produk lainnya.
2. Menurut penelitian dari lembaga riset HubSpot: pengguna ponsel 5X lebih punya kecenderungan untuk meninggalkan website Anda jika website tidak dioptimalkan agar sesuai dengan perangkat yang mereka punya. (Gawat kalau setidaknya ada 2/3 pelanggan yang akses website Anda dari ponsel mereka sebenarnya ingin melakukan pembelian pada hari itu juga)
3. Menurut penelitian dari lembaga riset MindTouch: Ini kasus nyata, pendapatan dari website ESPN.com melonjak 35% setelah mereka mendengarkan keluhan pengguna mereka dan mendesain ulang homepage mereka.
4. Menurut Adobe: 39% orang akan berhenti mengakses website jika gambar tidak dimuat-muat atau terlalu lama loading-nya.

Jadi intinya, website Anda sekarang ini fungsinya mirip seperti toko. Bayangkan user sebagai seorang calon pembeli yang masuk ke toko Anda. Jika mereka mengalami pengalaman yang buruk, contohnya si calon pembeli ini tidak bisa menemukan apa yang mereka butuhkan. Tak hanya itu, bisa juga pihak toko tidak berhasil menjangkau calon pembelinya, maka si calon pembeli pasti akan pergi dan tidak akan kembali lagi.

BAB II Dasar-Dasar UX

Sasaran Pembelajaran

Mahasiswa mampu memahami dasar-dasar User Experience (UX)

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer

Keterkaitan bahan pembelajaran dengan pokok bahasan lainnya

Materi ini memperkenalkan alur kerja yang bisa membantu tim, produk, startup dan perusahaan membuat proses yang kuat dan valuable untuk mengembangkan UX yang lebih baik bagi para pelanggan. Anda bisa menggunakan bagian proses yang berbeda secara terpisah, namun proses ini idealnya bekerja sangat baik bila dilakukan dalam serangkaian langkah

Manfaat atau pentingnya bahan pembelajaran ini

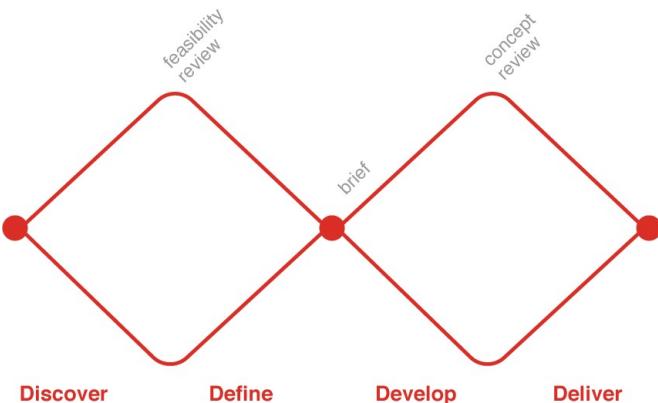
Memberi pemahaman dasar-dasar UX

Petunjuk belajar

Materi ini banyak mengadopsi metodologi Design Sprint yang digunakan beberapa tim di Google untuk memecahkan masalah dan tantangan seperti Self Driving Car dan Project Loon

Double Diamond

Alur kerja ini didasarkan pada apa yang kita sebut dalam lingkaran UX sebagai double diamond, dipopulerkan oleh British Design Council, dengan tim Anda terbagi untuk memahami ide melalui penelitian dan kemudian berkumpul untuk mendefinisikan tantangan, membaginya untuk membuat sketsa secara individual, berbagi ide, memutuskan apa yang terbaik ke depannya, pengujian dan validasi.



Model proses desain "double diamond" yang dipelopori oleh British Design Council, langkah-langkahnya melibatkan tahapan proyek berikut; *Memahami, Mendefinisikan, Membagi, Memutuskan, Prototipe dan Validasi*.

Menyiapkan langkah

Hal pertama adalah memulai dengan tantangan mendasar dan menulisnya seperti proposal, tanyakan pada diri Anda sendiri, "apa masalah yang sesungguhnya coba saya pecahkan?". Pernyataan tantangan adalah keterangan singkat yang ditetapkan ke proyek yang berisi tujuan Anda.

Tantangan ini bisa fitur produk saat ini yang perlu disaring atau produk yang sama sekali baru. Apapun tugas Anda, cukup sesuaikan bahasa agar sesuai dengan tujuan yang ingin Anda capai. Pernyataan harus dikaitkan dengan tujuan tim Anda, berfokus pada pengguna, memberikan inspirasi dan ringkas.

Berikut adalah beberapa contoh produk nyata yang telah dikerjakan:

- Merancang sebuah sistem untuk mengelola pengobatan dan perawatan lanjutan pasien penderita clubfoot.
- Membuat sebuah aplikasi yang menyederhanakan sistem keuangan kompleks dan menguranginya ke hal-hal penting saja.
- Merancang aplikasi seluler yang konsisten di seluruh platform yang berbeda tanpa mengorbankan merek.

Memperbarui pernyataan tantangan

Setelah Anda menulis beberapa variasi tujuan, presentasikan ke tim Anda untuk mendapatkan sebuah konsensus. Anda mungkin perlu memasukkan batas waktu karena ini akan membantu tim berfokus pada masalah. Jadi dengan penambahan tersebut, penyesuaian untuk daftar di atas bisa menjadi:

- Merancang sebuah sistem untuk mengelola pengobatan dan perawatan lanjutan anak-anak di bawah usia 2 tahun penderita clubfoot diluncurkan pada Q1 tahun ini.
- Membuat aplikasi keuangan sederhana yang memungkinkan Anda membeli dan menjual saham cukup dengan mengetuk tombol tanpa membutuhkan pengetahuan dasar dunia keuangan, dengan peluncuran awal Juli 2017.
- Menghasilkan panduan desain yang fleksibel di beberapa platform dan memosisikan merek perusahaan secara efektif pada setiap platform hingga akhir tahun ini.

Ketika pernyataan tantangan selesai, tampilkan dalam tempat yang menonjol sehingga Anda bisa melihatnya saat bekerja. Anda harus memeriksanya kembali secara konstan, bahkan mungkin memperbarui atau memodifikasinya selama proyek Anda berjalan.

Memvalidasi masalah

Langkah berikutnya adalah meneliti tantangan dan mempelajari masalah tersebut. Apa yang perlu Anda ketahui adalah apakah pemahaman tim Anda tentang masalah adalah valid. Cukup sering kita melihat masalah dari sudut pandang kita sendiri, yang berbahaya karena kebanyakan dari kita di dunia teknologi sebenarnya adalah power user dan pada kenyataannya merupakan pengguna minoritas. Kita adalah minoritas vokal dan bisa tertipu saat berpikir sesuatu dapat menjadi masalah padahal tidak.

Ada berbagai metode pengumpulan data untuk memvalidasi tantangan. Masing-masing bergantung pada tim dan jika Anda memiliki akses ke pengguna. Tujuannya adalah untuk mendapatkan pemahaman yang lebih baik dari masalah yang dihadapi.

Wawancara internal dengan para pemangku kepentingan



Wawancara dengan para pemangku kepentingan bisa informatif untuk menemukan wawasan dalam sebuah perusahaan atau tim. Proses wawancara termasuk melakukan wawancara kepada setiap anggota tim dan pemangku kepentingan di perusahaan Anda, dari pemasaran hingga keuangan. Ini akan membantu Anda menemukan apa yang mereka pikir tantangan nyata dan

apa solusi potensial yang bisa mereka pikirkan. Ketika saya mengatakan solusi, saya tidak berbicara tentang solusi teknis di sini, melainkan apa yang bisa menjadi skenario terbaik dan tujuan akhir bagi perusahaan atau produk. Misalnya menggunakan tantangan di atas "memiliki software clubfoot di 80% fasilitas medis hingga akhir tahun ini" dapat menjadi tujuan besar yang menjadi target.

Ada sebuah peringatan. Metode validasi adalah yang paling tidak disukai karena menghambat diskusi dan kolaborasi tim, berpotensi menciptakan suasana tertutup dalam sebuah organisasi. Meskipun demikian, ini bisa menghasilkan beberapa informasi bagus tentang klien dan tantangan desain yang bisa saja Anda lewatkan.

Presentasi kilat



Presentasi kilat adalah presentasi sangat singkat yang hanya berlangsung beberapa menit. Mirip dengan wawancara internal, namun kali ini Anda menghadirkan setiap pemangku kepentingan dalam satu ruangan. Kemudian Anda Memilih lima atau enam orang pemangku kepentingan (pemasaran, penjualan, desain, keuangan, penelitian dll.) untuk berbicara, masing-masing berfokus pada tantangan dari perspektif mereka selama maksimal 10 menit. Topiknya harus mencakup presentasi mereka:

- Tujuan bisnis
- Tantangan proyek dari sudut pandang mereka (ini bisa faktor teknis, pengumpulan penelitian, pembuatan desain dll..)
- Penelitian pengguna yang Anda miliki saat ini

Berikan waktu 5 menit di akhir untuk sesi pertanyaan, dengan orang yang dipilih mencatat semuanya. Setelah selesai, Anda mungkin ingin memperbarui tantangan untuk merefleksikan pembelajaran yang baru. Tujuannya adalah untuk mengumpulkan daftar poin-poin utama yang bisa mendorong fitur atau alur yang membantu Anda mencapai tujuan produk.

Wawancara pengguna



Wawancara pengguna adalah cara yang bagus untuk mempelajari tentang titik derita orang di setiap tugas yang diberikan.

Ini mungkin adalah cara terbaik untuk belajar tentang pengalaman pengguna, titik derita, dan alur. Aturlah setidaknya lima wawancara pengguna, lebih banyak lagi jika Anda memiliki akses kepada mereka. Jenis pertanyaan yang Anda tanyakan kepada mereka harus mencakup:

- Bagaimana mereka menyelesaikan tugas yang ada? Misalnya, Anda ingin menyelesaikan tantangan untuk aplikasi keuangan di atas, Anda bisa bertanya kepada mereka "bagaimana Anda membeli saham dan efek saat ini?"
- Apa yang mereka suka tentang alur ini?
- Apa yang tidak mereka suka tentang alur ini?
- Apa produk sejenis yang saat ini digunakan pengguna?
- Apa yang mereka suka?
- Apa yang tidak mereka suka?
- Jika mereka memiliki tongkat ajaib dan bisa mengubah satu hal tentang proses ini hal apakah itu?

Ide melakukan wawancara adalah agar pengguna berbicara tentang tantangan yang mereka alami. Ini bukanlah poin diskusi untuk Anda, itulah mengapa Anda harus tetap diam. Hal ini semakin benar ketika pengguna berhenti berbicara, selalu berikan waktu sebentar karena mereka bisa saja sedang mengumpulkan pemikirannya. Anda akan terkejut melihat betapa banyak orang yang akan terus berbicara setelah berhenti sejenak selama beberapa detik.

Catat seluruhnya dan jika mungkin rekam percakapan tersebut untuk membantu Anda merekam apa pun yang mungkin Anda lewatkan. Tujuannya adalah membandingkan tantangan terhadap wawasan pengguna yang Anda kumpulkan. Apakah mereka selaras? Apakah Anda mempelajari sesuatu yang membantu memperbarui pernyataan tantangan?

Penelitian bidang etnografi



Melihat pengguna dalam lingkungan alami mereka adalah cara yang bagus untuk memahami bagaimana pengguna mengatasi tantangan mereka sendiri.

Ini adalah bidang tempat Anda mengamati pengguna, dalam konteks saat melakukan sesuatu seperti bagaimana mereka berbelanja, bagaimana mereka melakukan perjalanan ke tempat kerja, bagaimana mereka mengirim pesan SMS dll.. Alasannya adalah karena dalam beberapa kasus orang akan memberi tahu apa yang mereka pikir ingin Anda dengarkan. Namun jika Anda menyaksikan sendiri pengguna melakukan tindakan dan tugasnya, ini bisa menjadi penuh wawasan. Pada dasarnya Anda mengamati tanpa mengganggu, mencatat hal-hal yang mereka rasa mudah atau sulit dan hal-hal yang mungkin mereka lewatkan. Tujuannya adalah untuk melibatkan diri Anda dalam lingkungan pengguna agar lebih berempati dengan titik derita mereka.

Teknik ini biasanya melibatkan beberapa pekerjaan yang dilakukan selama periode waktu yang lebih lama dan membutuhkan peneliti untuk memimpin bagian proyek ini. Namun inilah yang mungkin paling berwawasan karena Anda bisa melihat sekelompok orang yang Anda pelajari di lingkungan alami mereka.

Mengumpulkan semuanya

Setelah Anda menyelesaikan tahap pembelajaran proyek, Anda harus mengambil satu pemeriksaan terakhir pada tantangan Anda. Apakah Anda di jalur yang benar? Apakah ada

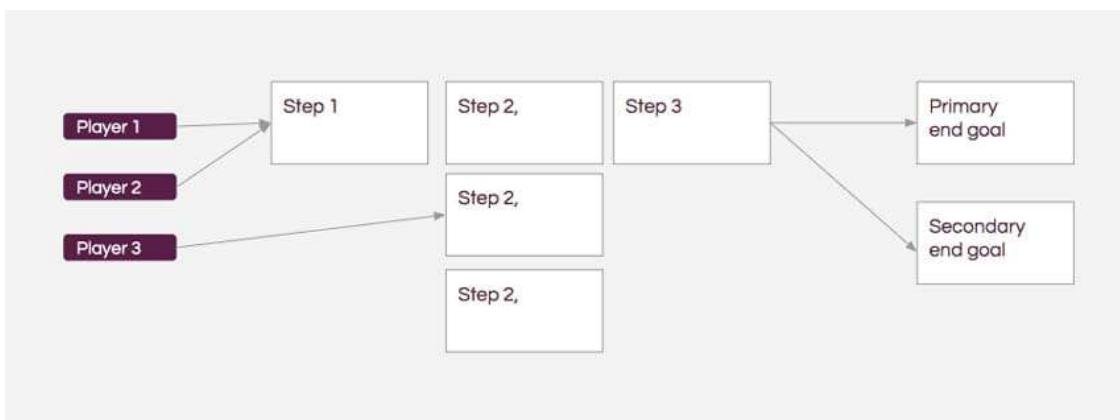
sesuatu yang perlu Anda sesuaikan? Tuliskan semua hal yang telah Anda pelajari dan kelompokkan mereka ke dalam kategori. Ini bisa menjadi dasar dari fitur atau alur, bergantung pada masalah yang Anda selesaikan. Juga bisa digunakan untuk memperbarui dan merevisi tantangan.

Setelah Anda memiliki masukan dan wawasan yang cukup, saatnya untuk menerapkan pengetahuan itu untuk membuat pemetaan proyek.

Pemetaan proyek

Masalah yang coba Anda selesaikan biasanya terdiri dari berbagai tipe orang (atau pemain), masing-masing dengan andil di alur proyek. Berdasarkan pembelajaran, Anda perlu mendaftar para pemain. Ini bisa jadi tipe pengguna atau pemangku kepentingan, misalnya, "dokter yang merawat clubfoot", "pasien yang menderita clubfoot", "perawat yang merawat pasien", dll.. Tuliskan masing-masing pemain di sisi kiri selembar kertas atau tulis pada papan tulis jika Anda memilikiinya. Di sisi sebelah kanan, tuliskan tujuan masing-masing pemain.

Yang terakhir untuk setiap pemain, tuliskan jumlah langkah yang diperlukan untuk mencapai tujuan mereka. Misalnya untuk "dokter yang merawat clubfoot" tujuannya adalah "menyembuhkan pasien yang menderita clubfoot", sehingga langkah-langkahnya adalah "mendaftarkan pasien dalam sistem", "memulai rencana kesehatan", "membuat siklus ulasan kesehatan medis" dan "melakukan prosedur medis".



Pemetaan proyek merencanakan langkah-langkah utama untuk setiap pengguna atau pemain dalam alur.

Hasilnya adalah pemetaan proyek dengan langkah-langkah utama dalam prosesnya. Anggap saja itu sebagai ringkasan proyek tanpa terlalu banyak detail. Ini juga memungkinkan anggota tim menilai apakah pemetaan cocok dengan pernyataan tantangan. Kemudian, ketika Anda memecah setiap langkahnya, akan ada detail lebih lanjut. Namun untuk saat ini, pemetaan proyek memberikan Anda rincian tingkat tinggi dari langkah yang perlu diambil pengguna untuk menyelesaikan tujuan akhir mereka.

