



HOW TO BE A WEB DEVELOPER (AND DESIGNER) AND GETTING PAID ONLINE

GALIHPRATAMA.NET
A FREE EBOOK BY GALIH PRATAMA



Ada apa aja?

Di eBook ini ada :

Hey.. Tunggu Sebentar..	4
Siapa sih yang bikin ini eBook?	5
Oh iya.....	7
Gimana sih cara jadi web developer?.....	8
Web Designer (UX/UI Designer)	8
Front-end Web Developer	9
Back-end Web Developer	10
Full-stack Web Developer	11
Perbandingan Front-end dan Back-end	12
Terus, saya harus pilih yang mana?	14
Apa sih yang harus pertama kali saya pelajari dan dimana saya harus belajar?	15
Web Designer.....	15
<i>Belajar User Experience Design</i>	16
<i>Belajar User Interface Design</i>	16
<i>Bedanya UX sama UI itu apa?</i>	17
<i>Belajar Tools Untuk Mendesain</i>	21
Front-end Web Developer	22
<i>Dasar Front End</i>	22
<i>Front-end Developer</i>	26
Back-end Web Developer	38
<i>Belajar Server-Side Programming Language</i>	38

<i>Belajar Database.....</i>	39
<i>Hal lain yang harus dipelajari.....</i>	40
Saya udah bisa nih, terus gimana cari duitnya?	42
Cara Pertama : Berbagi	42
Cara Kedua : Ikutan Kopdar dan Group-Group	44
Cara Ketiga : Mencari Pekerjaan Secara Online	44
<i>Cari Kerjaan di oDesk (atau situs lainnya).....</i>	45
<i>Cari Kerjaan di Group Facebook.....</i>	45
Cara Keempat : Membangun Portfolio / Personal Branding	45
Pertanyaan yang sering ditanyakan	47
Kalau jadi web developer, harus punya Macbook?	47
Ada gak komunitas yang mewadahi web developer?	50
Kalau mau bisa Web Development harus kuliah ga?	50
Tools yang digunakan biasanya apa aja sih?	50
Ada gak sih pengalaman pahit yang mas alami di bidang ini?.. Error!	
Bookmark not defined.	
Saya mau nanya yang lain mas..	50
Makasih ya!.....	51

Hey.. Tunggu Sebentar..

Kamu.. Iya kamu. Saya ngomong samu kok hehe ^_^ By the way, terima kasih sudah berminat membaca eBook pertama saya ini yang berjudul **“How to be a web developer (and designer) and getting paid online”**.

Mudah-mudahan eBook ini bisa berguna untuk kamu-kamu yang ingin memulai bekerja menjadi freelance ataupun bekerja secara remote.

Ebook ini saya dedikasikan semata-mata hanya untuk berbagi kepada masyarakat agar ilmu yang saya dapatkan bisa berguna pula untuk orang lain. Karena orangtua saya sendiri pun pernah mengatakan kepada saya bahwa **“Berbagilah ilmu kepada siapapun, kapanpun, dan dimanapun. Karena ilmu yang kamu dapatkan akan bermanfaat apabila kamu membagikannya kepada orang lain”**.

Saya tidak bisa menjamin setelah kamu membaca eBook ini akan bisa langsung menjadi pintar dan langsung mendapatkan uang. Namun, mudah-mudahan eBook ini bisa menjadi panduan dasar untuk kamu yang masih sangat baru di dunia Web Development ini.

Saya secara pribadi meminta maaf apabila eBook ini masih banyak sekali kekurangan dan kesalahan, karena saya sendiri pun menyadari bahwa saya tidak luput dari sebuah kesalahan.

Saya harap eBook ini tidak diperjualbelikan oleh siapapun dimanapun tanpa seizin saya langsung. Karena eBook ini adalah eBook yang saya berikan gratis kepada umum.

Boleh kok di copas, kalau bisa di improve tulisannya, tapi jangan lupa mencantumkan sumbernya ya!

Boleh juga di-print, asal jangan dijual dan mengklaim bahwa eBook ini adalah buatan kamu, oke sip? ;)

Siapa sih yang bikin ini eBook?

Sebelumnya kita belum kenal ya? Hehe. Kenalan dulu donk :p tak kenal maka tak sayang :p. Saya perkenalannya santai kok ^_^

Nama saya **Galih Pratama**, biasa dipanggil **Galih** sama teman-teman. Saya sendiri masih menginjak bangku kuliah di **Universitas Widyatama** ngambil **S1 Sistem Informasi** dan sekarang memasuki semester 3 (pada saat terakhir eBook ini ditulis).

Nih fotonya kalau penasaran orangnya kayak gimana. Orangnya agak-agak ~~ganteng~~ gendut gitu sih kalau kata orang. #abaikan



Kerjaan saya sehari-hari antara **kuliah**, **ngoding**, **ngedesain**, **makan**, **tidur**, ~~pacaran~~ dan **fesbukan**. Saya biasa mengerjakan project-project Website, mulai dari ~~tugas kuliah temen~~, UKM sampai beberapa project Pemerintahan. Alhamdulillah dari project tersebut saya bisa bayar kuliah sendiri dan beli gadget-gadget impian tanpa minta dari orang tua, walaupun angkanya tidak seberapa, namun harus tetap bersyukur :D

Oh iya, saya gak menutup kemungkinan kalau misalnya kalian mau ngajak saya atau kerjasama bareng saya, saya ~~single~~ available kok untuk kerjasama dengan kalian :D

Punya hobi bikin **video tutorial** yang sering saya upload di channel youtube saya (cari aja di youtube nama saya, yang fotonya bentuk kepala berkecamata). Kadang-kadang juga suka bikin artikel blog atau bikin eBook (kayak yang sekarang ini kamu baca!).

Kesibukan saya lainnya yaitu bisnis **Hosting dan Domain** di **DayaHosting** dan **Modul Tutorial Android Lengkap** (yang sebenarnya kalau boleh jujur itu hasil kumpulan tugas-tugas kuliah dulu di D1 ITB :p). Dan rencananya bakal bikin situs belajar Koding dan Desain juga di **BelajarKoding.net**.

Malah jadi curhat nih haha, pokoknya kalau yang mau kepoin saya tinggal add/follow aja fb saya di fb.me/pratamagalih, twitter saya [@galpratama](https://twitter.com/galpratama) atau kontak email saya di hi@galihpratama.net ^ _ ^



Oh iya, kalau mau ngobrol-ngobrol bareng, langsung aja bisa ngobrol di Facebook, atau kalau mau kopdar juga boleh kok, tinggal kontak

saya aja di Facebook, kalau saya lagi santai hehe . Kadang-kadang sih saya sering ikut kopdar-kopdar freelancer gitu atau kopdar internet marketer, buat kamu yang mau ikutan juga silahkan sekalian ~~nambah gebetan~~ silaturahmi juga yuk!

Oh iya...

Kalau kamu memang merasa eBook ini bermanfaat buat kamu, kamu **boleh kok donasi ke saya**, tapi tenang aja saya gak maksa kok!