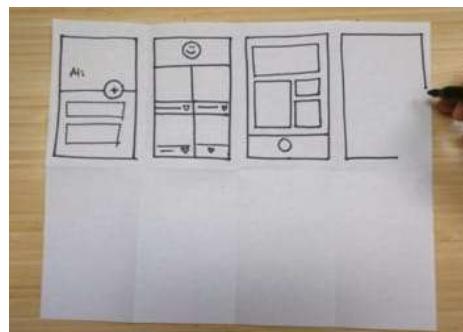
Wireframing dan storyboarding

Crazy 8s

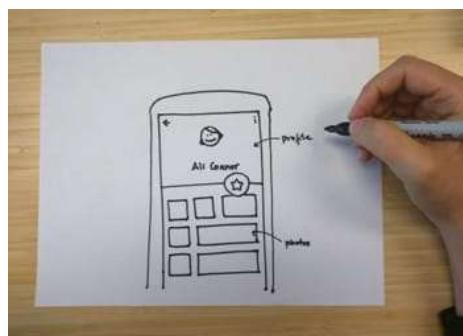
Untuk ini, disarankan metode yang disebut crazy 8s yang meliputi pelipatan kertas dua kali lebih banyak sehingga Anda memiliki delapan panel. Kemudian pada setiap panelnya Anda menggambar sebuah ide berdasarkan semua yang telah Anda pelajari sejauh ini. Berikan diri Anda sepuluh menit agar muncul dengan ide-ide untuk mengisi semua panel yang berjumlah delapan. Jika Anda memberikan diri Anda waktu lebih dari 20 menit, Anda bisa mulai menunda-nunda, pergi membuat kopi, memeriksa email, mengobrol dengan tim Anda dan pada dasarnya menghindari melakukan pekerjaan. Anda ingin menciptakan rasa urgensi dalam langkah ini karena memaksa Anda untuk bekerja dengan cepat dan lebih efektif.

Jika Anda bekerja dengan tim, suruh mereka semua untuk melakukan hal ini juga. Proses ini akan menyentak otak Anda dan membuat Anda berpikir tentang tantangan. Biasanya sketsa akan menjadi wireframe desain antarmuka.

Setelahnya, Anda dan semua orang di tim menyajikan ide-idenya ke kelompok. Setiap orang harus menjelaskan masing-masing delapan ide mereka secara rinci dan mengapa mereka memilih untuk mengambil jalur tersebut. Ingatkan setiap anggota tim untuk menggunakan pembelajaran untuk pemberian ide-ide mereka. Setelah semua orang mengemukakan idenya, saatnya memilih ide-ide tersebut. Setiap orang mendapat dua titik tempelan dan bisa memberikan suara pada ide mana pun. Mereka bisa memberikan kedua suaranya untuk sebuah ide jika mereka benar-benar menyukainya.



Crazy 8s adalah cara yang bagus untuk memasukkan semua ide Anda ke dalam laman.

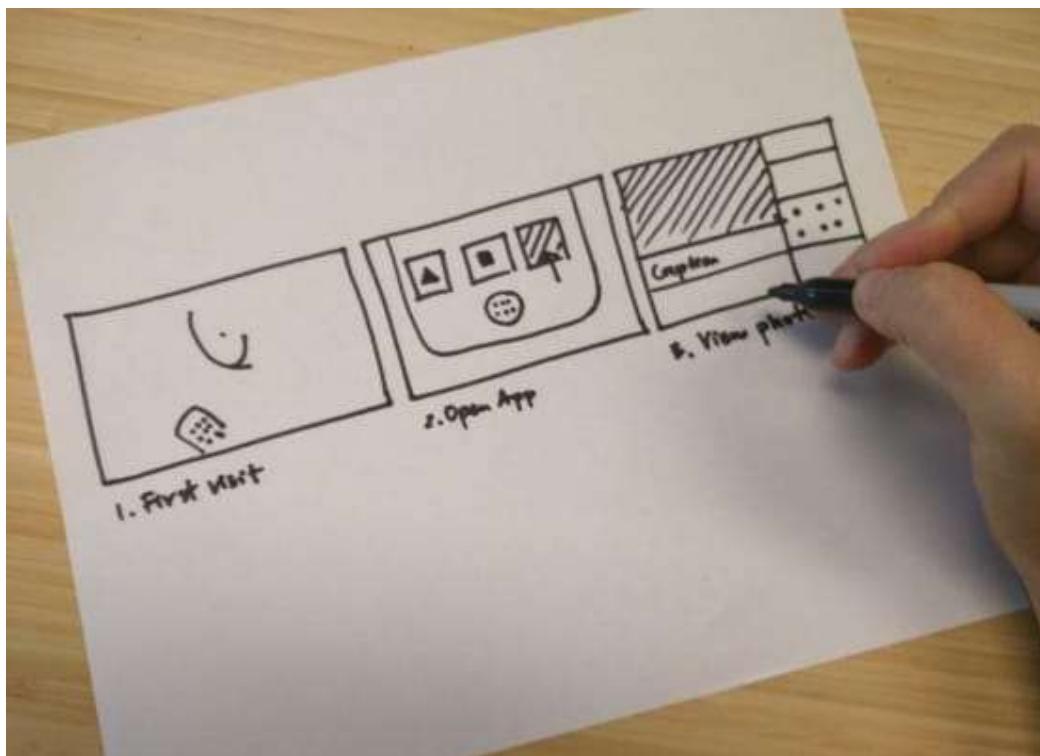


Sekarang Anda harus melakukan desain terperinci berdasarkan apa yang telah Anda pelajari.

Menyaring desain Anda

Setelah pemungutan suara mengambil ide dengan suara terbanyak dan membuat sketsa ide akhir. Anda juga bisa meminjam ide lain yang Anda dengar dari rekan kerja. Berikan diri Anda waktu sepuluh menit untuk menyelesaikan tugas ini. Setelah selesai, presentasikan kembali ide ini ke tim Anda dan lakukan pemungutan suara seperti sebelumnya.

Membuat storyboard ide



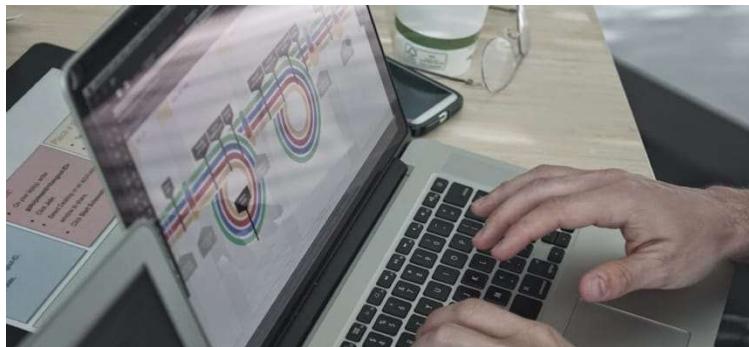
Storyboard melibatkan perpaduan sketsa dan ide Anda ke dalam alur komprehensif.

Dengan desain di tangan, saatnya untuk membuat storyboard interaksi dengan pengguna. Di titik ini Anda sebaiknya sudah berpikir tentang langkah berbeda yang diambil pengguna. Cukup biasa untuk menggabungkan salah satu dari desain rekan Anda ke dalam alur. Anda memerlukan proses langkah demi langkah yang jelas dengan beberapa titik di mana pengguna mungkin berbeda. Lihat kembali pemetaan proyek untuk memvalidasi desain terhadap tujuan Anda.

Membuat prototipe

Membuat prototipe bukan tentang menciptakan potongan kode yang sempurna, namun untuk membuat sesuatu yang bisa dipercaya bila digunakan oleh seseorang. Alat yang digunakan untuk membuat prototipe berbeda dari orang ke orang. Beberapa alat seperti Keynote atau Powerpoint karena memaksa Anda untuk memikirkan alur dan tidak merancang detail. Anda mungkin ingin meluangkan waktu untuk mempelajari alat seperti Balsamiq, Marvel atau Framer yang bisa memberikan kontrol perilaku yang lebih banyak. Apapun alat (bantu) yang

Anda gunakan pastikan itu adalah alat yang membuat Anda fokus pada alur dan terlihat nyata. Anda harus menguji prototipe pada pengguna yang nyata sehingga sebisa mungkin dapat dipercaya tapi pada saat yang bersamaan tidak memerlukan berminggu-minggu jam kerja untuk dibuat.



Prototipe harus cukup nyata untuk bisa dipercaya.

Membuat prototipe adalah keseimbangan antara waktu dan realitas, jadi berhati-hatilah agar tidak melenceng ke salah satu sisi secara ekstrim. Bila tidak, waktu Anda bisa saja terbuang percuma.

Pengujian-kegunaan desain Anda

Akan bagus sekali jika Anda memiliki lab pengujian. Bila tidak, membuat lab tidak sulit asalkan Anda memperhatikan pembuatan lingkungan yang nyaman bagi pengguna serta tidak mengganggu mereka. Pengujian biasanya melibatkan pengguna dan dua orang dari tim Anda, satu mencatat dan lainnya mengajukan pertanyaan. Persiapan yang baik adalah dengan menggunakan aplikasi seperti Hangouts dan merekam tindakannya, ini juga berguna jika Anda menginginkan seluruh tim untuk mengamati dari ruangan yang berbeda. Hal ini cukup menakutkan bagi kami sebagai pembuat aplikasi untuk melakukannya saat kami melihat desain kami keluar di alam liar. Ini bisa menjadi pengalaman yang menyegarkan dan menenangkan.



Storyboard termasuk menempatkan semua sketsa dan ide bersama-sama ke dalam alur yang komprehensif.

Pertanyaan untuk ditanyakan

Saat menguji desain Anda, minta pengguna untuk melakukan tugas di aplikasi dan minta mereka agar berbicara dengan suara keras serta mengungkapkan apa yang mereka lakukan dan mengapa. Ini mungkin terdengar aneh dilakukan, namun hal ini membantu Anda mengetahui apa yang mereka pikirkan. Cobalah untuk tidak mengganggu atau memberi tahu mereka apa yang harus dilakukan saat mereka terhenti. Cukup tanyakan kepada pengguna mengapa mereka mengambil alur tertentu setelah mereka menyelesaikan (atau TIDAK menyelesaikan).

Apa yang perlu Anda ketahui:

- Apa yang mereka suka dari prototipe?
- Apa yang mereka tidak suka dari prototipe?
- Apa saja titik deritanya?
- Mengapa alur bekerja
- Mengapa alur tidak bekerja
- Apa yang ingin mereka tingkatkan?
- Apakah keseluruhan desain/alur memenuhi kebutuhan mereka?

Mengunjungi kembali desain dan rentetan pengujian lagi

Anda memiliki prototipe yang bekerja dengan masukan. Sekarang saatnya merevisi desain Anda, dan menganalisis apa yang berhasil dan apa yang tidak. Jangan takut untuk membuat storyboard wireframe yang benar-benar baru dan membuat prototipe baru. Memulai lagi dari awal bisa membuat alur yang lebih baik dibandingkan mencoba untuk memindahkan sesuatu pada prototipe Anda sebelumnya. Cobalah agar jangan terlalu sayang karena itu hanyalah prototipe.

Setelah puas dengan desain, Anda bisa mengujinya lagi dan menyempurnakannya lagi. Dalam kasus di mana prototipe sama sekali tidak mencapai target, Anda mungkin berpikir proyek itu gagal. Nyatanya, tidak. Anda mungkin menghabiskan waktu development lebih sedikit dibandingkan jika Anda telah membangun desain dan mengetahui lebih banyak tentang apa yang benar-benar disukai pengguna. Dengan design sprints, kami memiliki filosofi yaitu Anda menang atau Anda belajar, jadi jangan terlalu menyalahkan diri sendiri jika ide tersebut tidak bekerja seperti yang direncanakan.

Buatlah!

Anda telah menguji ide. Pengguna menyukainya. Pemangku kepentingan berinvestasi karena mereka telah terlibat sejak awal. Sekarang saat yang tepat untuk membuatnya. Sekarang, Anda harus memiliki gagasan yang jelas tentang apa yang perlu dilakukan dan apa prioritas dari pengalaman ini. Pada setiap tonggak bersejarah proyek, Anda mungkin ingin memperkenalkan pengujian kegunaan untuk membantu memvalidasi pekerjaan dan menjaga Anda tetap di jalur.

Saya tidak bisa menekankan betapa pentingnya mencari tahu sebanyak mungkin informasi sebelum Anda berkomitmen untuk banyak pekerjaan, waktu dan energi pada sesuatu yang mungkin saja tidak menjadi solusi yang tepat.

Artikel ini seharusnya bisa memberikan landasan dasar bagi Anda tentang UX dan arti pentingnya. UX bukanlah sesuatu yang harus dipandang sebagai peran seorang desainer atau peneliti. Ini sebenarnya adalah tanggung jawab semua orang yang terlibat dalam proyek sehingga saya selalu merekomendasikan keterlibatan dalam setiap kesempatan.

BAB III STUDI KASUS

Sasaran Pembelajaran

Mahasiswa mampu secara khusus menjelaskan dan memahami obyek yang diteliti secara khusus sebagai suatu 'kasus'

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer

Keterkaitan bahan pembelajaran dengan pokok bahasan lainnya

Materi ini memperkenalkan alur kerja yang bisa membantu tim, produk, startup dan perusahaan membuat proses yang kuat dan valuable untuk mengembangkan UX yang lebih baik bagi para pelanggan.

Manfaat atau pentingnya bahan pembelajaran ini

Mahasiswa dapat mencapai penyesuaian diri yang lebih baik

Apa Yang Membuat Sebuah Situs Seluler Bagus?

Pengguna seluler sangat berorientasi pada tujuan. Mereka berharap bisa mendapatkan apa yang mereka butuhkan, dengan segera, dan dengan cara mereka sendiri.

Penelitian ini berlangsung selama 119 jam per-orang dengan partisipan dari AS. Para partisipan diminta untuk melakukan tugas-tugas kunci di berbagai situs seluler. Termasuk pengguna iOS dan Android, dan pengguna menguji situs tersebut di ponsel mereka sendiri. Untuk setiap situs, para partisipan diminta untuk menyuarakan pemikiran mereka dengan keras karena mereka menyelesaikan tugas-tugas yang berfokus pada konversi seperti melakukan pembelian atau pemesanan reservasi.

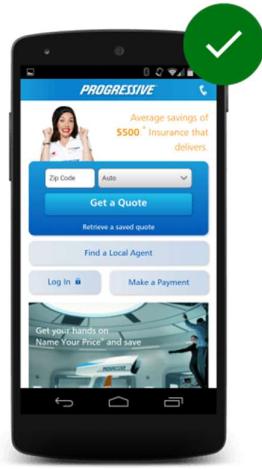
Penelitian ini menemukan 25 prinsip desain situs seluler, dikelompokkan ke dalam lima kategori.

Navigasi situs dan laman Beranda

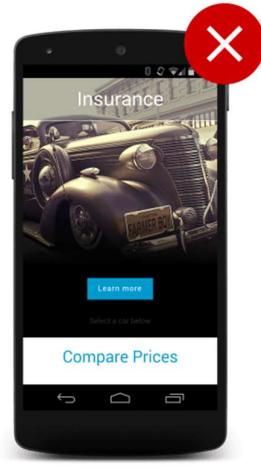
Indikator berhasil: Fokuskan beranda seluler Anda pada cara menghubungkan pengguna ke materi yang mereka cari.

Pertahankan panggilan untuk aksi di depan dan tengah

Menyediakan tugas sekunder melalui menu atau "paro bawah" (bagian dari laman web yang tidak bisa dilihat tanpa gulir ke bawah).



LAKUKAN: Memudahkan ketersediaan semua tugas yang paling sering dipakai pengguna.



JANGAN: Membuang ruang berharga paro-atas dengan panggil-untuk-aksi tidak jelas seperti "ketahui selengkapnya".

Pertahankan menu yang singkat dan manis



LAKUKAN: Pertahankan menu yang singkat dan manis.

Pengguna seluler tidak memiliki kesabaran untuk meng gulir melalui daftar panjang opsi untuk menemukan apa yang mereka inginkan. Tata ulang menu Anda agar menggunakan item sesedikit mungkin, tanpa harus mengorbankan kegunaan.