Saya harap sih kamu baca ebook ini keseluruhan, kalau bisa sih sekalian sambil **ngasih kritik dan saran** buat eBook saya ini, karena eBook ini menurut saya masih banyak banget kekurangannya, dan feedback kamu sangat berguna buat eBook ini!

Kalian bisa donasi ke saya lewat **BCA (1392837315)**, **BNI (0274351081)** ataupun **PayPal (hanamura.iost@gmail.com)**. Mau donasi berapapun, seratus perak pun saya terima kok hehe ^_^

Tapi tenang aja, eBook ini **bener-bener gratis** lho ya! Jadi kamu gak usah kuatir harus bayar atau gimana, **100% free** ^_^ Semoga dengan ebook ini kalian bisa mendapatkan ilmu!

Oh iya satu lagi, jangan lupa ya buat **subscribe** di channel **Youtube** saya atau bisa **stay tune** di **BelajarKoding.net**, biar nanti kalau ada info-info terbaru kayak misalnya ada eBook baru, nanti bisa saya share ke kamu!

Gimana sih cara jadi web developer?

Saya anggap untuk kamu yang baca bagian ini sama sekali buta arah dengan dunia web development atau masih bingung soal dunia persilatan web development. Kalau kamu yang sudah mengerti mungkin bisa langsung skip saja bagian ini (atau sekalian ga usah baca eBooknya :p #becanda)

Sebenarnya jadi web developer itu mudah kok, asal **mau belajar** dan **belajar dari sumber yang up-to-date**. Selain itu kamu juga harus **paksain** untuk **belajar materi berbahasa inggris**. Kenapa? Karena materi soal web development itu banyak banget yang bagus yang rata-rata berbahasa inggris.

Kali ini saya coba untuk jelaskan dulu apa sebenarnya web developer dan istilah-istilah yang kadang membingungkan untuk kamu.

Web Developer yaitu seseorang yang membangun sebuah Website. Web Developer pada umumnya dapat dibagi kepada beberapa bagian :

- [Web Designer \(UI/UX Designer\)](#)
- [Front-end Web Developer](#)
- [Back-end Web Developer](#)
- [Full-stack Web Developer](#)

Sebenarnya selain diatas masih banyak bagian-bagian lainnya, namun saya hanya akan membahas yang ada diatas saja (karena kalau saya kasih semua juga nanti malah jadi pusing bacanya :p)

Web Designer (UX/UI Designer)

Web Designer merupakan seseorang yang mendesain sebuah website. Maksud dari **mendesain** ini adalah sang designer ini yang membuat gambaran website akan seperti apa (wireframe) dan terlihat seperti apa (mockup).



Web Designer biasanya bekerja menggunakan tools seperti **Adobe Photoshop, Adobe Illustrator, Sketch (Mac)**, dan tools-tools lainnya. Mereka pun biasanya menggambar terlebih dahulu di kertas sebelum mereka membuat desainnya di aplikasi.

Saya sendiri lebih suka memanggilnya sebagai **UX/UI Designer** (walaupun dalam konteksnya UX/UI ini tidak hanya di dalam website saja).

Web Designer ini pada dasarnya **sama sekali tidak bermain dengan kode**. Namun beberapa Web Designer memiliki keahlian **HTML** dan **CSS** bahkan **Javascript** (walaupun sebenarnya untuk keahlian ini biasanya wajib dimiliki **Front-end Web Developer**)

Front-end Web Developer

Front-end Web Developer merupakan seseorang yang mengerjakan kode layout sebuah website yang sebelumnya layout dan gambaran websitenya sudah dikerjakan oleh **Web Designer**.



Seorang Front-end Developer merupakan seseorang yang mengurus segala hal yang terjadi pada bagian website yang diakses dan digunakan oleh client (client-side).

Front-end Web Developer setidaknya harus memiliki keahlian di **HTML**, **CSS** dan **Javascript**. Ketiga keahlian tersebut merupakan fondasi penting untuk seorang Front-end Developer.

Akan menjadi nilai tambah apabila seorang Front-end Web Developer memiliki keahlian untuk menggunakan **Framework** seperti **Bootstrap** dan **jQuery**.

Terkadang seorang Front-end Web Developer ini merangkap tugas menjadi Web Designer.

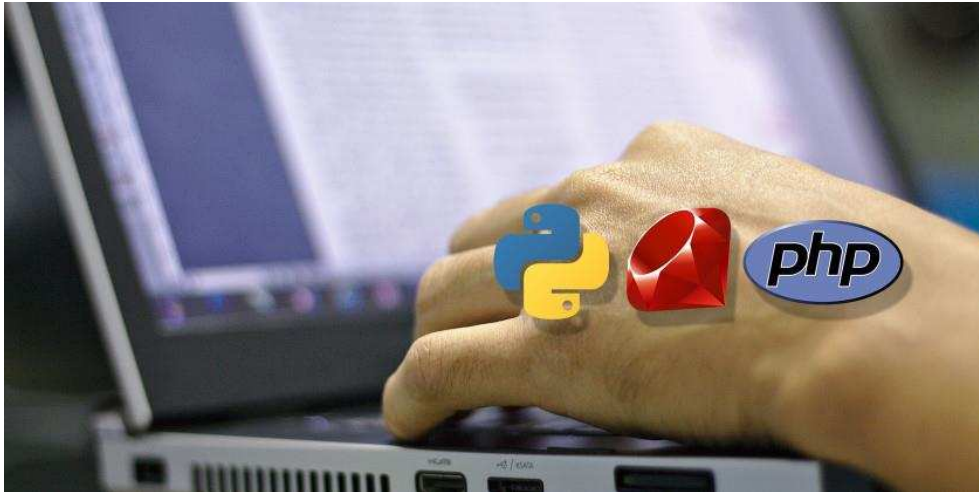
Front-end Web Developer ini biasanya tidak mengurus hal seperti **Database** atau bahasa pemrograman **PHP**, walaupun iya hanya untuk menampilkan data yang telah dibuat oleh Back-end. Bagian tersebut biasadikerjakan oleh Back-end Web Developer.

Back-end Web Developer

Back-end Web Developer merupakan seseorang yang mengerjakan sistem bagaimana sebuah website bekerja. Sistem website ini merupakan sistem yang berjalan di dalam suatu server (server-side).

Back-end Web Developer-lah yang bermain dengan bahasa pemrograman server (server-side language) seperti **PHP**, **JSP**, atau

ASP. Mereka pula-lah yang biasanya mengurus urusan Database seperti MySQL, Oracle, dan lain lain. (Pada perusahaan besar biasanya pekerjaan ini dikerjakan oleh Database Analyst)



Seorang **Backend Developer** harus memiliki dan memahami konsep algoritma yang kuat (yang biasanya dipelajari pada awal-awal semester pada kuliah) dan beberapa konsep-konsep pemrograman (seperti OOP atau Object Oriented Programming)

Back-end Developer pun harus bisa memastikan bahwa aplikasi (website) yang ia buat aman dari serangan-serangan orang yang tak bertanggung jawab.