Mempermudah cara kembali ke laman beranda

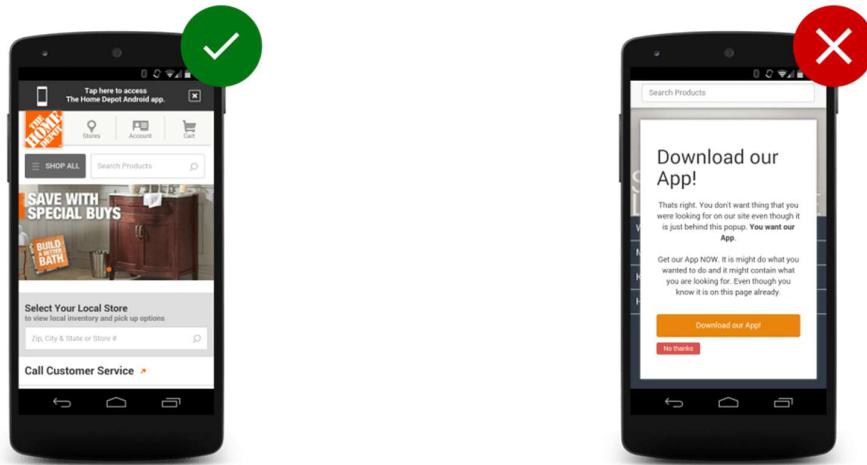


LAKUKAN: Mempermudah cara kembali ke laman beranda.

Pengguna ingin kembali ke beranda ketika mereka mengetuk logo di sudut kiri atas laman seluler, dan mereka bisa frustrasi bila tidak tersedia atau tidak bekerja.

Jangan biarkan promosi mencuri perhatian

Pengantara pemasangan aplikasi besar (mis. promosi selayar-penuh yang menyembunyikan materi dan meminta pengguna untuk memasang aplikasi) menjengkelkan pengguna dan mempersulit saat melakukan tugas. Selain menjengkelkan pengguna, situs yang menggunakan pengantara pemasangan aplikasi tidak akan lolos Uji Ramah Google Seluler, yang bisa berdampak negatif terhadap peringkat penelusuran mereka.



LAKUKAN: Promosi harus mudah ditutup dan tidak mengganggu pengalaman pengguna.

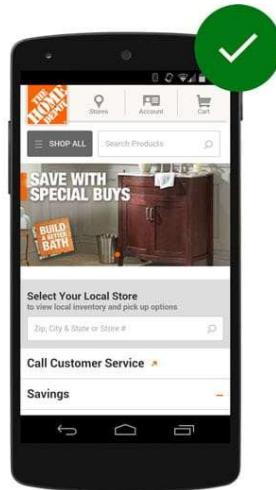
JANGAN: Pengantara (kadang-kadang disebut membanting pintu) sering menjengkelkan pengguna dan membuat menggunakan situs adalah sebuah penderitaan.

Penelusuran situs

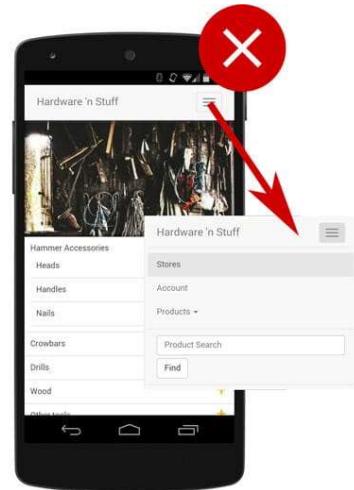
Indikator berhasil: Membantu pengguna seluler menemukan apa yang mereka cari dengan sangat cepat.

Membuat penelusuran situs terlihat

Pengguna yang mencari informasi biasanya membuka penelusuran, sehingga bidang penelusuran harus menjadi salah satu item utama yang mereka lihat di laman Anda. Jangan menyembunyikan kotak telusur di menu.



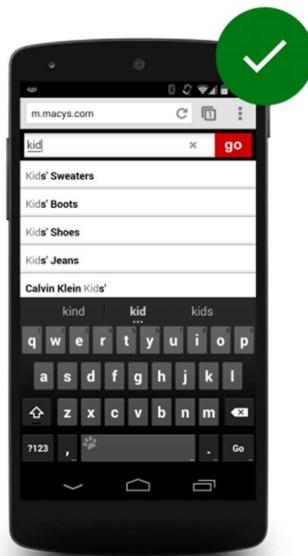
LAKUKAN: Membuat penelusuran terlihat



JANGAN: Menyembunyikan penelusuran di menu luapan

Pastikan hasil penelusuran situs relevan

Pengguna tidak memindai beberapa laman dari hasil penelusuran untuk menemukan apa yang mereka cari. Permudah pengguna dengan menyelesaikan-otomatis kueri, mengoreksi kesalahan eja, dan menyarankan kueri terkait. Daripada menciptakan kembali sesuatu yang sudah ada, pertimbangkan produk yang kuat seperti Google Penelusuran Khusus.



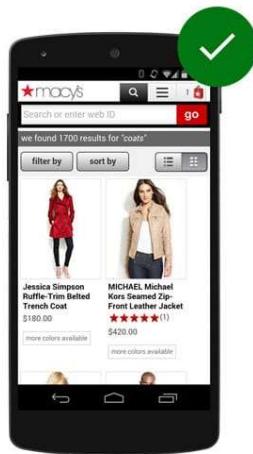
LAKUKAN: Macy hanya mengembalikan barang anak-anak (kids).



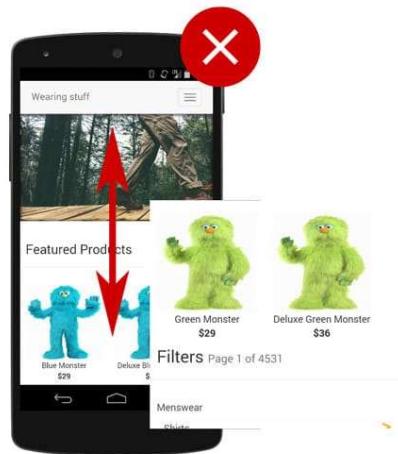
JANGAN: Mengembalikan hasil dengan kata anak (kid) di dalamnya.

Mengimplementasikan filter untuk mempersempit hasil

Partisipan penelitian mengandalkan filter untuk menemukan apa yang mereka cari, dan meninggalkan situs yang tidak memiliki filter yang efektif. Menempatkan filter di atas hasil penelusuran, dan membantu pengguna dengan menampilkan berapa banyak hasil yang dikembalikan ketika filter tertentu diterapkan.

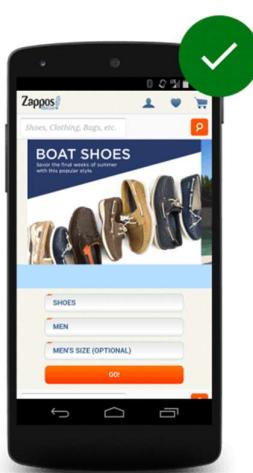


LAKUKAN: Permudah untuk memberi filter.



JANGAN: Menyembunyikan fungsionalitas filter.

Memandu pengguna agar hasil penelusuran situs lebih baik



LAKUKAN: Membantu pengguna untuk menemukan apa yang mereka cari dengan memandu mereka ke arah yang tepat.

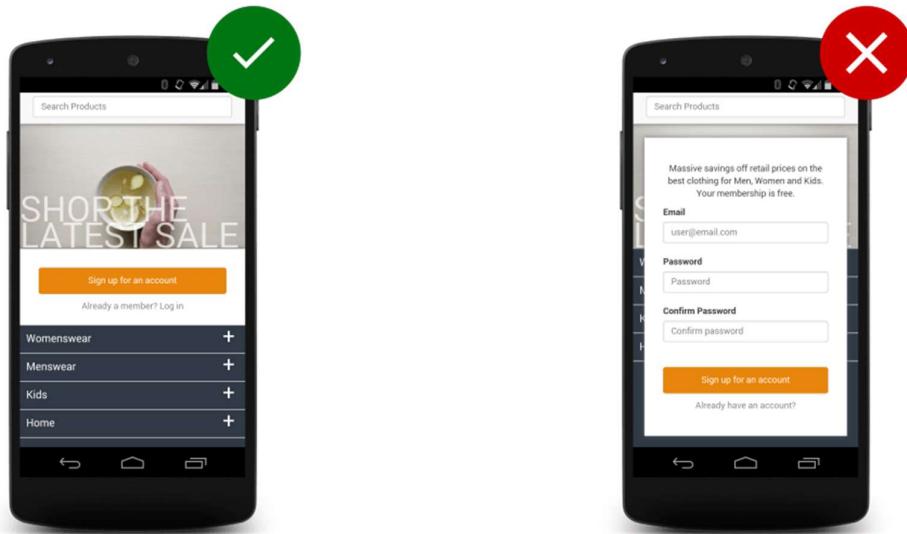
Untuk situs dengan segmen pengguna yang beragam, ajukan beberapa pertanyaan sebelum menyajikan kotak telusur, dan menggunakan respons pengguna sebagai filter kueri penelusuran untuk memastikan bahwa pengguna mendapatkan hasil dari segmen yang paling relevan.

Niaga dan konversi

Indikator berhasil: Memahami perjalanan pelanggan Anda dan membiarkan pengguna melakukan konversi dengan cara mereka sendiri.

Biarkan pengguna menjelajahi sebelum mereka berkomitmen

Partisipan penelitian frustrasi oleh situs yang mengharuskan pendaftaran di awal untuk melihat situs, terutama ketika mereka tersebut masih terdengar asing. Meskipun informasi pelanggan mungkin integral untuk bisnis Anda, memintanya terlalu dini bisa menyebabkan pendaftaran yang lebih sedikit.

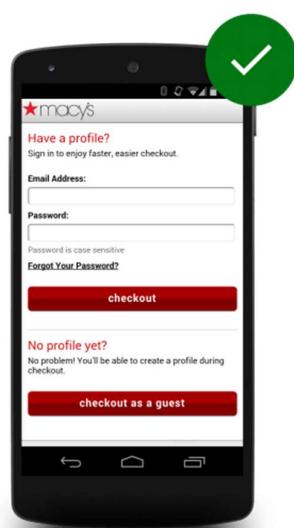


LAKUKAN: Izinkan pengguna menjelajahi situs tanpa harus mendaftar masuk.

Izinkan pengguna membeli sebagai tamu

JANGAN: Menempatkan login atau registrasi terlalu awal dalam sebuah situs.

LAKUKAN: Izinkan pengguna membeli dengan akun tamu.



Partisipan penelitian menganggap checkout tamu "nyaman", "sederhana", "mudah", dan "cepat". Pengguna kesal oleh situs yang memaksa mereka mendaftarkan sebuah akun saat melakukan pembelian, terutama ketika manfaat dari akun tidak jelas.

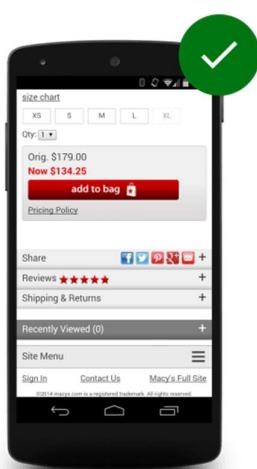
Menggunakan informasi yang ada untuk memaksimalkan kemudahan

Ingat dan pra-isi preferensi untuk pengguna terdaftar. Tawarkan layanan checkout pihak ketiga yang familiar untuk pengguna baru.

Menggunakan tombol click-to-call untuk tugas yang kompleks

Pada perangkat dengan kemampuan menelepon, tautan click-to-call memungkinkan pengguna untuk melakukan panggilan telepon hanya dengan mengetuk tautan. Pada kebanyakan perangkat seluler, pengguna menerima konfirmasi sebelum nomor dihubungi, atau tampil menu yang akan menanyakan pengguna bagaimana sebaiknya nomor ditangani.

Buatlah mudah untuk diselesaikan pada perangkat lain



LAKUKAN: Berikan cara mudah bagi pengguna untuk melanjutkan browsing atau berbelanja di perangkat lain.

Pengguna sering kali ingin menyelesaikan tugas pada perangkat lain. Misalnya, mereka mungkin ingin menampilkan item pada layar yang lebih besar. Atau mereka mungkin sedang sibuk dan harus menyelesaikannya nanti. Dukung perjalanan pelanggan ini dengan memungkinkan pengguna untuk berbagi item di jaringan sosial, atau dengan memperbolehkan pengguna meng-email sendiri tautan secara langsung dari dalam situs.

Entri formulir

Indikator berhasil: Sediakan pengalaman konversi yang mulus dan tanpa friksi dengan formulir yang bisa dipakai.

Merampingkan entri informasi

Secara otomatis maju ke bidang berikutnya ketika pengguna menekan Return. Secara umum, semakin sedikit pengguna melakukan ketukan, semakin baik.

Memilih masukan yang paling sederhana

Gunakan tipe masukan yang paling tepat untuk setiap skenario. Gunakan elemen seperti datalist untuk menyediakan nilai yang disarankan untuk bidang.

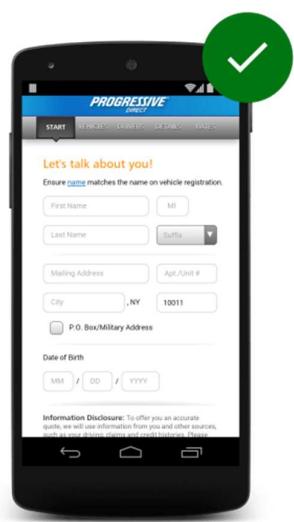
Menyediakan kalender visual untuk pemilihan tanggal



LAKUKAN: gunakan widget kalender jika memungkinkan.

Beri label tanggal awal dan akhir dengan jelas. Pengguna sebaiknya tidak perlu meninggalkan situs dan memeriksa aplikasi kalender hanya untuk menjadwalkan suatu tanggal.

Meminimalkan kesalahan formulir dengan label dan validasi real-time



LAKUKAN: Pra-isi materi apabila memungkinkan.

Labeli input dengan benar dan validasi input secara real-time.

Mendesain formulir efisien

Manfaatkan isiotomatis sehingga pengguna bisa dengan mudah melengkapi formulir dengan data pra-isi. Pra-isi bidang dengan informasi yang sudah Anda tahu. Misalnya, ketika mengambil alamat pengiriman dan penagihan, cobalah untuk menggunakan `requestAutocomplete` atau memperbolehkan pengguna menyalin alamat pengiriman ke alamat penagihan mereka (atau sebaliknya).

Kegunaan dan faktor form

Indikator berhasil: Menyenangkan pengguna seluler dengan hal-hal kecil yang meningkatkan pengalaman mereka.

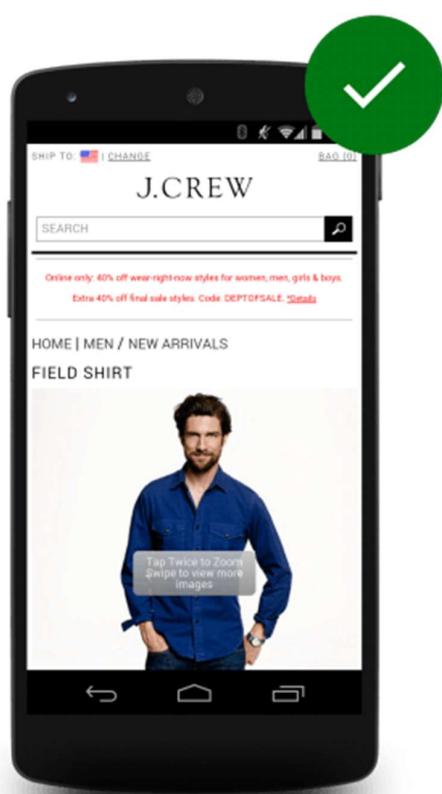
Mengoptimalkan seluruh situs Anda untuk perangkat seluler

Menggunakan layout responsif yang bisa berubah berdasarkan ukuran dan kemampuan perangkat pengguna. Partisipan penelitian menemukan bahwa situs dengan campuran laman yang dioptimalkan untuk seluler dan desktop, lebih sulit digunakan dibandingkan situs khusus untuk desktop.

Jangan buat pengguna melakukan pinch-to-zoom

Pengguna merasa nyaman dengan pengguliran situs secara vertikal, namun tidak secara horizontal. Hindari elemen dengan lebar tetap dan besar. Gunakan kueri media CSS untuk menerapkan penataan gaya yang berbeda untuk layar berbeda. Jangan membuat materi yang hanya terlihat baik pada lebar tampilan yang terlihat tertentu. Situs yang memaksa pengguna untuk gulir secara horizontal tidak akan lolos Uji Ramah-Google Seluler, yang bisa berdampak negatif terhadap peringkat penelusuran.

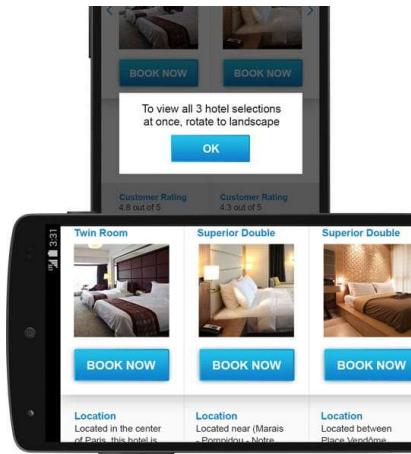
Membuat gambar produk yang bisa diperbesar



LAKUKAN: Membuat gambar produk yang bisa diperbesar sehingga mudah untuk melihat detailnya.

Pelanggan retail berharap situs mengizinkan mereka melihat tampilan dekat resolusi tinggi dari produk. Partisipan penelitian merasa kecewa ketika mereka tidak bisa melihat apa yang mereka beli.

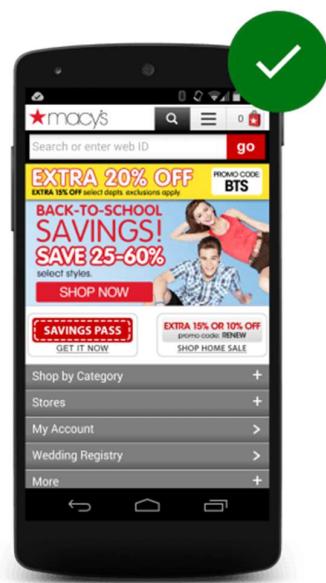
Memberi tahu pengguna orientasi yang terbaik



LAKUKAN: Memberi tahu pengguna orientasi yang terbaik.

Partisipan penelitian biasanya tetap menggunakan orientasi layar yang sama sampai sesuatu mendorong mereka untuk beralih. Desain untuk mode lanskap dan potret, atau dorong pengguna agar beralih ke orientasi optimal. Pastikan bahwa panggil-untuk-aksi yang penting bisa diselesaikan bahkan jika pengguna mengabaikan saran untuk beralih orientasi.

Menjaga pengguna tetap di satu jendela browser



LAKUKAN: Macy mempertahankan pengguna di situs mereka dengan memberikan kupon di situs.

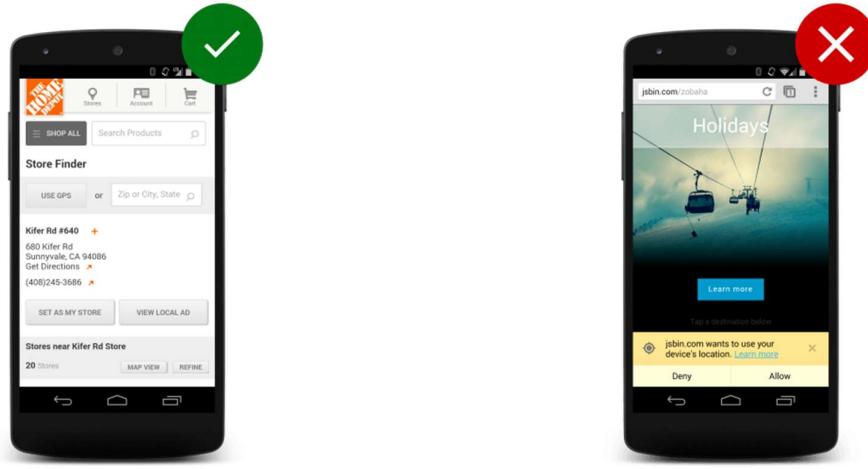
Pengguna mungkin mengalami kesulitan saat beralih antar jendela dan mungkin tidak dapat menemukan jalan kembali ke situs. Hindari panggil-untuk-aksi yang membuka jendela baru. Identifikasi setiap proses yang mungkin menyebabkan pengguna mencari di luar situs dan menyediakan fitur agar mereka tetap berada di situs Anda. Misalnya, jika Anda menerima kupon, langsung tawarkan kepada mereka di situs, bukannya memaksa pengguna mencari penawaran di situs lainnya.

Hindari pelabelan "situs lengkap"

Ketika partisipan penelitian melihat opsi untuk "situs lengkap" (situs desktop) versus "situs seluler", mereka berpikir bahwa situs seluler kekurangan materi dan memilih "lengkap", yang mengarahkan mereka ke situs desktop.

Jelaskan mengapa Anda memerlukan lokasi pengguna

Pengguna harus selalu mengerti alasan Anda meminta lokasi mereka. Partisipan penelitian yang mencoba untuk memesan hotel di kota lain menjadi bingung ketika sebuah situs perjalanan mendeteksi lokasi mereka dan menawarkan hotel di kota mereka yang sekarang ini. Biarkan bidang lokasi kosong secara default, dan izinkan pengguna untuk mengisinya melalui panggil-untuk-aksi yang jelas seperti "Find Near Me".



LAKUKAN: Selalu minta akses ke lokasi berdasarkan isyarat pengguna.

JANGAN: Memintanya secara langsung di beranda saat situs sedang memuat akan menjadikan pengalaman pengguna yang buruk.

BAB IV VARIABEL FONT

Sasaran Pembelajaran

Mahasiswa mampu mengimplementasikan variable font dalam membuat program maupun aplikasi web dan mobile

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer

Keterkaitan bahan pembelajaran dengan pokok bahasan lainnya

Pada materi ini, kita akan melihat seperti apa yang disebut font, bagaimana kita dapat menggunakannya dalam pekerjaan kita, dan potensi yang dimiliki font

Manfaat atau pentingnya bahan pembelajaran ini

Memberi pemahaman dasar tentang font dalam pembuatan program maupun aplikasi web dan mobile

Petunjuk belajar

Untuk memahami penggunaan dan potensi yang dimiliki font, pertama-tama harus mengeksplorasi bagaimana tipografi dan pemuatan font bekerja dalam laman web

Sebagaimana diketahui bahwa font memberikan visualisasi dari huruf sehingga kita bisa mengenali sebuah huruf untuk dirangkai menjadi sebuah kata dan kalimat. Dalam UI UX penggunaan font yang tepat, jelas dan mudah dibaca dapat membuat pengguna merasa nyaman membaca informasi yang disajikan, sehingga dapat mendukung dalam pemberian pengalaman pengguna.

Istilah Font dan jenis huruf digunakan secara bergantian di industri pengembang. Namun ada perbedaan antara jenis huruf dan font itu sendiri: jenis huruf termasuk seluruh keluarga desain, seperti jenis huruf Roboto. Sementara font adalah salah satu file digital dari keluarga itu, seperti "Roboto Bold" atau "Roboto Italic." Dengan kata lain, jenis huruf adalah apa yang Anda lihat, dan font adalah apa yang Anda gunakan.

Sebagai contoh font adalah Roboto, font ini dirancang dan dikembangkan oleh Christian Robertson. Font yang termasuk dalam keluarga Roboto adalah sebagai berikut:

Roboto

SUNGLASSES

Self-driving robot ice cream truck
Fudgesicles only 25¢

ICE CREAM

Marshmallows & almonds

#9876543210

Music around the block

Summer heat rising up from the boardwalk

Thin

Thin Italic

Light

Light Italic

Regular

Regular Italic

Medium

Medium Italic

Bold

Bold Italic

Black

Black Italic

Tantangan untuk Desainer dan Pengembang

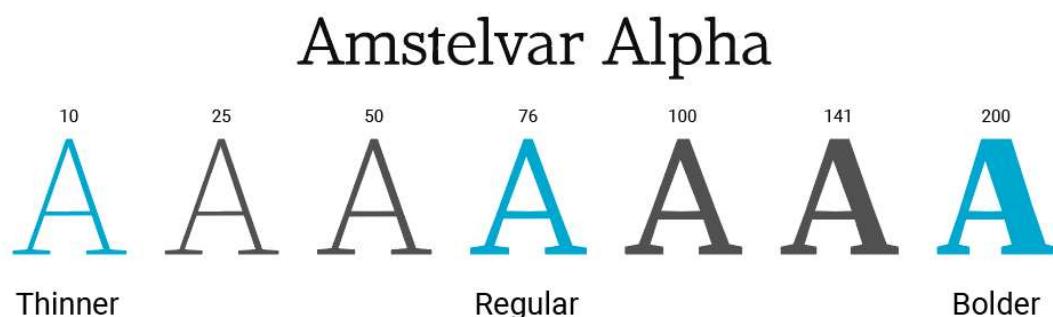
Ketika seorang desainer grafis menyiapkan karya mereka, mereka biasanya mengekspor karya seni terakhir dengan cara yang mananamkan semua font yang digunakan, atau mereka menyediakan arsip ekspor yang berisi semua asset yang digunakan secara terpisah, sedikit seperti situs web dasar. Biaya untuk perancang tergantung pada kesesuaian jenis huruf untuk konteks render (misalnya, dapat dibaca pada ukuran kecil) dan lisensi untuk menggunakan font.

Di web, kami harus mempertimbangkan kedua aspek tersebut, ditambah biaya bandwidth yang terkait. Untuk setiap anggota keluarga jenis huruf yang digunakan dalam desain kami, kami harus meminta file font lain untuk diunduh oleh pengguna kami sebelum mereka dapat melihat teks itu. Hanya termasuk gaya Reguler dan Tebal, ditambah rekan italic-nya, dapat berjumlah hingga 500rb atau lebih data. Ini telah menjadi poin penting bagi para desainer dan pengembang web, karena pengalaman tipografi yang lebih kaya akan dikenakan biaya. Ini bahkan sebelum kita membahas bagaimana font ditampilkan, dan pola mundur atau pemuatan tertunda yang akan kita gunakan (seperti "FOIT" dan "FOUT").)

Anatomi variable font

Font variabel adalah kumpulan gaya master, dengan satu master 'default' sentral (biasanya gaya font Reguler) dan beberapa "sumbu" terdaftar yang mengikat master pusat ke master lainnya. Sebagai contoh, sumbu Bobot mungkin menghubungkan master gaya Ringan ke gaya default dan melalui ke master gaya Bold. Gaya individu yang dapat ditemukan di sepanjang sumbu ini disebut instance.

Sebagai contoh, font variabel Amstelvar memiliki tiga master untuk sumbu Bobotnya: Master reguler berada di tengah, dan dua master, lebih tipis dan lebih tebal, berada di ujung yang berlawanan dari sumbu. Di antara ini ada 200 contoh yang berpotensi yang dapat dipilih oleh perancang atau pengembang:



Typeface Amstelvar, dirancang oleh David Berlow, tipe designer dan typographer di Font Bureau.

Spesifikasi OpenType menentukan sumbu lain, seperti Lebar, Ukuran Optik, Miring dan Miring. Semua sumbu ini berbagi master default yang sama, dan kita dapat menggabungkan mereka untuk menjelajahi sejumlah gaya tipografi eksponensial, seperti kekuatan 2, 3 dan 4 lakukan untuk angka.

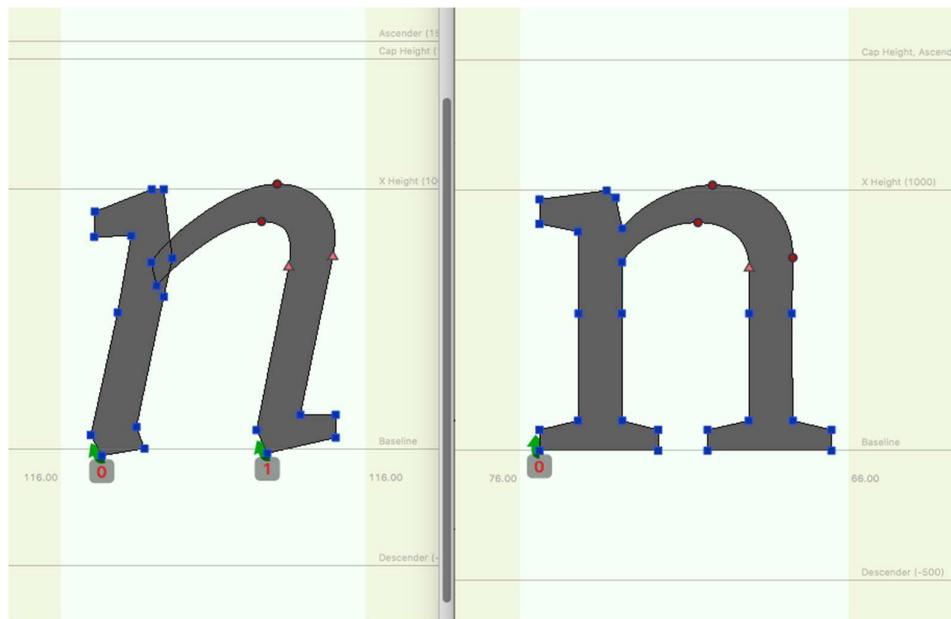
Amstelvar juga memiliki tiga master dalam sumbu Lebar: Sekali lagi Regular berada di tengah sumbu, dan dua master, sempit dan lebih lebar, berada di ujung yang berlawanan dari sumbu ini. Ini tidak hanya menyediakan semua lebar gaya Reguler, tetapi juga semua lebar dan bobot digabungkan.

Di antara kapak terdaftar Amstelvar (lebar, berat, dan ukuran optik) ada ribuan gaya. Ini mungkin tampak seperti pembunuhan besar-besaran, tetapi pertimbangkan bahwa Amstelvar hanya mendukung sistem penulisan Latin. Mempertimbangkan kebutuhan semua skrip dunia, dan banyak aplikasi tipografi saat ini, kualitas pengalaman membaca dapat sangat ditingkatkan oleh keragaman jenis gaya dalam font. Dan, jika tanpa penalti performa, pengguna dapat menggunakan beberapa atau sebanyak gaya yang mereka inginkan - terserah desain mereka.

Italics sedikit berbeda

Cara orang Italia menangani dalam font variabel menarik, karena ada dua pendekatan perbedaan. Jenis-jenis seperti Helvetica atau Roboto memiliki kontur yang kompatibel dengan interpolasi, sehingga gaya Romawi dan Miringnya dapat diinterpolasi antara dan sumbu Miring dapat digunakan untuk beralih dari Romawi ke Miring.

Tipografi lainnya (seperti Garamond, Baskerville, atau Bodoni) memiliki kontur mesin terbang Romawi dan Italia yang tidak kompatibel dengan interpolasi. Misalnya, kontur yang biasanya mendefinisikan huruf kecil Romawi "n" tidak cocok dengan kontur yang digunakan untuk mendefinisikan huruf kecil Italia "n". Alih-alih menginterpolasi satu kontur ke kontur lainnya, sumbu Italic beralih dari kontur Romawi ke kontur Italic.



Kontur “n” Amstelvar dalam bahasa Italia (12 poin, berat reguler, lebar normal), dan dalam bahasa Romawi. Gambar dipasok oleh David Berlow, tipe designer dan typographer di Font Bureau.

Setelah beralih ke Italic, sumbu yang tersedia untuk pengguna harus sama dengan yang untuk Romawi, sama seperti karakter yang ditetapkan harus sama.

Kemampuan substitusi mesin terbang juga dapat dilihat untuk mesin terbang individu, dan digunakan di mana saja di ruang desain font variabel. Misalnya, desain tanda dolar dengan dua bilah vertikal berfungsi paling baik pada ukuran titik yang lebih besar, tetapi pada ukuran titik yang lebih kecil, desain dengan hanya satu bilah lebih baik. Ketika kami memiliki piksel lebih sedikit untuk rendering mesin terbang, desain dua batang dapat menjadi tidak terbaca. Untuk mengatasi hal ini, seperti halnya sumbu Italic, substitusi mesin terbang dari satu mesin terbang ke mesin terbang lain dapat terjadi di sepanjang sumbu Ukuran Optik pada titik yang ditentukan oleh perancangan tipe.