Full-stack Web Developer

Full-stack Web Developer bisa dikatakan sebagai **Superman** atau **Single Fighter**. Hal tersebut disebabkan karena seorang Fullstack Developer harus mengerti dan mengerjakan semua yang dikerjakan oleh **Web Designer**, **Front-end** dan **Back-end Developer**.

Pada banyak kasus di perusahaan (terutama di UKM dan perusahaan non-startup), seseorang yang bekerja menjadi web developer terkadang menjadi **Full-stack Web Developer** dikarenakan SDM yang terbatas (termasuk gajinya T_T).

Perbandingan Front-end dan Back-end

Pasti kamu masih bertanya-tanya apa sih perbedaan antara Front-end dan Back-end, walaupun sudah saya jelaskan sebelumnya di atas :p

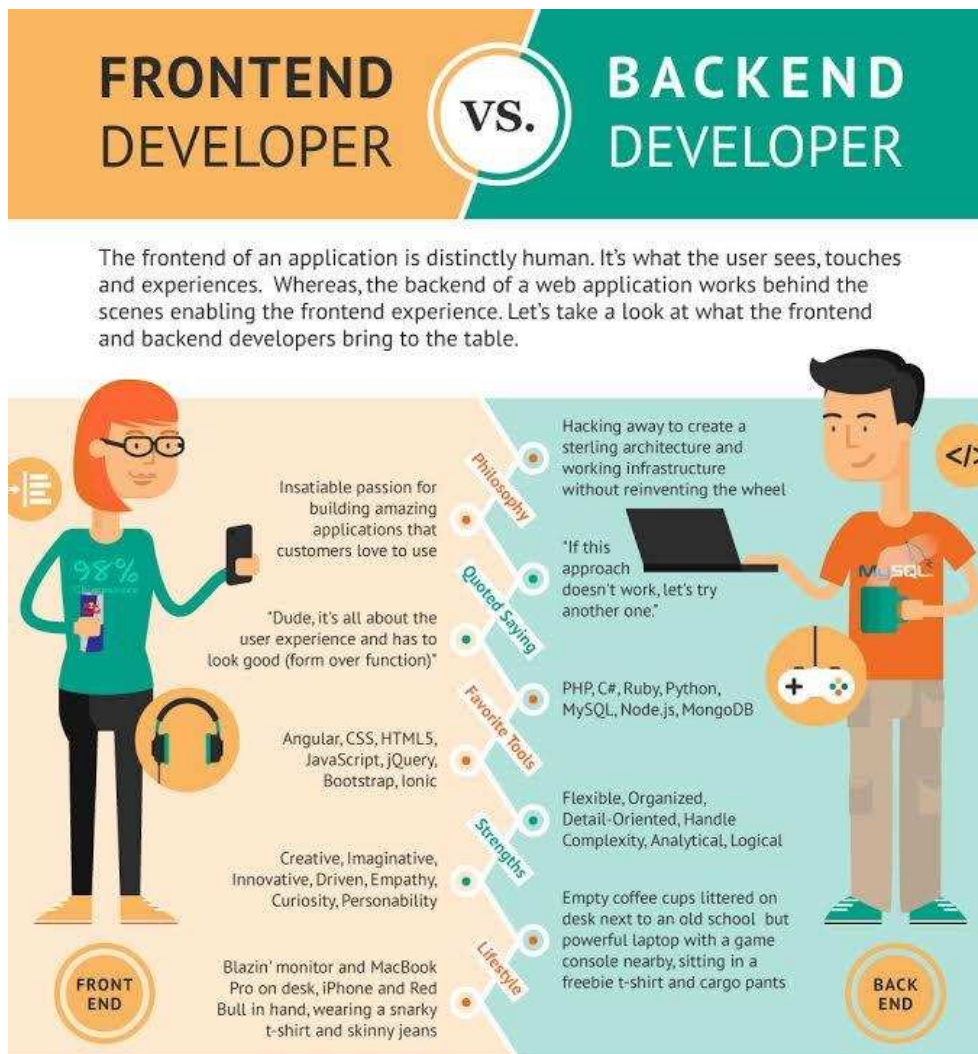
Seperti “kata” nya, **Front** artinya depan dan **Back** artinya belakang. Front-end mengurus hal-hal yang ada di depan dan terlihat. Sedangkan Back-end mengurus hal-hal yang ~~gaib~~ tidak terlihat dan berjalan di sistem. Tapi bukan berarti kalian ngerjainnya membelakangi monitor ya :p



Keduanya saling bergantung satu sama lain. Front-end dan Back-end developer akan bekerja bersama-sama untuk membuat suatu aplikasi tersebut berjalan dengan baik.

Namun terkadang antara Front-end dan Back-end biasanya akan ada konflik dan perbedaan pendapat yang akan sering terjadi saat kalian akan membuat suatu website. Contohnya? Silahkan alami sendiri ya, nanti juga kalian bakal tau sendiri kok :p

Berikut ini merupakan infografis gambaran perbedaan Front-end Developer dan Back-end Developer



A PERFECT & NECESSARY COMBO

The two parts have to work together seamlessly in order for your app to work. It is an alliance between the two that will push your apps forward. Therefore there is no versus when it comes to frontend and backend development. You simply need them both in order to create a fantabulous user experience for your customers.

For more information on how to make your backend work seamlessly with your Angular frontend, visit Backand.com

Back&

Terus, saya harus pilih yang mana?

Tergantung keinginan dan kemampuan kamu. Kalau saya sendiri bisa dikategorikan sebagai **Full-stack Web Developer** (itu pun dikarenakan tuntutan pekerjaan pada saat dahulu bekerja). Namun secara pribadi saya lebih senang menjadi **Front-End Web Developer** atau **Web Designer**.



Apabila kelebihan kamu di bagian logika, hitung-hitungan, atau kamu senang mengerjakan suatu sistem, ada baiknya kamu mengambil **Back-end Web Developer**

Namun, apabila kamu lebih senang mengurus bagaimana sebuah website bekerja yang diakses dan digunakan oleh client, kamu disarankan untuk mengambil **Front-end Web Developer**.

Untuk kamu yang tidak mau berurusan dengan ribuan kode dan kamu memiliki kemampuan di bagian desain, kamu cocok untuk mengambil **Web Designer**.

Untuk kamu yang lebih senang ke bagian visual kamu disarankan untuk mengambil **Web Designer** atau **Front-end Web Developer**.

Pilihlah yang sesuai dengan passion kamu. Do what you love and love what you do.

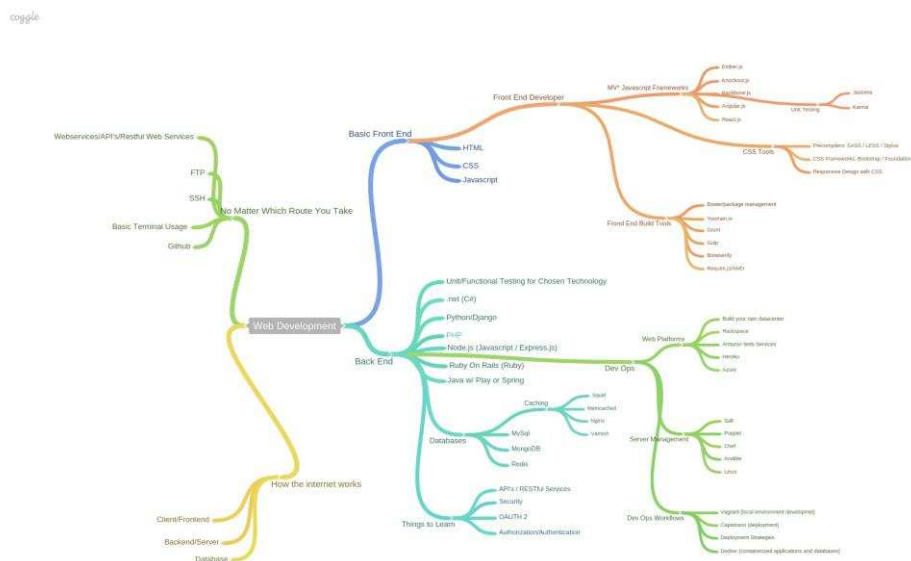
Apa sih yang harus pertama kali saya pelajari dan dimana saya harus belajar?

Setiap fokus yang akan kamu ambil akan berbeda-beda ilmu yang akan kamu pelajari. Namun, ilmu-ilmu tersebut akan saling berhubungan satu sama lain. Semakin banyak ilmu yang kamu pelajari akan semakin baik pula untuk kamu, asalkan **fokus!**

Untuk semua tulisan dibawah ini, saya akan bertumpu kepada mind-map dibawah ini, namun akan saya kembangkan lagi agar kalian-kalian lebih mudah pahamnya

Link ke mindmapnya :

<https://coggle.it/diagram/52e97f8c5a143de239005d1b/56212c4e4c505e0045c0d3bda59b77e5977c2c9bd40f3fd0b451bdcf8da4aa52>

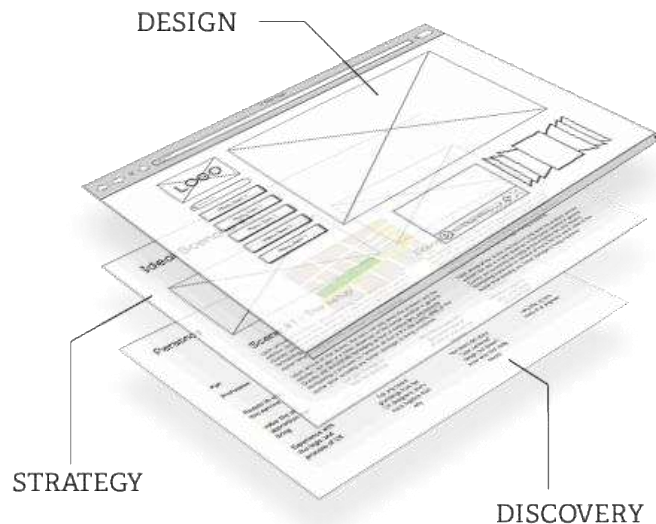


Web Designer

Untuk kamu yang ingin jadi (dan tidak terbatas pada) **Web Designer**, beberapa hal yang harus kamu pelajari yaitu :

Belajar User Experience Design

User Experience Design atau yang biasa disebut **UX Design** yaitu adalah:



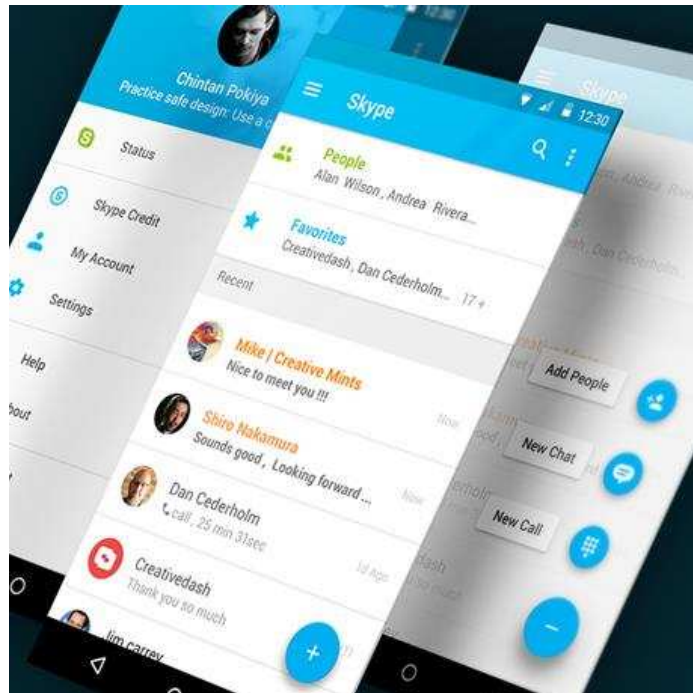
Proses meningkatkan kepuasan pengguna (pengguna aplikasi, pengunjung website) dalam meningkatkan kegunaan dan kesenangan yang diberikan dalam interaksi antara pengguna dan produk

Bahasa gampangnya, **UX Design** itu proses membuat sebuah website atau aplikasi yang kamu buat menjadi mudah untuk digunakan dan tidak membingungkan ketika digunakan oleh pengguna.

Dasar dari **User Experience Design** ini bisa kamu pelajari di situs **uxapprentice.com** (berbahasa inggris) atau di **uniteux.com** (situs blog berbahasa indonesia)

Belajar User Interface Design

User Interface Design atau yang bahasa Indonesianya itu **Desain Antarmuka Pengguna** adalah :



Desain antarmuka untuk mesin dan perangkat lunak, seperti komputer, peralatan rumah tangga, perangkat mobile, dan perangkat elektronik lainnya, dengan fokus pada memaksimalkan pengalaman pengguna.

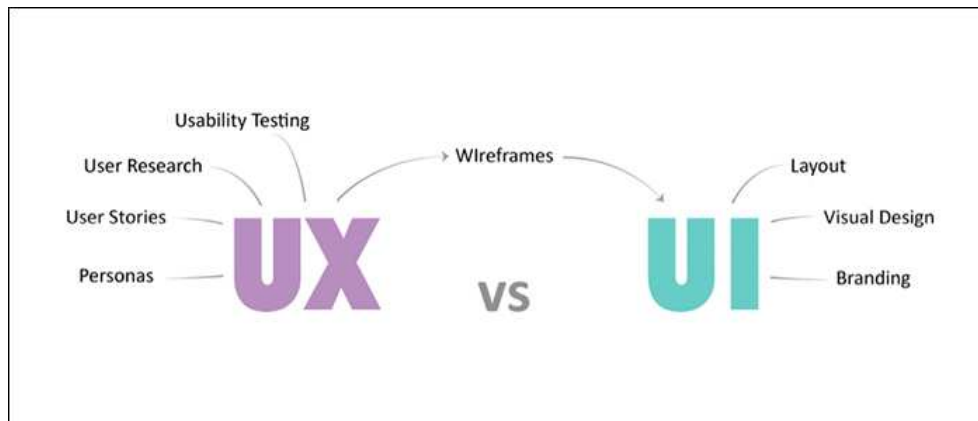
Bahasa gampangnya yaitu **UI Design** adalah bagaimana suatu website atau aplikasi yang kamu buat terlihat seperti apa. Orang biasa menyebutnya sebagai tampilan atau desain sebuah website.