Singkatnya, di mana kontur memungkinkan untuk itu, desainer tipe dapat membuat font yang interpolasi antara berbagai master dalam ruang desain multi-dimensi. Ini memberi Anda kontrol granular atas tipografi Anda, dan banyak kekuatan.

Definisi axis

Karena pengembang font menentukan sumbu mana yang tersedia dalam font mereka, penting untuk memeriksa dokumentasi font untuk mengetahui apa yang tersedia. Misalnya, dalam font variabel Gingham yang dirancang oleh Christoph Koeberlin, ada dua sumbu yang tersedia, Lebar dan Berat. Font variabel Amstelvar tidak mengandung sumbu miring, tetapi memang memiliki sumbu yang disebut Grade, ditambah banyak sumbu lainnya.

Sumbu Grade menarik karena mengubah bobot font tanpa mengubah lebar, sehingga jeda baris tidak berubah. Dengan bermain dengan sumbu Grade, Anda dapat menghindari dipaksa untuk bermain-main dengan perubahan pada sumbu Bobot yang mempengaruhi lebar keseluruhan, dan kemudian berubah ke poros Lebar yang mempengaruhi bobot keseluruhan. Ini dimungkinkan karena gaya default Amstelvar telah didekonstruksi dalam 4 aspek dasar bentuk: bentuk hitam atau positif, bentuk putih atau negatif, dan dimensi x dan y. Keempat aspek ini dapat dicampur untuk membentuk gaya lain, seperti Lebar dan Berat, dengan cara warna primer dapat dicampur untuk membuat warna lain.

Amstelvar Alpha

AmstelvarAlpha-Variations.ttf is an [exploratory OpenType 1.8 font](#) made with a combination of Font Bureau's python font tools and the fontmake tool developed by Google.

“The quick brown fox jumps over the lazy dog.”

Anda dapat melihat contoh kerja dan kode sumber untuk contoh di atas di sini. Lima axis terdaftar plus Kelas memiliki tag 4 karakter yang digunakan untuk menetapkan nilainya dalam CSS:

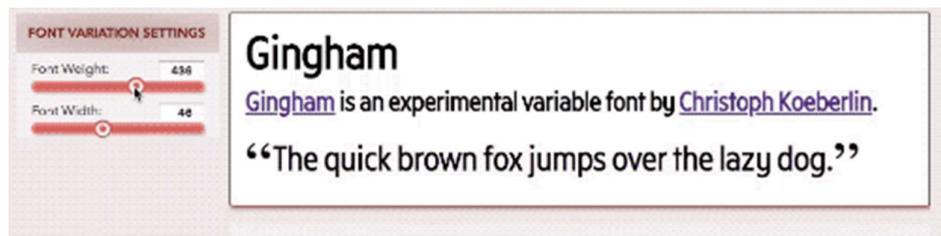
Nama-nama Axis dan CSS values	
Weight	wght
Width	wdth
Slant	slnt
Optical Size	opsz
Italics	ital
Grade	GRAD

Untuk menambahkan font variabel terlebih dahulu, kita harus menautkannya, seperti font kustom apa pun:

```
@font-face {  
    font-family: 'AmstelvarAlpha';  
    src: url('../fonts/AmstelvarAlpha-VF.ttf');  
}
```

Cara kita mendefinisikan nilai sumbu adalah dengan menggunakan variasi font properti CSS yang memiliki serangkaian nilai yang memasangkan tag sumbu dengan lokasi contoh:

```
#font-amstelvar {  
    font-family: 'AmstelvarAlpha';  
    font-variation-settings: 'wdth' 400, 'wght' 98;  
}
```



Dalam contoh ini Anda dapat melihat sumbu Bobot dan Lebar diubah dengan cepat. Anda dapat melihat contoh kerja dan kode sumber untuk contoh di atas di sini.

Tanggung jawab penata letak

Menetapkan nilai sumbu diturunkan ke selera pribadi dan menerapkan praktik tipografi terbaik. Bahaya dengan teknologi baru apa pun adalah kemungkinan penyalahgunaan, dan pengaturan yang terlalu artistik atau eksploratif juga dapat mengurangi keterbacaan teks yang sebenarnya. Untuk judul, menjelajahi sumbu yang berbeda untuk membuat desain artistik yang hebat memang mengasyikkan, tetapi untuk tubuh menyalin risiko ini membuat teks menjadi tidak terbaca.



Salah satu contoh yang bagus dari ekspresi artistik ditunjukkan di atas, sebuah eksplorasi dari jenis huruf Decovar oleh Mandy Michael. Anda dapat melihat contoh kerja dan kode sumber untuk contoh di atas di sini.



Ada juga kemungkinan untuk mengeksplorasi karakter animasi dengan font variabel. Di atas adalah contoh dari berbagai sumbu yang digunakan dengan jenis huruf Zycon. Lihat contoh animasi langsung di Axis Praxis.

Variable fonts performance gains

Font variabel OpenType memungkinkan kita untuk menyimpan beberapa variasi tipe keluarga ke dalam file font tunggal. Monotype menjalankan eksperimen dengan menggabungkan 12 font input untuk menghasilkan delapan bobot, melintasi tiga lebar, melintasi gaya Italia dan

Romawi. Menyimpan 48 font individual dalam file font variabel tunggal berarti pengurangan 88% dalam ukuran file.

Di sisi lain, jika Anda menghidupkan font di antara pengaturan, ini dapat menyebabkan masalah kinerja browser. Pelajari lebih lanjut tentang ini di Surma's Supercharged.

Dengan fonta variabel, pembuat aplikasi dan situs web dapat menawarkan pengalaman tipografi yang sangat kaya yang mengekspresikan setiap merek, tanpa bandwidth dan biaya latensi sebelumnya. Namun, jika Anda menggunakan font tunggal seperti Roboto Regular dan tidak ada yang lain, Anda mungkin melihat keuntungan bersih dalam ukuran font jika Anda beralih ke font variabel dengan banyak sumbu. Seperti biasa, itu tergantung pada kasus penggunaan Anda.

Fallbacks dan dukungan browser

Dukungan saat ini terbatas, tetapi font variabel akan berfungsi hari ini di luar kotak di Chrome dan Safari, dengan dukungan segera hadir untuk Edge 17 dan Firefox. Lihat caniuse.com untuk lebih jelasnya. Dimungkinkan untuk menggunakan @support di CSS Anda untuk membuat fallback yang layak:

```
@supports (font-variation-settings: 'wdth' 200) {  
  @font-face {  
    /* https://github.com/TypeNetwork/Amstelvar */  
    font-family: AmstelvarAlpha;  
    src: url('../fonts/AmstelvarAlpha-VF.ttf');  
    font-weight: normal;  
    font-style: normal;  
  }  
  
  #font-amstelvar {  
    font-family: AmstelvarAlpha;  
    font-variation-settings: 'wdth' 400, 'wght' 98;  
  }  
}
```

BAB V AKSESIBILITAS

Sasaran Pembelajaran

Mahasiswa mengetahui apa yang dimaksud aksesibilitas dan cara menerapkannya pada web-development, mengetahui cara membuat situs web yang bisa diakses dan bisa digunakan oleh setiap orang, mengetahui cara menyertakan aksesibilitas dasar dengan dampak minimal pada development, mengetahui apa saja fitur HTML yang tersedia dan cara menggunakannya untuk meningkatkan aksesibilitas, mengetahui tentang berbagai teknik aksesibilitas modern untuk membuat pengalaman aksesibilitas yang brilian

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer

Manfaat atau pentingnya bahan pembelajaran ini

Memberi pemahaman mengenai aksesibilitas

Dengan memahami aksesibilitas, cakupannya, dan dampaknya bisa membuat Anda menjadi web developer yang lebih baik. Panduan ini dimaksudkan untuk membantu Anda memahami cara membuat situs web bisa diakses dan bisa digunakan oleh setiap orang.

"Aksesibilitas" bisa jadi sulit dieja, namun tidak harus sulit dicapai. Dalam panduan ini, Anda akan melihat beberapa cara mudah untuk membantu meningkatkan aksesibilitas dengan upaya minimal, cara menggunakan apa yang ditanamkan ke HTML untuk membuat antarmuka yang lebih bisa diakses dan sempurna, dan cara memanfaatkan sebagian teknik modern untuk memoles pengalaman yang bisa diakses.

Anda juga akan menemukan banyak dari teknik ini yang akan membantu membuat antarmuka yang lebih menyenangkan dan mudah digunakan oleh *semua* pengguna, tidak cuma bagi penyandang cacat.

Tentu saja, banyak developer yang hanya memiliki pemahaman kabur mengenai apa yang dimaksud dengan aksesibilitas — sesuatu yang harus dilakukan pada kontrak pemerintah, daftar periksa, dan pembaca layar, ya? — dan banyak juga yang gagal paham. Misalnya, banyak developer merasa bahwa mengurus aksesibilitas akan memaksa mereka harus memilih antara membuat pengalaman yang menyenangkan serta menarik, dan pengalaman bisa diakses yang ganjil dan jelek.

Tentu saja, sama sekali bukan itu masalahnya, jadi mari kita jelaskan sebelum melangkah lebih jauh. Apa yang dimaksud aksesibilitas, dan apa yang akan kita pelajari di sini?

Apa yang dimaksud dengan aksesibilitas?

Umumnya, bila kita menyebut sebuah situs bisa diakses, maksud kami adalah materi situs yang tersedia, dan fungsionalitasnya bisa dioperasikan, secara harfiah, oleh *siapa saja*. Sebagai developer, mudah menganggap bahwa semua pengguna bisa melihat dan menggunakan keyboard, mouse, atau layar sentuh, dan bisa berinteraksi dengan materi laman Anda sama seperti yang Anda lakukan. Hal ini bisa menghasilkan pengalaman yang berfungsi dengan baik untuk sebagian orang, namun menimbulkan beragam masalah, dari gangguan biasa hingga penghenti-tampilan pada pengguna lainnya.

Aksesibilitas merujuk pada pengalaman pengguna yang mungkin berada di luar lingkup sempit pengguna "pada umumnya", yang mungkin mengakses atau berinteraksi dengan berbagai hal secara berbeda dari yang Anda perkirakan. Khususnya, ini menyangkut pengguna yang mengalami semacam kelemahan atau cacat — dan ingatlah bahwa pengalaman bisa bersifat non-fisik atau sementara.

Misalnya, walaupun kita cenderung memusatkan diskusi aksesibilitas pada pengguna yang memiliki kelemahan, kita bisa mengaitkan dengan pengalaman menggunakan antarmuka yang tidak bisa diakses oleh kita karena alasan lain. Pernahkah Anda mengalami masalah menggunakan situs versi desktop di ponsel, atau melihat pesan "This content is not available in your area", atau tidak bisa menemukan menu yang familier di tablet? Semua itu adalah masalah aksesibilitas.

Setelah mengetahui selengkapnya, Anda akan tahu bahwa menangani masalah aksesibilitas dalam pengertian yang lebih luas dan lebih umum ini biasanya akan selalu memperbaiki pengalaman pengguna bagi siapa saja. Mari kita amati sebuah contoh:

The screenshot shows a modal window titled 'jo@example.com' containing a payment and order summary. The window includes fields for 'Pay with' (with a dropdown menu labeled 'Select payment method'), 'Ship to' (with a dropdown menu labeled 'Choose address'), and a 'Remember details?' checkbox (which is checked). Below these are sections for 'Subtotal' (\$10.00), 'Shipping' (\$0.00), 'Tax' (\$0.90), and 'Total' (\$10.90). At the bottom is a large 'Place Order' button.

Subtotal		\$10.00
Shipping		\$0.00
Tax		\$0.90
Total		\$10.90

Formulir ini memiliki sejumlah masalah aksesibilitas sebagai berikut:

- Kontras teks rendah, sehingga sulit dibaca oleh pengguna yang memiliki penglihatan lemah.

- Adanya label di sebelah kiri dan bidang-bidang di sebelah kanan menyulitkan banyak orang untuk mengaitkannya, dan hampir mustahil bagi orang yang perlu memperbesar tampilan untuk menggunakan laman tersebut; bayangkan melihatnya di ponsel dan harus menggeser untuk mengetahui apa yang terjadi.
- Label "Remember details?" tidak dikaitkan dengan kotak centang, jadi Anda harus mengetuk atau mengeklik hanya pada segi empat kecil, bukan cuma mengeklik label; selain itu, orang yang menggunakan pembaca layar akan kesulitan mengetahui asosiasinya.

Sekarang, mari kita goyangkan tongkat sihir aksesibilitas dan lihat formulir yang telah diperbaiki masalahnya. Kita akan membuat teksnya lebih gelap, memodifikasi desainnya agar label dekat dengan apa yang dilabeli, dan memperbaiki label untuk dikaitkan dengan kotak centang sehingga Anda juga bisa beralih dengan mengeklik labelnya.

Subtotal	\$10.00
Shipping	\$0.00
Tax	\$0.90
Total	\$10.90

Pay with:
Select payment method ▾

Ship to:
Choose address ▾

Remember details?

Place Order

Mana yang lebih suka Anda gunakan? Jika Anda bilang "versi yang bisa diakses", berarti Anda sudah memahami premis utama panduan ini. Sering kali, sesuatu yang menjadi pemblokir penuh bagi segelintir pengguna juga menjadi hal yang menyakitkan bagi banyak pengguna lainnya, jadi dengan memperbaiki masalah aksesibilitas berarti Anda memperbaiki pengalaman bagi siapa saja.

Pedoman Aksesibilitas Konten Web

Sepanjang panduan ini kita akan merujuk Web Content Accessibility Guidelines (WCAG) 2.0, yaitu serangkaian panduan dan praktik terbaik yang dikumpulkan oleh pakar aksesibilitas untuk menjawab apa yang dimaksud dengan "aksesibilitas" secara metodis. Sejumlah negara sebenarnya mewajibkan penggunaan panduan ini dalam persyaratan legal aksesibilitas web mereka.

WCAG disusun oleh empat prinsip yang sering kali disebut dengan singkatan *POUR*:

- **Perceivable:** Bisakah pengguna mempersepsi materi? Ini membantu mengingatkan kita bahwa hanya karena sesuatu bisa dipersepsi orang dengan satu indera, misalnya penglihatan, tidak berarti semua pengguna bisa mempersepsinya.
- **Operable:** Bisakah pengguna menggunakan komponen UI dan menyusuri materi? Misalnya, sesuatu yang mengharuskan interaksi mengarahkan ke atas tidak bisa dioperasikan oleh orang yang tidak bisa menggunakan mouse atau layar sentuh.
- **Understandable:** Bisakah pengguna memahami materi? Bisakah pengguna memahami antarmuka dan apakah cukup konsisten untuk menghindari kebingungan?
- **Robust:** Bisakah materi dimanfaatkan oleh beragam agen-pengguna (browser)? Bisakah digunakan bersama teknologi pendukung?

Walaupun WCAG menyediakan ringkasan yang komprehensif mengenai apa yang dimaksudnya dengan materi yang bisa diakses, ini juga bisa sedikit membebani. Untuk membantu meminimalkannya, grup WebAIM (Web Accessibility in Mind) telah meringkas panduan WCAG ke dalam sebuah daftar periksa yang mudah diikuti, yang ditargetkan secara khusus untuk materi web.

Daftar periksa WebAIM bisa memberi Anda rangkuman tingkat tinggi mengenai hal-hal yang perlu diimplementasikan, sekaligus menautkan ke spesifikasi WCAG yang mendasarinya jika Anda membutuhkan definisi yang diluaskan.

Dengan alat (bantu) ini, Anda bisa memetakan arah pekerjaan aksesibilitas dan menjadi percaya diri karena, asalkan proyek memenuhi kriteria yang dijelaskan, pengguna akan memiliki pengalaman positif saat mengakses materi Anda.

Memahami keberagaman pengguna

Saat mempelajari tentang aksesibilitas, akan membantu bila memiliki pemahaman mengenai keberagaman di antara para pengguna di seluruh dunia dan aneka topik aksesibilitas yang memengaruhi mereka. Untuk menjelaskan lebih jauh, inilah sesi tanya-jawab informatif dengan Victor Tsaran, Technical Program Manager di Google, yang buta total.