Kamu bisa mendapatkan inspirasi desain dari **behance.com**, **dribbble.com** atau **webdesignserved.com**.

Bedanya UX sama UI itu apa?

Banyak orang yang salah kaprah bahwa UI sama UX itu adalah suatu hal yang sama. Pada faktanya UX dan UI itu berbeda, namun satu sama lain saling berhubungan.

Dibawah ini merupakan perbedaan antara User Interface dan User Experience (dikutip dari thecdm.ca)

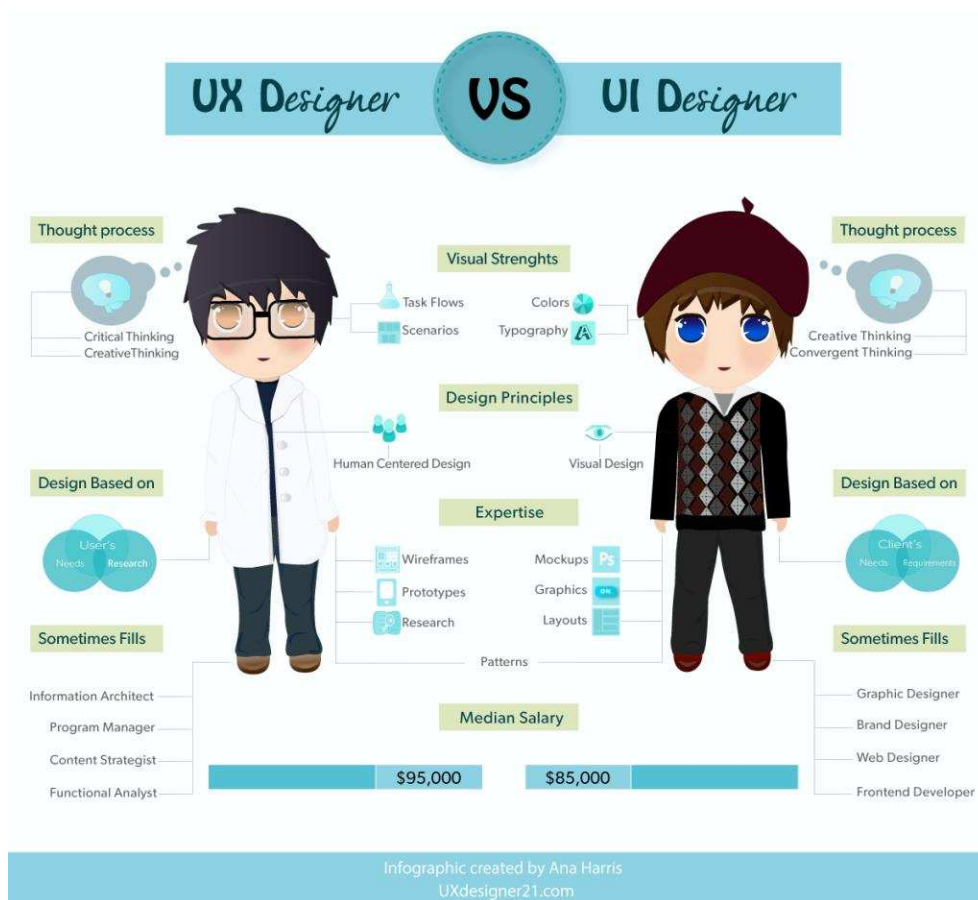


Pada dasarnya, **User Experience** adalah tentang “memahami pengguna”. Tujuan UX adalah mencari tahu siapa mereka, apa yang mereka capai dan apa cara terbaik bagi mereka untuk melakukan “sesuatu”.

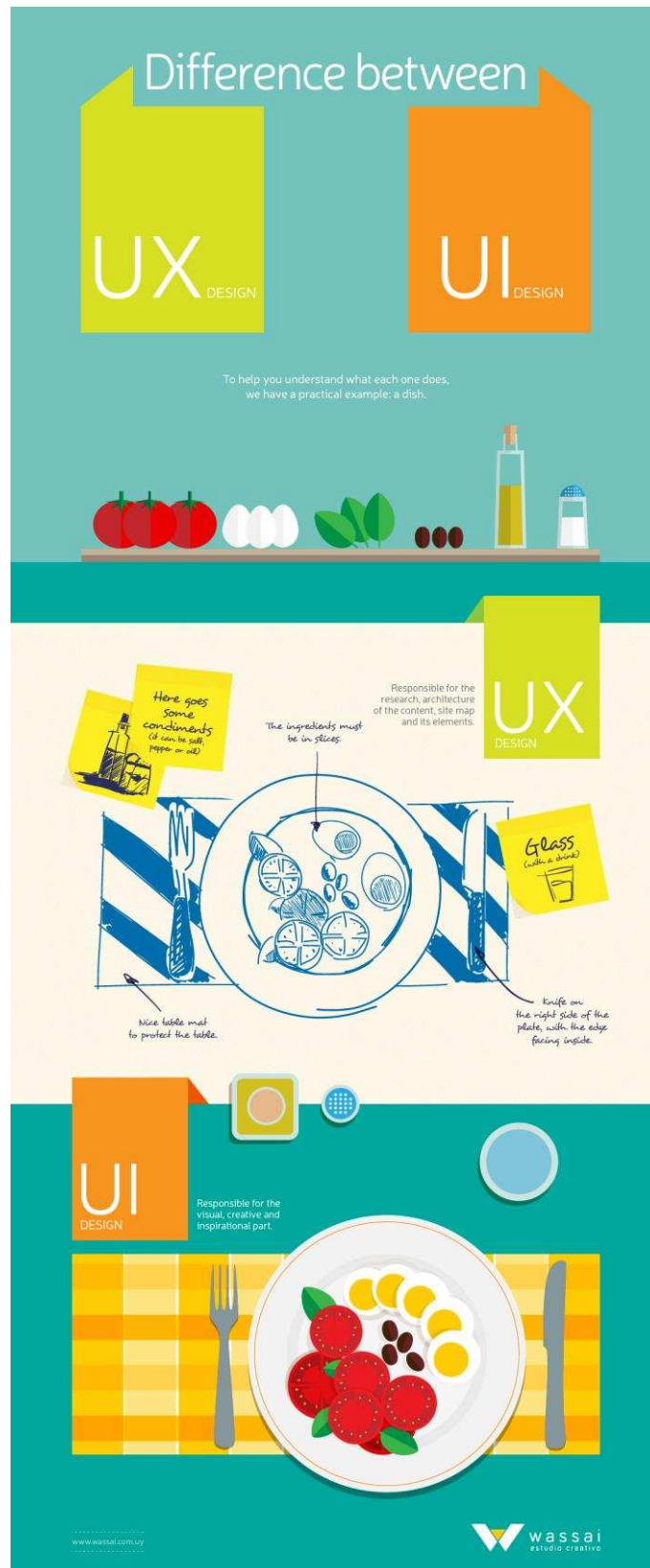
UX berkonsentrasi pada bagaimana sebuah produk terasa dan apakah itu memecahkan masalah bagi pengguna.