Apa yang Anda kerjakan di Google?

Di Google, pekerjaan saya adalah membantu memastikan produk kami berfungsi untuk semua macam pengguna, tanpa memandang kemampuan atau pun kelemahan.

Kelemahan macam apa yang dimiliki pengguna?

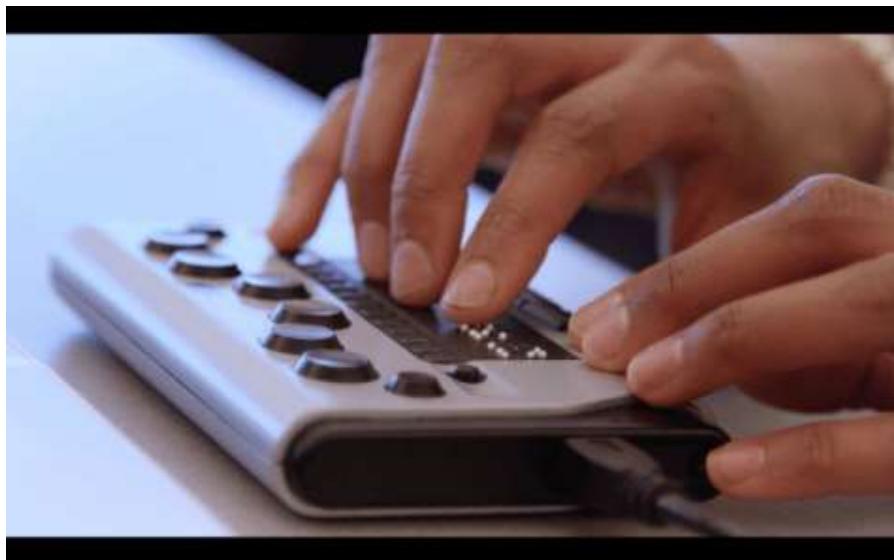
Bila kita memikirkan tentang kelemahan, yang akan menyulitkan seseorang untuk mengakses materi kita, banyak orang akan langsung membayangkan pengguna tuna netra seperti saya. Memang ada benarnya, kelemahan ini bisa sangat membuat frustrasi atau bahkan mustahil untuk menggunakan banyak situs web.

Banyak teknik web modern memiliki efek samping tidak menguntungkan pada pembuatan situs yang tidak berfungsi dengan baik pada alat yang digunakan oleh pengguna tuna netra untuk

mengakses web. Akan tetapi, sebenarnya aksesibilitas lebih dari itu. Kami rasa ada gunanya membayangkan kelemahan ke dalam empat kelompok besar: visual, motor, pendengaran, dan kognitif.

Mari kita bahas satu per satu. Bisakah Anda berikan beberapa contoh kelemahan visual?

Kelemahan visual bisa dibagi ke dalam beberapa kategori: Pengguna yang tidak memiliki penglihatan, seperti saya, dapat menggunakan pembaca layar, braille, atau kombinasi keduanya.

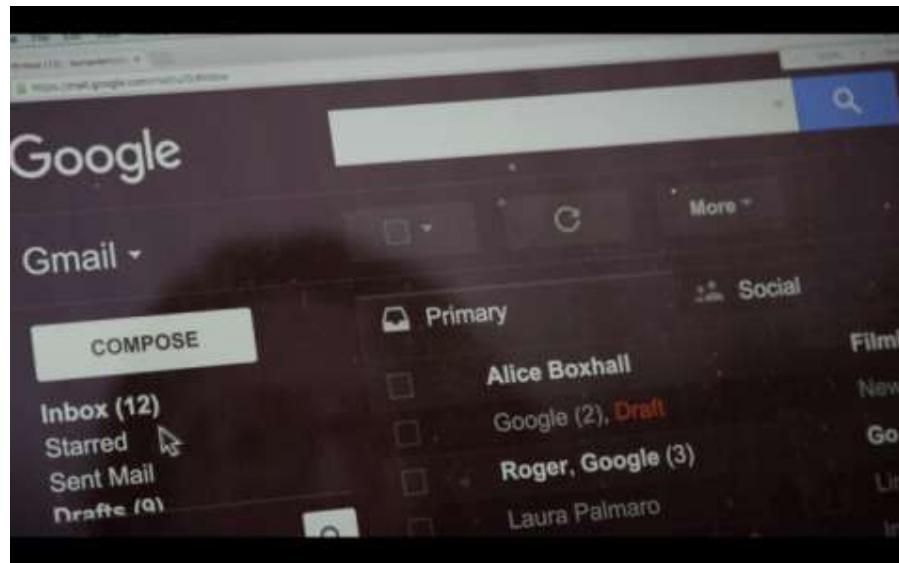


Pembaca braille

Sekarang, sebenarnya jarang ada orang yang tidak bisa melihat sama sekali, namun tetap ada, mungkin saja Anda mengenal atau berjumpa dengan setidaknya satu orang yang tidak bisa melihat sama sekali. Akan tetapi juga jauh lebih banyak orang yang kita sebut pengguna yang berpenglihatan lemah.

Inilah kelompok terbesar, mulai dari orang-orang yang seperti istri saya, yang tidak memiliki kornea — jadi walaupun pada dasarnya ia bisa melihat sesuatu, ia kesulitan membaca barang cetak dan secara legal dianggap buta — hingga orang yang memiliki penglihatan buruk dan perlu mengenakan kaca mata resep yang sangat tinggi.

Ada banyak sekali macamnya, dan dengan sendirinya juga ada banyak macam akomodasi yang digunakan oleh orang-orang yang ada dalam kategori ini: ada yang menggunakan pembaca layar atau tampilan braille (saya bahkan pernah mendengar ada seorang wanita yang membaca braille yang ditampilkan di layar karena lebih mudah dilihat daripada teks tercetak), atau mereka mungkin menggunakan teknologi teks-ke-ucapan tanpa fungsionalitas lengkap pembaca layar, atau mereka dapat menggunakan pembesar tampilan layar yang memperbesar bagian layar, atau mereka mungkin cuma menggunakan zoom di browser untuk membuat semua font tampak lebih besar. Mereka mungkin juga menggunakan opsi kontras tinggi seperti mode kontras tinggi di sistem operasi, ekstensi browser kontras tinggi atau tema kontras tinggi untuk situs web.



Mode kontras tinggi

Banyak pengguna bahkan menggunakan kombinasi dari semua ini, seperti teman saya Laura yang menggunakan kombinasi mode kontras tinggi, zoom di browser, teks-ke-ucapan.

Penglihatan rendah adalah sesuatu yang bisa dikaitkan dengan banyak orang. Sebagai awal, kita semua mengalami penurunan penglihatan seiring usia, jadi sekalipun Anda belum mengalaminya mungkin Anda pernah mendengar orang tua Anda mengeluhkan hal ini. Namun banyak orang mengalami frustrasi saat membawa laptopnya keluar ruangan saat matahari cerah dan tiba-tiba tidak bisa membaca apa-apa! Atau orang yang telah menjalani bedah laser atau mungkin cuma harus membaca sesuatu dari seluruh ruangan mungkin telah menggunakan salah satu akomodasi yang tadi saya sebutkan. Jadi menurut saya agak mudah bagi developer untuk memiliki empati bagi pengguna yang berpenglihatan rendah.

Oh, jangan lupa dengan orang-orang yang memiliki penglihatan warna yang buruk — sekitar 9% pria memiliki semacam defisiensi penglihatan warna! Plus sekitar 1% wanita. Mereka mungkin kesulitan membedakan merah dan hijau, atau kuning dan biru. Pertimbangkan tentang hal itu bila Anda nanti mendesain validasi formulir.

Bagaimana dengan kelemahan motorik?

Ya, kelemahan motorik, atau kelemahan ketangkasan fisik. Kelompok ini meliputi orang-orang yang lebih suka tidak menggunakan mouse, mungkin karena mengalami semacam RSI atau sesuatu dan merasa nyeri bila menggunakannya, hingga orang yang mungkin lumpuh fisik dan memiliki jangkauan gerak terbatas untuk anggota tubuh tertentu.

Pengguna yang mengalami kelemahan motorik mungkin menggunakan keyboard, perangkat switch, kontrol suara, atau bahkan perangkat pelacak mata untuk berinteraksi dengan komputer mereka.



Perangkat pelacak mata

Mirip dengan kelemahan penglihatan, mobilitas juga bisa menjadi masalah sementara atau situasional: Mungkin Anda mengalami patah pada tangan yang biasa memegang mouse. Mungkin trackpad rusak di laptop, atau Anda sedang naik kereta yang bergoncang-goncang. Bisa jadi ada banyak situasi yang menghalangi mobilitas pengguna, dan dengan memastikan kita mempedulikan mereka berarti kita memperbaiki pengalaman keseluruhan, untuk siapa saja yang memiliki kelemahan permanen maupun mereka yang untuk sementara tidak bisa menggunakan UI berbasis pointer.

Bagus, mari kita bicara tentang kelemahan pendengaran.

Kelompok ini bisa meliputi orang yang sepenuhnya tuli hingga orang yang sulit-mendengar. Dan mirip sekali dengan penglihatan, pendengaran kita cenderung menurun bersama usia. Banyak dari kita menggunakan kemampuan umum seperti alat bantu dengar.



Teks layar

Bagi pengguna yang lemah pendengaran, perlu dipastikan bahwa kita tidak mengandalkan suara, jadi pastikan menggunakan sesuatu seperti teks video dan transkrip, serta menyediakan semacam alternatif, jika suara merupakan bagian dari antarmuka.

Dan seperti yang kita ketahui pada kelemahan motorik dan penglihatan, mudah sekali kita bayangkan bahwa orang yang pendengarannya bekerja dengan baik akan turut merasakan manfaat akomodasi ini. Banyak teman saya bilang mereka suka bila video dilengkapi teks dan transkrip karena dengan begitu jika mereka berada di kantor yang tidak bersekat dan mereka tidak membawa headphone, mereka tetap bisa menonton video!

Baiklah, bisa jelaskan sedikit tentang kelemahan kognitif?

Ada banyak macam kondisi kognitif seperti ADD, Disleksia, dan Autis, yang berarti orang-orang demikian ingin atau perlu mengakses sesuatu dengan cara berbeda. Akomodasi untuk kelompok ini dengan sendirinya sangat beragam sekali, namun pasti menemukan semacam tumpang tindih dengan area lain, seperti penggunaan fungsionalitas zoom untuk mempermudah membaca atau berkonsentrasi. Selain itu, para pengguna ini mungkin merasa bahwa desain yang benar-benar minimal adalah yang paling bagus karena meminimalkan distraksi dan beban kognitif.

Saya kira siapa saja bisa mengaitkan dengan stres kelebihan beban kognitif, jadi jelas bahwa jika kita membuat sesuatu yang berfungsi dengan baik untuk orang yang memiliki kelemahan kognitif, maka kita akan membuat sesuatu yang akan menjadi pengalaman menyenangkan bagi siapa saja.

Jadi, bagaimana kesimpulan pendapat Anda tentang aksesibilitas?

Bila Anda mengamati berbagai macam kemampuan dan ketidakmampuan yang dimiliki orang, Anda bisa melihat bahwa mendesain dan membangun produk hanya bagi orang yang memiliki penglihatan, pendengaran, ketangkasan, dan kognisi sempurna, nampaknya itu benar-benar sempit. Ini berlawanan dengan tujuan semula, karena kita akan membuat pengalaman yang lebih menimbulkan stres dan kurang berguna bagi siapa saja, dan bagi sebagian pengguna akan menghasilkan pengalaman yang sebenarnya akan mengecualikan mereka sama sekali.

Dalam wawancara ini, Victor mengidentifikasi serangkaian kelemahan, dan memasukkannya ke dalam empat kategori besar: *visual*, *motorik*, *pendengaran*, dan *kognitif*. Ia juga menunjukkan bahwa setiap kelemahan bisa bersifat *situasional*, *sementara*, atau *permanen*.

Mari kita amati beberapa contoh sungguhan dari kelemahan akses dan melihat ke dalam kategori serta tipe apa memasukkannya. Perhatikan, beberapa kelemahan mungkin dimasukkan dalam lebih dari satu kategori atau tipe.

	Situasional	Sementara	Permanen
Visual		goncangan	kebutaan
Motor	memegang bayi	lengan patah, RSI*	RSI*
Pendengaran	kantor yang berisik		
Kognitif		goncangan	

*Repetitive Strain Injury - Cedera Akibat Tegang Berulang: mis., sindrom saluran karpal, radang siku, jari pemicu

Langkah berikutnya

Kita sudah banyak membahas hal-hal pokok! Anda telah membaca tentang

- apa yang dimaksud aksesibilitas dan mengapa hal ini penting bagi siapa saja
- daftar periksa aksesibilitas WCAG dan WebAIM
- berbagai tipe kelemahan yang harus Anda pertimbangkan

Untuk panduan selebihnya, kita akan mendalami beberapa aspek praktis dalam pembuatan situs web yang bisa diakses. Kita akan menyusun upaya ini dalam tiga bidang bahasan utama:

- **Fokus:** Kita akan mengamati cara membangun sesuatu yang bisa dioperasikan dengan keyboard sebagai ganti mouse. Tentunya hal ini penting bagi pengguna yang memiliki kelemahan motorik, namun juga memastikan bahwa UI Anda cocok untuk semua pengguna.
- **Semantik:** Kita akan memastikan bahwa kita mengekspresikan antarmuka pengguna dengan cara sempurna yang bisa digunakan pada berbagai macam teknologi pendukung.
- **Penataan gaya:** Kita akan mempertimbangkan desain visual dan mengamati beberapa teknik untuk membuat elemen visual antarmuka seluwes dan seberguna mungkin.

Masing-masing pokok bahasan bisa mengisi keseluruhan kursus, jadi kita tidak akan membahas setiap aspek pembuatan situs web yang bisa diakses. Akan tetapi, kita akan memberi Anda informasi yang cukup untuk memulai, dan menunjukkan beberapa tempat yang bagus untuk mempelajari tentang setiap topik tersebut.

5.1 Pengantar Fokus

Dalam pelajaran ini, kita akan membicarakan tentang *fokus* dan cara mengelolanya dalam aplikasi Anda. Fokus merujuk pada kontrol mana pada layar (item masukan seperti bidang, kotak centang, tombol, atau tautan) yang saat ini menerima masukan dari keyboard, dan dari clipboard bila Anda menempelkan materi.

Inilah tempat yang bagus untuk mulai mempelajari tentang aksesibilitas karena kita semua tahu cara menggunakan keyboard, mudah untuk menghubungkan dan menguji, serta menguntungkan hampir semua pengguna.

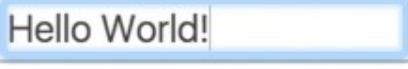
Pengguna yang memiliki gangguan motorik, yang boleh jadi berupa paralisis permanen hingga pergelangan yang keseleo, mungkin mengandalkan keyboard atau perangkat switch untuk menyusuri laman Anda, jadi strategi fokus yang baik adalah penting untuk menyediakan pengalaman yang bagus bagi mereka.

Dan bagi pengguna berpengalaman yang mengetahui setiap pintasan keyboard di mesin mereka, karena bisa dengan cepat menyusuri situs Anda menggunakan keyboard saja, tentu akan membuat mereka menjadi lebih produktif.

Karena itu, strategi fokus yang diimplementasikan dengan baik akan memastikan setiap orang yang menggunakan aplikasi Anda akan memiliki pengalaman yang lebih baik. Nanti akan kita lihat dalam pelajaran berikutnya bahwa upaya yang Anda berikan pada fokus adalah basis penting untuk mendukung pengguna teknologi pendukung, juga tentunya, semua pengguna.

Apa yang dimaksud dengan fokus?

Fokus menentukan tempat kejadian keyboard di laman pada saat tertentu. Misalnya, jika Anda memfokus bidang masukan teks dan mulai mengetik, bidang masukan akan menerima kejadian keyboard dan menampilkan karakter yang Anda ketikkan. Walaupun memiliki fokus, bidang masukan juga akan menerima masukan yang ditempelkan dari clipboard.



Hello World!

Item yang saat ini difokus sering kali ditunjukkan oleh *lingkaran fokus*, gaya yang bergantung pada browser maupun segala penataan gaya yang diterapkan oleh penulis laman tersebut. Chrome, misalnya, biasanya menyorot elemen yang difokus dengan border biru, sedangkan Firefox menggunakan border putus-putus.