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

Untuk referensi, saya merekomendasikan buku **Don't Make Me Think** untuk kamu baca.



Masih bingung juga? Kamu bisa cari-cari informasi di Google dengan mengetikkan **“UX vs UI”** atau **“difference between UX and UI”**. Kalau masih bingung juga, tidur dulu deh atau ga cari ~~pekerjaan~~ makan dulu baru nanti baca eBook ini lagi :p



Belajar Tools Untuk Mendesain

Pastinya kamu harus belajar tools-tools untuk mendesain. Banyak tools yang bisa kamu gunakan untuk membuat sebuah Desain Website.

Biasanya para designer menggunakan **Adobe Photoshop** sebagai toolsnya. Ada pula yang menggunakan **Adobe Illustrator** sebagai toolsnya.

Bahkan untuk pengguna Mac ada tools yang bernama **Sketch** yang bisa digunakan untuk membuat desain.

Perlu diperhatikan ketiga aplikasi yang saya sebutkan diatas adalah tools berbayar. Ada yang gratis? Pastinya ada dong!

Kamu bisa pakai tools seperti **Inkscape** yang open-source dan dapat diunduh secara gratis. Adapun tools seperti **GIMP** (namun saya tidak terlalu merekomendasikannya karena agak sulit digunakan) yang bisa kamu pakai.

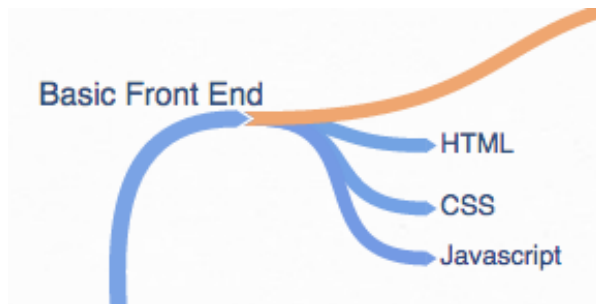
Untuk mempelajari semua diatas, kamu bisa belajar di **Youtube** kok! Banyak tutorial bertebaran disana dan yang pasti tutorial disana rata-rata berbahasa inggris. Jadi, ayo asah skill berbahasa inggris kamu!

Saya akan coba berikan beberapa referensi untuk kamu yang ingin belajar lewat youtube.

- **Dasar-dasar Tools Photoshop Untuk Web Designer**
<https://www.youtube.com/watch?v=Bjj7itEWpW4>
- **Adobe Photoshop CS6 Tutorial for Beginners**
<https://www.youtube.com/watch?v=7ZlXagXwcn4>
- **Learning Photoshop from Scratch**
<https://www.youtube.com/watch?v=VSEQ3NTxJGk>

Front-end Web Developer

Untuk Front-end Developer, ada beberapa tahapan yang dapat kamu pelajari terlebih dahulu :



Dasar Front End

Belajar HTML

Salah satu ilmu yang sangat wajib harus dipelajari oleh front-end developers yaitu mempelajari HTML. Bagi yang belum tau HTML, kepanjangannya itu adalah HyperText Markup Language. Perlu diketahui HTML itu bukanlah bahasa pemrograman, tapi bahasa markah.

Susunan penulisan HTML itu biasanya ditandai dengan buka tag dengan dikelilingi oleh "<>" dan ditutup oleh "</>". Contohnya seperti :

```
<strong>Halo, saya tebal loh!</strong>
```

akan menghasilkan seperti ini di browser :

Halo, saya tebal loh!

Ngerti kan intinya? Masih gampang kok kalau HTML itu kok!

Belajar CSS

Nah, bisa dibilang kalau CSS itu merupakan make-up nya HTML. CSS ini gunanya untuk mempercantik halaman website-mu yang sudah dibuat dan ditulis menggunakan HTML.

Sebelum adanya CSS ini, semua “hal-hal yang membuat manis” ini dilakukan di HTML langsung. Maksudnya gimana tuh?

Jadi, misalnya kalau mau ingin ngewarnain background halaman, caranya dengan menambahkan property :

`<body background="red"> atau `

Nah, beda cerita kalau kamu menggunakan CSS, kamu bisa melakukannya dengan:

index.html

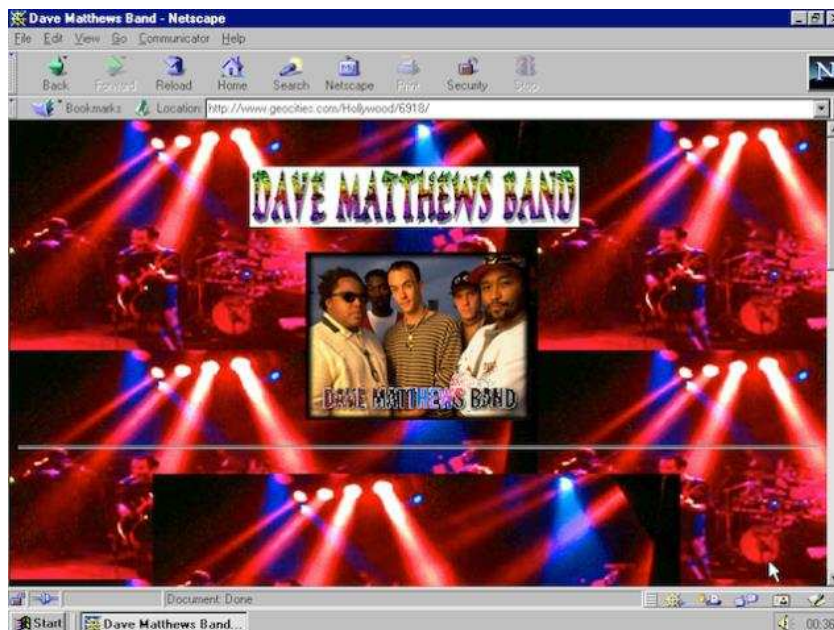
`<body> <p class="kamu"> Kamu.. Iya kamu.. </p> <body>`

style.css

`body { background-color: #ff0000; }
p.kamu { font-family: Helvetica,Arial,sans-serif; }`

CSS ini bakal kamu pakai gak cuma untuk mempermanis tampilan website, karena sejatinya di modern web development CSS adalah hal yang sangat penting.

Dan CSS ini memecahkan masalah dalam web development dahulu, yang dimana kalau dulu mau mempercantik tampilan website itu sangat sangat terbatas dan pusing.



Yap, website yang kamu lihat diatas merupakan salah satu contoh website yang dibuat dahulu, saat dimana Geocities masih berjaya (dan sekarang sudah ditutup oleh pihak Yahoo).