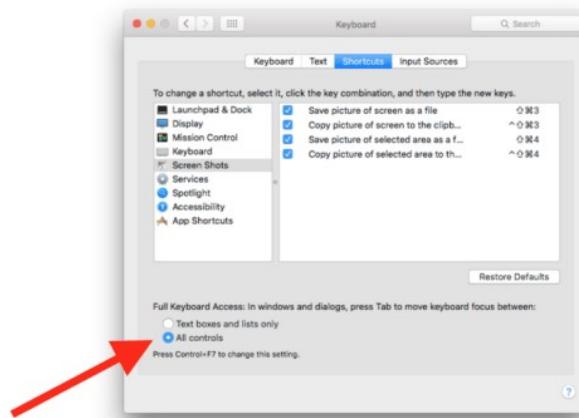


Sign Up!

Sebagian pengguna mengoperasikan komputer mereka hampirnya sepenuhnya dengan keyboard atau perangkat masukan lainnya. Bagi para pengguna tersebut, fokus sangatlah penting; inilah sarana utama mereka untuk melakukan apa saja di layar. Karena alasan itu, daftar periksa Web AIM menyatakan di bagian 2.1.1 bahwa **semua fungsionalitas laman harus tersedia menggunakan keyboard**, kecuali jika ada yang tidak bisa Anda lakukan dengan keyboard, misalnya menggambar bebas.

Sebagai pengguna, Anda bisa mengontrol elemen mana yang saat ini difokus dengan menggunakan Tab, Shift+Tab, atau tombol panah. Di Mac OSX, cara kerjanya sedikit berbeda: walaupun Chrome selalu memungkinkan Anda melakukan navigasi dengan Tab, Anda perlu menekan Option+Tab untuk mengubah fokus di browser lain seperti Safari. (Anda bisa mengubah setelan ini di bagian Keyboard pada System Preferences.)

System Preferences > Keyboard



Urutan pemrosesan fokus maju dan mundur pada elemen interaktif lewat Tab disebut *urutan tab*. Langkah penting untuk memastikan bahwa Anda mendesain laman dengan urutan tab yang logis akan kita bahas nanti.

Apa yang dapat difokus?

Elemen HTML interaktif bawaan seperti bidang teks, tombol, dan daftar pilihan secara *implisit dapat difokus*, maksudnya, secara otomatis elemen itu akan disisipkan ke dalam urutan tab dan memiliki penanganan kejadian keyboard bawaan tanpa intervensi developer.

Implicitly focusable

automatically in the
tab order + built-in
keyboard event handling

Username

Click Me!

Aisle Seat ▾

Namun tidak semua elemen dapat difokus; paragraf, div, dan beragam elemen laman lainnya tidak difokus saat Anda berpindah tab di laman, dan memang didesain demikian. Umumnya tidak perlu memfokus sesuatu jika pengguna tidak bisa berinteraksi dengannya.



Mencoba fokus

Mari kita coba beberapa teknik fokus yang baru saja kita diskusikan. Dengan menggunakan Chrome, masuklah ke [laman tiruan situs maskapai ini](#) dan cari tiket tertentu **hanya dengan menggunakan masukan keyboard**. Laman tersebut tidak akan menerima masukan mouse, jadi Anda tidak bisa memalsukan latihan (bukannya kami tidak percaya Anda ;-).

A screenshot of a travel search form titled "Where ya headed, mate?". The form includes fields for Full Name, Address, and travel dates. It also has a dropdown for Preferred seat type and a checkbox for promotional offers. A "Search" button is at the bottom right. The background features a scenic landscape with a beach and ocean.

Udacity *Down Under* Flights Hotels Rental Cars

One Way Round Trip Multi City

Where ya headed, mate?

Full Name

Enter your address

Departure Arrival Depart Date Return Date
mm/dd/yyyy mm/dd/yyyy

Preferred seat type
No preference

Receive promotional offers?

Search

About us Join our newsletter Need Support? © 2015 Udacity

Parameter tiket yang harus Anda tetapkan adalah:

- sekali jalan
- ke Melbourne
- berangkat tanggal 12 Oktober 2017 (10/12/2017)

- kembali tanggal 23 Oktober 2017 (10/23/2017)
- kursi dekat jendela
- tidak ingin menerima penawaran promosi

Bila Anda berhasil menyelesaikan formulir tanpa kesalahan masukan dan mengaktifkan tombol Search, formulir akan dikosongkan dan disetel ulang begitu saja. Lanjutkan dan selesaikan formulir, kemudian kembalilah.

Mari kita periksa bagaimana formulir menggunakan masukan keyboard Anda. Mulai dengan beberapa penekanan Tab pertama, browser menyorot item navigasi untuk Flights, Hotels, dan Rental Cars. Karena terus menekan Tab Anda melanjutkan ke grup radiobutton untuk memilih dari Round Trip, One Way, atau Multi City dengan menggunakan tombol panah.

Lanjutkan sampai bidang nama dan alamat, dengan mengisi informasi yang diperlukan. Bila sampai di elemen pemilihan tujuan, Anda bisa menggunakan tombol panah untuk memilih kota, atau bisa mulai mengetikkan untuk mengisi bidang pelengkapan otomatis. Begitu pula, dalam bidang tanggal, Anda bisa menggunakan tombol panah atau cuma mengetikkan tanggal.

Memilih tipe kursi juga akan menggunakan tombol panah, atau Anda bisa mengetikkan "w", "a", atau "n" untuk lompat ke opsi kursi. Kemudian Anda bisa menonaktifkan default penawaran promosi dengan menekan spasi saat kotak centang difokus. Terakhir, fokus tombol Search dan tekan Enter untuk mengirim formulir.

Sangat praktis berinteraksi dengan formulir cuma dengan menggunakan keyboard dan tidak perlu beralih ke mouse dan kembali menyelesaikan tugas. Karena semua elemen yang digunakan dalam formulir adalah tag HTML asli dengan fokus implisit, formulir bekerja dengan baik bersama keyboard, dan Anda tidak perlu menulis kode untuk menambahkan atau mengelola perilaku fokus.

5.2. Pentingnya Urutan DOM

Menggunakan elemen asli merupakan cara bagus untuk mempelajari tentang perilaku fokus karena elemen asli secara otomatis disisipkan ke dalam urutan tab berdasarkan posisinya di DOM.

Misalnya, Anda mungkin memiliki tiga elemen tombol yang berurutan dalam DOM. Menekan Tab akan memfokus setiap tombol secara berurutan. Cobalah mengeklik blok kode di bawah ini untuk memindah titik mulai navigasi fokus, kemudian tekan Tab untuk memindah fokus melewati tombol-tombol.

```
<button>I Should</button>
<button>Be Focused</button>
<button>Last!</button>
```

Akan tetapi, penting untuk diingat bahwa, dengan menggunakan CSS, dimungkinkan menempatkan sesuatu dalam satu urutan di DOM namun muncul dalam urutan berbeda di layar. Misalnya, jika Anda menggunakan properti CSS seperti `float` untuk memindah satu tombol ke kanan, tombol-tombol itu akan muncul dalam urutan berbeda di layar. Namun, karena urutannya dalam DOM tetap sama, maka begitu pula urutan tabnya. Bila pengguna berpindah tab di laman, tombol akan mendapat fokus dalam urutan yang tidak intuitif. Cobalah mengeklik blok kode di bawah ini untuk memindah titik mulai navigasi fokus, kemudian tekan Tab untuk memindah fokus melewati tombol-tombol.

```
<button style="float: right">I Should</button>
<button>Be Focused</button>
<button>Last!</button>
```

Berhati-hatilah saat mengubah posisi visual elemen di layar dengan menggunakan CSS. Hal ini bisa menyebabkan urutan tab melompat-lompat, seolah acak, sehingga membingungkan pengguna yang mengandalkan keyboard. Karena alasan ini, daftar periksa Web AIM menyatakan [di bagian 1.3.2](#) bahwa urutan navigasi dan pembacaan, seperti yang ditentukan oleh urutan kode, harus logis dan intuitif.

Sebagai aturan, cobalah berpindah-pindah tab melewati berbagai laman sesering mungkin sekadar untuk memastikan Anda bukan secara tidak sengaja mengacaukan urutan tab. Ini adalah kebiasaan yang baik untuk diterapkan, dan hal ini tidak memerlukan banyak usaha.

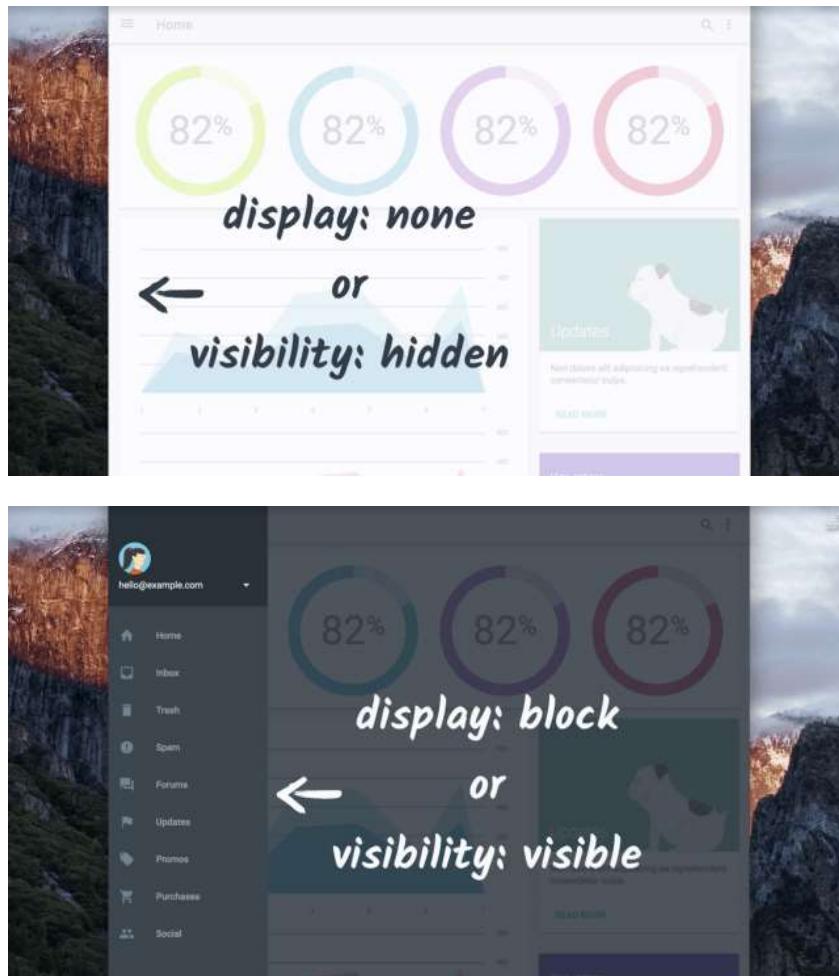
Materi tak terlihat

Bagaimana jika Anda memiliki materi yang saat ini tidak ditampilkan, namun tetap perlu ada di DOM, misalnya navigasi samping yang responsif? Bila Anda memiliki elemen seperti ini yang menerima fokus bila sedang tidak tampak di layar, elemen ini bisa terlihat seakan fokusnya menghilang dan muncul kembali saat pengguna berpindah tab di laman — ini jelas sebuah efek yang tidak diinginkan. Idealnya, kita harus mencegah agar panel tidak mendapat fokus bila sedang di luar layar, dan hanya bisa difokus bila pengguna bisa berinteraksi dengannya.



Kadang-kadang Anda perlu melakukan semacam pekerjaan detektif untuk mengetahui ke mana larinya fokus. Anda bisa menggunakan `document.activeElement` dari konsol untuk mengetahui elemen mana yang saat ini difokus.

Setelah mengetahui elemen di luar layar mana yang sedang difokus, Anda bisa menyetelnya ke `display: none` atau `visibility: hidden`, kemudian menyetelnya kembali ke `display: block` atau `visibility: visible` sebelum menampilkannya kepada pengguna.



Secara umum, kami mendorong developer untuk berpindah-pindah tab dalam situs mereka sebelum mempublikasikan untuk mengetahui apakah urutan tab tidak menghilang atau melompat dari urutan logis. Jika ternyata menghilang atau melompat, pastikan Anda telah menyembunyikan materi di luar layar dengan benar melalui `display: none` atau `visibility: hidden`, atau apakah Anda telah menyusun ulang posisi fisik elemen dalam DOM sehingga berada dalam urutan logis.

5.3 Menggunakan tabindex

Urutan tab default yang disediakan oleh posisi DOM elemen asli sudah praktis, namun ada kalanya Anda ingin memodifikasi urutan tab, dan secara fisik memindahkan elemen di HTML tidak selalu merupakan solusi yang optimal, apalagi layak. Untuk kasus-kasus ini, Anda bisa menggunakan atribut HTML `tabindex` untuk menyetel posisi tab elemen secara eksplisit.

`tabindex` bisa diterapkan ke elemen apa saja — walaupun tidak harus berguna pada setiap elemen — dan menggunakan serangkaian nilai integer. Dengan menggunakan `tabindex`, Anda bisa menetapkan urutan eksplisit untuk elemen laman yang bisa difokus, menyisipkan elemen yang tidak bisa difokus ke dalam urutan tab, dan membuang elemen dari urutan tab. Misalnya:

`tabindex="0"`: Menyisipkan sebuah elemen ke dalam urutan tab alami. Elemen bisa difokus dengan menekan tombol `Tab`, dan elemen bisa difokus dengan memanggil metode `focus()`-nya

```
<custom-button tabindex="0">Press Tab to Focus Me!</custom-button>
```

`tabindex="-1"`: Membuang elemen dari urutan tab alami, namun elemen tetap bisa difokus dengan memanggil metode `focus()`-nya

```
<button id="foo" tabindex="-1">I'm not keyboard focusable</button>
<button onclick="foo.focus();">Focus my sibling</button>
```

`tabindex="5"`: Semua `tabindex` yang lebih besar dari 0 akan melompatkan elemen ke depan urutan tab alami. Jika ada beberapa elemen dengan `tabindex` lebih besar dari 0, urutan tab akan mulai dari nilai terendah yang lebih besar dari nol dan terus ke atas. Menggunakan `tabindex` yang lebih besar dari 0 dianggap sebuah **anti-pola**.

```
<button>I should be first</button>
<button>And I should be second</button>
<button tabindex="5">But I jumped to the front!</button>
```

Hal ini khususnya berlaku untuk elemen non-masukan seperti header, gambar, atau judul artikel. Penambahan `tabindex` ke elemen semacam itu adalah kontra-produktif. Jika memungkinkan, sebaiknya susun kode sumber Anda agar urutan DOM menyediakan urutan tab yang logis. Jika Anda menggunakan `tabindex`, batasi pada tombol kontrol interaktif khusus seperti tombol, tab, tarik-turun, dan bidang teks; yakni, elemen yang mungkin dikira pengguna untuk menyediakan masukan.

Jangan khawatir pengguna pembaca layar akan melewatkkan materi penting karena tidak memiliki `tabindex`. Sekalipun materi sangat penting, seperti gambar, jika itu bukanlah hal yang akan berinteraksi dengan pengguna, maka tidak ada alasan membuatnya bisa difokus. Pengguna

pembaca layar tetap bisa memahami materi gambar asalkan Anda menyediakan dukungan atribut alt yang sesuai, yang nanti akan kita bahas sebentar lagi.

5.4 Mengelola fokus pada level laman

Inilah skenario di mana tabindex tidak hanya berguna, melainkan diperlukan. Anda mungkin membangun laman tunggal yang sempurna dengan beragam bagian materi, yang tidak semuanya terlihat sekaligus. Di laman semacam ini, mengeklik tautan navigasi mungkin mengubah materi yang terlihat tanpa melakukan penyegaran laman.

Bila ini terjadi, Anda mungkin akan mengidentifikasi area materi yang dipilih, memberinya sebuah tabindex berupa -1 sehingga tidak muncul dalam urutan tab alami, dan memanggil metode focus-nya. Teknik ini, yang disebut *mengelola fokus*, akan membuat konteks yang dipersepsi pengguna tetap sinkron dengan materi visual situs.