Belajar Javascript

Masuk ke Javascript, kamu mau gak mau harus memahami dasar pemrograman. Setidaknya kamu memahami **algoritma** terlebih dahulu sebelum kamu menyentuh bahasa ini. Kalau kata google sih **Algoritma** itu artinya :

“Algoritma secara sederhana merupakan urutan langkah- langkah logis untuk menyelesaikan masalah yang disusun secara sistematis.”

Campangnya algoritma itu seperti susunan-susunan rencana hidupmu dan cara menjalankannya dalam sehari. Misalnya, pada jam 5 kamu bangun untuk menunaikan shalat subuh, setelah itu jam 6 kamu sarapan. Kurang lebih algoritma itu seperti itu.


```
// life motto
if (sad() === true) {
  sad().stop();
  beAwesome();
}
```

Baru, setelah itu kamu pahami **Javascript** itu. Oh ya inget kalau **Javascript** itu **BUKAN Java**. Terus kenapa namanya **Javascript**? Itu semata-mata untuk alasan “marketing” agar bahasa Javascript itu terkenal (kamu bisa baca sejarahnya di google). sama sekali gak ada hubungannya dengan **Java**!

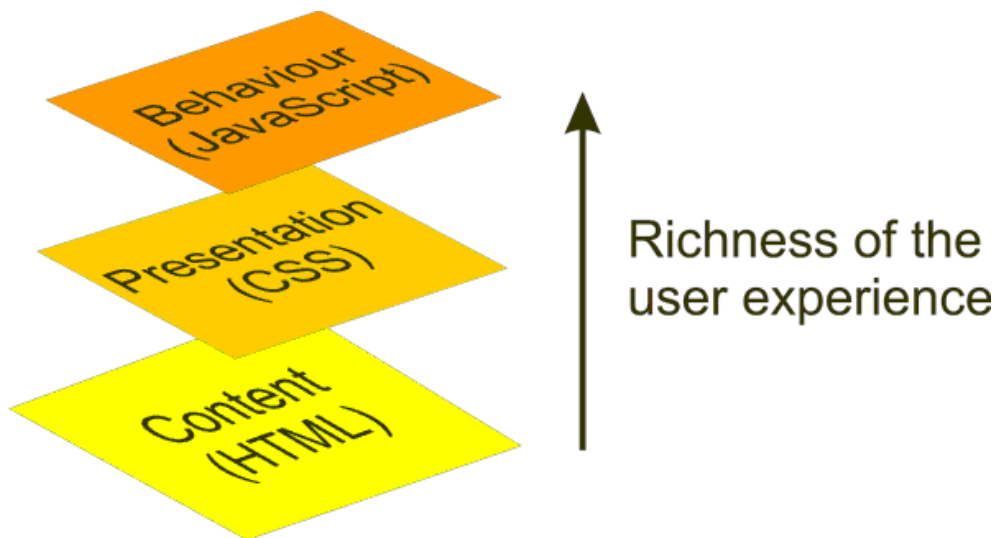
Oh ya jangan lupa pelajari jQuery juga ya setelah kamu paham akan Javascript.

Apa saya harus bisa semuanya? Banyak bener! Pusing saya mas!

Kalau kamu memang ingin menjadi seorang Front-end developer, ya mau gak mau kamu harus paham ketiga tersebut. HTML, CSS dan Javascript merupakan pondasi seorang Front-end Web Developer.

Dibilang pusing sih iya tapi kalau udah paham gak sesulit itu kok, bahkan dibilang **asyik**!

Ngomong-ngomong, itu baru dasarnya loh! Bukannya nakut-nakutin, tapi kalau kamu sudah terjun di dunia Web Development, kamu harus siap dengan arus kencang perkembangannya. Sedikit ketinggalan aja misalnya 2-3 tahun, kamu bakal kudet banget.



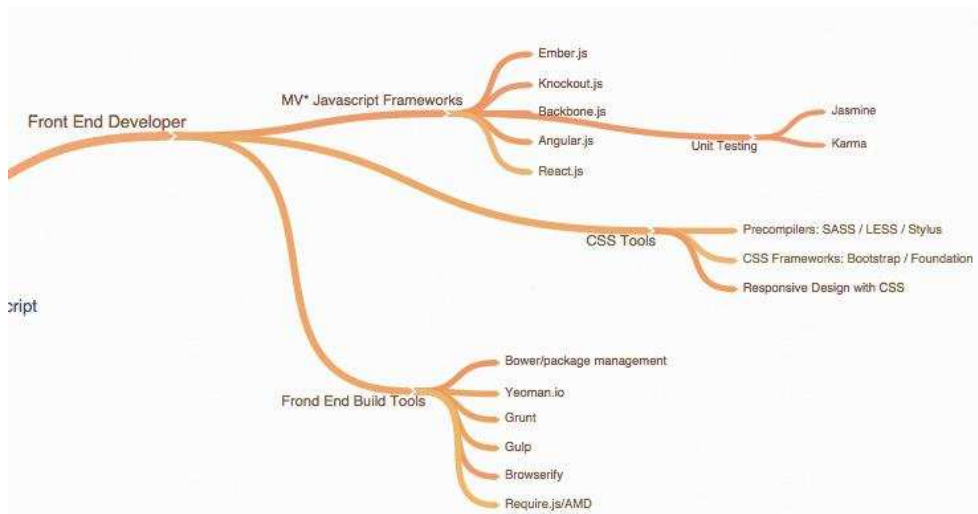
Terus, Dimana Belajarnya?

Sebenarnya ada banyak banget referensi untuk belajar website, dari yang gratis maupun yang berbayar. Tapi sih untuk kamu yang ingin memahami konsep dasar kamu bisa coba belajar di **W3Schools**. Disana kamu bisa belajar dasar-dasar tentang **HTML**, **CSS** maupun **Javascript**.

Saya pun mempunyai rencana untuk membuat modul pembelajaran soal Web Design maupun yang lainnya, namun sampe saat eBook ini ditulis, modulnya belum beres 😞. Jadi bagi kamu yang penasaran stay tune aja nanti di **BelajarKoding.net** atau tinggal follow Facebook saya, mudah-mudahan saya bakal informasikan kalau sudah ada

Front-end Developer

Nah, akhirnya setelah kamu memahami ketiga hal tersebut (ingat dipahami, bukan dihapalkan, karena nanti kamu pun akan hapal dengan sendirinya), baru deh masuk ke **dunia yang sebenarnya** – jeng jeng jeng jeng!



Waduh banyak bener mas? Gak salah? Sebenarnya kalau kamu gak memahami semua hal diatas sih gak masalah. Kamu cukup perlu pelajari yang menurut kamu penting saja, karena saya pun belum memahami semua hal yang digambarkan diatas (ngerti Angular aja belum saya mah #DaAkumahApaAtuh) :p

Yaudah, saya jelaskan ya apa saja yang harus dipelajari untuk menjadi Front-end Developer sesungguhnya

Belajar Tools CSS

Nah, untuk kamu yang ingin meningkatkan produktivitas dan kemampuan kamu, kamu bisa menggunakan tools CSS yang bermacam-macam.