5.5 Mengelola fokus di komponen

Mengelola fokus saat Anda mengubah sesuatu di laman adalah hal penting, namun kadang-kadang Anda perlu mengelola fokus pada level kontrol — misalnya, jika sedang membangun komponen khusus.

Pertimbangkan elemen select asli. Elemen ini bisa menerima fokus dasar, bila ada, Anda bisa menggunakan tombol panah untuk mengekspos fungsionalitas tambahan (opsi yang bisa dipilih). Jika sedang membangun elemen select khusus, Anda tentu ingin mengekspos perilaku semacam ini sehingga pengguna yang terutama mengandalkan keyboard tetap bisa berinteraksi dengan kontrol Anda.

```
<!-- Focus the element using Tab and use the up/down arrow keys to navigate -->
<select>
  <option>Aisle seat</option>
  <option>Window seat</option>
  <option>No preference</option>
</select>
```

Boleh jadi sulit mengetahui perilaku keyboard mana yang akan diimplementasikan, namun ada dokumen berguna yang bisa Anda rujuk. Panduan [Accessible Rich Internet Applications \(ARIA\) Authoring Practices](#) mencantumkan daftar tipe komponen dan macam-macam tindakan keyboard yang didukungnya. Kita akan membahas ARIA secara lebih detail nanti, namun untuk saat ini mari kita gunakan panduan tersebut untuk membantu menambahkan dukungan keyboard ke sebuah komponen baru.

Mungkin Anda sedang mengerjakan beberapa [Elemen Khusus](#) baru yang menirukan serangkaian tombol radio, namun dengan penampilan dan perilaku tiruan yang unik.

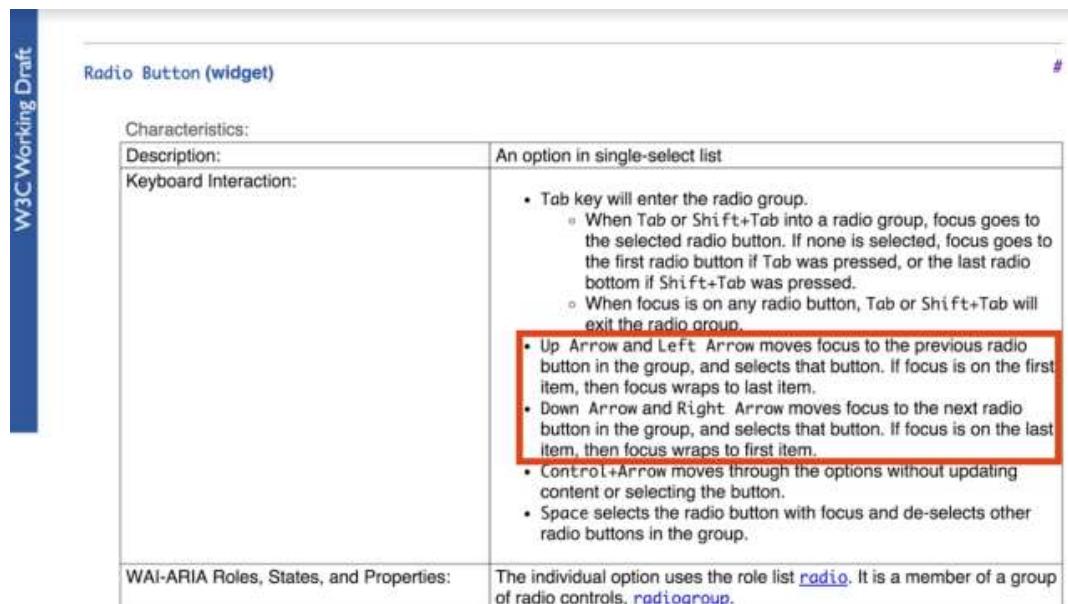
```

<radio-group>
  <radio-button>Water</radio-button>
  <radio-button>Coffee</radio-button>
  <radio-button>Tea</radio-button>
  <radio-button>Cola</radio-button>
  <radio-button>Ginger Ale</radio-button>
</radio-group>

```

Untuk menentukan dukungan keyboard macam apa yang dibutuhkan, Anda perlu memeriksa [panduan ARIA Authoring Practices](#). Bagian 2 berisi daftar pola desain, dan dalam daftar itu adalah [tabel karakteristik untuk grup tombol radio](#), komponen yang ada yang paling cocok dengan elemen baru Anda.

Seperti yang bisa Anda lihat dalam tabel, salah satu perilaku keyboard umum yang harus didukung adalah tombol panah naik/turun/kiri/kanan. Untuk menambahkan perilaku ini ke komponen baru, kita akan menggunakan teknik yang disebut *roving tabindex*.



The screenshot shows a table titled "Radio Button (widget)" under the "W3C Working Draft" heading. The table has two main sections: "Characteristics:" and "WAI-ARIA Roles, States, and Properties:". The "Characteristics:" section contains a table with two columns: "Description:" and "Keyboard Interaction:". The "Description:" column contains the text "An option in single-select list". The "Keyboard Interaction:" column contains a list of keyboard interactions. A red box highlights the following items from this list:

- Tab key will enter the radio group.
 - When Tab or Shift+Tab into a radio group, focus goes to the selected radio button. If none is selected, focus goes to the first radio button if Tab was pressed, or the last radio button if Shift+Tab was pressed.
 - When focus is on any radio button, Tab or Shift+Tab will exit the radio group.
- Up Arrow and Left Arrow moves focus to the previous radio button in the group, and selects that button. If focus is on the first item, then focus wraps to last item.
- Down Arrow and Right Arrow moves focus to the next radio button in the group, and selects that button. If focus is on the last item, then focus wraps to first item.
- Control+Arrow moves through the options without updating content or selecting the button.
- Space selects the radio button with focus and de-selects other radio buttons in the group.

The "WAI-ARIA Roles, States, and Properties:" section contains the text: "The individual option uses the role list [radio](#). It is a member of a group of radio controls, [radiogroup](#)".

Roving tabindex bekerja dengan menyetel tabindex ke -1 untuk semua anak selain yang saat ini aktif.

```

<radio-group>
  <radio-button tabindex="0">Water</radio-button>
  <radio-button tabindex="-1">Coffee</radio-button>
  <radio-button tabindex="-1">Tea</radio-button>
  <radio-button tabindex="-1">Cola</radio-button>
  <radio-button tabindex="-1">Ginger Ale</radio-button>
</radio-group>

```

Selanjutnya komponen menggunakan event listener keyboard untuk menentukan tombol mana yang ditekan pengguna; bila terjadi, maka ini akan menyetel tabindex anak yang difokus sebelumnya ke -1, menyetel tabindex anak yang akan difokus ke 0, dan memanggil metode padanya.

```
<radio-group>
  // Assuming the user pressed the down arrow, we'll focus the next available child
  <radio-button tabindex="-1">Water</radio-button>
  <radio-button tabindex="0">Coffee</radio-button> // call .focus() on this element
  <radio-button tabindex="-1">Tea</radio-button>
  <radio-button tabindex="-1">Cola</radio-button>
  <radio-button tabindex="-1">Ginger Ale</radio-button>
</radio-group>
```

Bila pengguna mencapai anak yang terakhir (atau pertama, bergantung pada arahnya menggerakkan fokus), Anda akan membuat loop dan memfokus lagi anak yang pertama (atau terakhir).

Anda bisa mencoba contoh yang telah selesai di bawah ini. Periksa elemen di DevTools untuk mengamati pergerakan tabindex dari satu tombol radio ke tombol berikutnya.

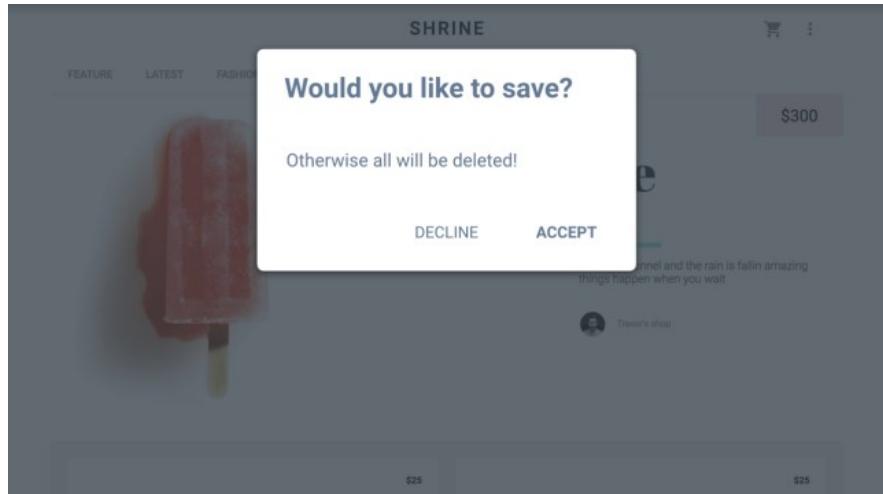
Anda bisa menampilkan [sumber lengkap elemen ini](#) melalui GitHub.

5.6 Modal dan jebakan keyboard

Kadang-kadang saat mengelola fokus, Anda bisa masuk ke suatu situasi dan tidak bisa keluar lagi. Perhatikan sebuah widget pelengkapan otomatis yang mencoba mengelola fokus serta merekam perilaku tab, namun mencegah pengguna meninggalkannya bila belum selesai. Ini disebut *jebakan keyboard*, dan hal ini bisa sangat mengesalkan pengguna. Bagian 2.1.2 pada daftar periksa Web AIM menangani masalah ini, yang menyatakan bahwa [fokus keyboard tidak boleh dikunci atau terjebak pada satu elemen laman tertentu](#). Pengguna harus bisa mengarah ke dan dari semua elemen laman hanya dengan menggunakan keyboard.

Anehnya, ada kalanya perilaku ini malah sebenarnya yang diinginkan, seperti jendela modal. Biasanya, bila modal ditampilkan, Anda ingin pengguna mengakses materi di baliknya. Anda mungkin menambahkan overlay untuk menutupi laman secara visual, namun itu tidak menghentikan pergerakan fokus keyboard keluar dari modal secara tidak sengaja.

Dalam instance seperti ini Anda bisa mengimplementasikan jebakan keyboard sementara untuk memastikan bahwa Anda menjebak fokus hanya saat modal ditampilkan kemudian memulihkan fokus item yang sebelumnya difokus bila modal ditutup.



Ada beberapa proposal mengenai cara untuk mempermudah hal ini bagi developer, termasuk elemen <dialog>, namun belum mendapatkan dukungan browser yang luas.

Lihat [artikel MDN](#) ini untuk informasi selengkapnya mengenai <dialog>, dan [contoh modal](#) untuk informasi selengkapnya mengenai jendela modal.

Perhatikan dialog modal yang dinyatakan oleh div yang berisi beberapa elemen, dan div yang menyatakan overlay latar belakang. Mari kita bahas langkah-langkah dasar yang dibutuhkan untuk mengimplementasikan jebakan keyboard sementara di situasi ini.

1. Dengan menggunakan `document.querySelector`, pilih div modal dan overlay serta simpan referensinya.
2. Saat modal dibuka, simpan referensi ke elemen yang telah difokus ketika modal dibuka sehingga Anda bisa mengembalikan fokus ke elemen itu.
3. Gunakan *keydown listener* untuk menangkap tombol-tombol yang ditekan saat modal dibuka. Anda juga mendengarkan klik di overlay latar belakang, dan menutup modal jika pengguna menekliknya.
4. Berikutnya, ambil kumpulan elemen yang bisa difokus dalam modal. Elemen yang bisa difokus pertama dan terakhir akan berfungsi sebagai "sentinel" untuk memberi tahu Anda kapan membuat loop fokus ke depan atau ke belakang agar tetap berada dalam modal.
5. Tampilkan jendela modal dan fokus elemen yang bisa difokus pertama.
6. Saat pengguna menekan Tab atau Shift+Tab, pindahkan fokus ke depan atau ke belakang, dengan melakukan loop pada elemen terakhir atau pertama.
7. Jika pengguna menekan Esc, tutuplah modal. Hal ini sangat membantu karena memungkinkan pengguna menutup modal tanpa menelusuri tombol tutup tertentu, dan hal ini menguntungkan bahkan untuk pengguna yang menggunakan mouse.

8. Bila tombol modal ditutup, sembunyikan ini dan overlay latar belakang, serta pulihkan fokus ke elemen yang telah difokus sebelumnya yang telah disimpan.

Prosedur ini memberi Anda jendela modal yang berguna dan tidak mengesalkan, yang bisa digunakan siapa saja secara efektif.

Untuk detail selengkapnya, Anda bisa memeriksa [kode contoh](#) ini, dan menampilkan contoh langsung dari [laman yang telah selesai](#).

BAB VI PENGANTAR SEMANTIK

Sasaran Pembelajaran

Mahasiswa mengetahui apa yang dimaksud aksesibilitas dan cara menerapkannya serta mengetahui tentang berbagai teknik aksesibilitas modern untuk membuat pengalaman aksesibilitas yang brilian

Kemampuan mahasiswa yang menjadi prasyarat

Mahasiswa sudah menguasai materi Interaksi Manusia Komputer

Manfaat atau pentingnya bahan pembelajaran ini

Memberi pemahaman mengenai aksesibilitas

Anda telah melihat cara membuat situs bisa diakses oleh pengguna yang tidak bisa menggunakan mouse atau perangkat penunjuk — baik karena cacat fisik, masalah teknologi, atau preferensi pribadi — dengan menangani penggunaan keyboard-saja. Walaupun memerlukan kehati-hatian dan pertimbangan, ini bukanlah pekerjaan yang sangat berat jika Anda merencanakannya dari awal. Setelah pekerjaan dasar selesai, Anda tinggal menuju ke situs yang bisa diakses penuh dan lebih cemerlang.

Dalam pelajaran ini, kita akan mendasarkan pada pekerjaan itu dan membuat Anda memikirkan tentang faktor aksesibilitas lainnya, misalnya cara membangun situs web untuk mendukung [pengguna seperti Victor Tsaran](#), yang tidak bisa melihat layar.

Terlebih dahulu, kita akan mendapatkan latar belakang mengenai *teknologi pendukung*, yakni istilah umum untuk alat seperti pembaca layar untuk membantu pengguna yang cacat agar tetap bisa mengakses informasi.

Berikutnya, kita akan mengamati beberapa konsep umum pengalaman pengguna, dan menjadikannya sebagai dasar untuk mendalami pengalaman pengguna teknologi pendukung.

Terakhir, kita akan melihat cara menggunakan HTML secara efektif untuk membuat pengalaman yang baik bagi para pengguna ini, dan bagaimana hal itu sedikit tumpang tindih dengan cara kita menangani fokus sebelumnya.

6.1 Teknologi pendukung

Teknologi pendukung adalah istilah umum untuk perangkat, perangkat lunak, dan alat yang membantu penyandang cacat melakukan suatu tugas. Dalam pengertian paling luas, teknologi ini bisa berupa teknologi rendah seperti kruk untuk berjalan atau kaca pembesar untuk

membaca, atau teknologi tinggi seperti lengan robotik atau perangkat lunak pengenalan gambar di ponsel cerdas.



Teknologi pendukung bisa meliputi sesuatu yang umum seperti zoom di browser zoom, atau spesifik seperti pengontrol game yang didesain khusus. Teknologi ini berupa perangkat fisik terpisah seperti tampilan braille, atau diimplementasikan lengkap dalam perangkat lunak seperti kontrol suara. Teknologi pendukung bisa ditanamkan dalam sistem operasi seperti pembaca layar, atau berupa pengaya seperti ekstensi Chrome.



Garis antara teknologi pendukung dan teknologi secara umum adalah kabur; semua teknologi dimaksudkan untuk membantu orang melakukan tugas atau hal lain. Dan teknologi sering kali bisa masuk ke dan keluar dari kategori "pendukung".

Misalnya, salah satu produk sintesis ucapan komersial paling awal adalah kalkulator bicara untuk tuna netra. Kini sintesis ucapan ada di mana-mana, dari arah mengemudi hingga asisten virtual. Sebaliknya, teknologi yang pada mulanya serba guna sering kali digunakan sebagai pendukung. Misalnya, orang yang lemah penglihatan mungkin menggunakan zoom kamera di ponsel cerdas untuk melihat lebih baik benda kecil di dunia nyata.