Precompilers

“Precompilers itu apaan sih? Perasaan daritadi aneh-aneh istilahnya ih mas -_-”

Istilah gampangnya, precompilers itu adalah suatu tools yang mengubah kode yang kamu buat menjadi kode yang dapat dibaca oleh suatu aplikasi (dalam kasus ini, aplikasinya adalah Browser)

Beberapa precompiler yang bisa kamu gunakan itu ada bermacam-macam, namun kebanyakan orang menggunakan **SAAS/SCSS** atau **LESS**.

“Terus, bedanya sama CSS apa?”

Bisa dibilang precompiler ini adalah **alat bantu** nya front-end web developer. Kalau kata si Teteh **SASS** sih dia bilangnyanya gini, dan termasuk kelebihan dari si Teteh **SASS** :



Sass makes CSS fun again. Sass is an extension of CSS3, adding nested rules, variables, mixins, selector inheritance, and more. It's translated to well-formatted, standard CSS using the command line tool or a web-framework plugin.

Kalau kamu yang biasa berkutat di programming apalagi di programming berbasis OOP, pasti langsung seneng liat tulisan diatas. Kalau kamu yang gak ngerti pasti pusing kan? :p

Intinya, preprocessor ini buat ngebantu kalian buat bikin CSS lebih enak buat ditulis dan di maintenance. Karena CSS saja tidak cukup.

Bayangkan aja kalau kalian punya aplikasi segede Facebook, terus kode stylingnya hanya dimaintenance dengan menggunakan CSS saja, pastinya kepala kalian akan **pusing pala barbie** untuk ngatur2 semuanya.

CSS Framework

Nah kalau untuk bagian ini kayaknya kebanyakan dari kalian udah sering denger. Tau apa itu **Twitter Bootstrap** kan? Itu adalah salah satu CSS Framework

CSS Framework ini sendiri adalah lagi lagi merupakan **tools** untuk memudahkan kita membuat sebuah website. Karena Bootstrap ini sudah memiliki **style** dasar yang bisa kita gunakan langsung



“Widihh.. Enak banget dong mas? Jadi saya ga usah ngedesain lagi dong?”

Ya ga gitu juga kalee.. Dengan **CSS Framework** ini, kalian akan bisa lebih cepat untuk membuat aplikasi. Dan juga lebih cepat untuk mengubah-ubah styling yang kamu inginkan. Karena dasar (skeleton)

dari stylingnya sendiri udah ada. Kalian tinggal ubah-ubah atau bikin baru saja.

Awas kalau udah sering pake Framework keenakan nanti kreativitasnya berkurang :p

Nah, macam-macam framework itu ada banyak. Namun yang paling sering digunakan itu adalah **Bootstrap** dan **Foundation**.

Belajar Responsive Web Design (RWD)

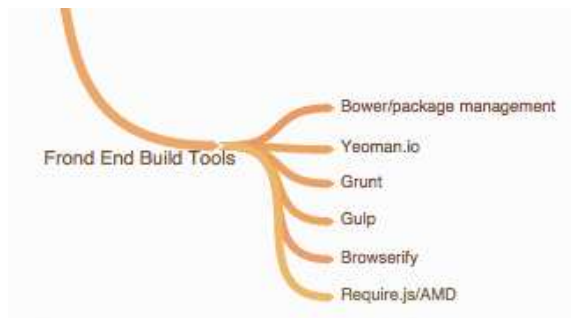
Kamu pernah lihat kan suatu website, pas kamu buka di Mobile (Android, iOS) ternyata tampilannya masih bagus (gak hancur)? Nah teknik tersebut namanya Responsive Web Design



Teknik **RWD** ini merupakan hal yang menurut saya **sangat penting** karena sekarang jamannya sudah serba mobile, masa website kamu gak bisa diakses di mobile? Kan kebangetan :p

Cara membuat website menjadi repsonsive ini bisa kamu pelajari kalau kamu telah memahami CSS, atau kalau kamu menggunakan CSS Framework bawaannya sudah **Responsive** kok!

Belajar Front-end Build Tools



Front-end Build ini gak beda jauh dengan yang saya bahas sebelumnya, yaitu intinya untuk mempermudah dan mempercepat workflow kita selagi bekerja.

Package Management (Bower / NPM)

Bagi kalian yang belum tahu **package management**, kurang lebih itu adalah sistem dimana jika kita membutuhkan suatu library (misalnya Bootstrap, atau jQuery) kita tinggal panggil aja lewat **Bower**.



Kok logonya burung? Yaiyalah yang punya juga Twitter :p

Kan ribet tuh kalau misalnya kita harus download, extract, copy. Butuh step extra buat melakukan hal tersebut dan memakan waktu juga. Lain hal kalau misalnya kamu tinggal panggil aja si **Bootstrap** atau **jQuery** lewat command-line dengan cara

> bower install bootstrap

Command diatas tuh kayak kita manggil dan nyuruh si Mamang **Bower** buat pasangin jQuery sama Bootstrap.



***“Mang, bawain Bootstrap yang paling baru kesini ya.
Jangan lupa ditaro di folder public/”***

Kita bisa nyuruh si Mamang **Bower** mau dimana dipasangnya, misalnya ditaro di folder tertentu. Atau versi tertentu, bisa juga kok! Kita pun bisa nyuruh si Mamang **Bower** ini buat pasangin sesuatu, kita tinggal kasih link **GitHub** nya aja, nanti si Mamang ngebantuin kita.

Si Mamang **Bower** ini juga bisa ngebantuin kita buat update kalau misalnya si **Bootstrap** atau **jQuery** nya udah jadul. Tinggal suruh aja

> bower update <package>

Baik banget ya ternyata si Mamang Bower ini!

Gulp / Grunt

Nah, gak kalah dengan Mamang **Bower**, kang **Grunt** atau kang **Gulp** (fungsinya sama aja, cuma beda produk) ini juga ga kalah baiknya.

```
module.exports = function(grunt) {

  grunt.initConfig({
    jshint: {
      files: ['Gruntfile.js', 'src/**/*.js', 'test/**/*.js'],
      options: {
        globals: {
          jQuery: true
        }
      }
    },
    watch: {
      files: ['<del>src</del> jshint.files %>'],
      tasks: ['jshint']
    }
  });

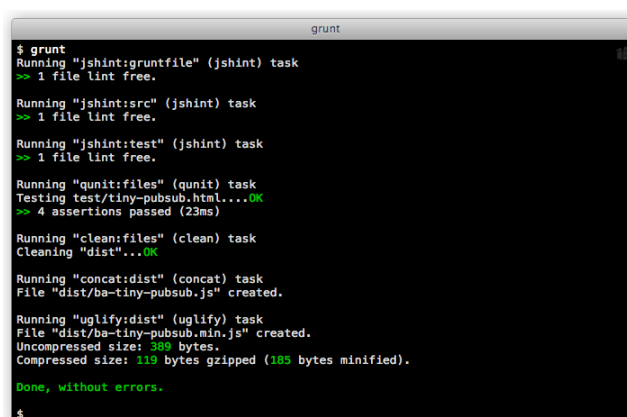
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-contrib-watch');

  grunt.registerTask('default', ['jshint']);

};
```

Akang **Grunt** ini juga bisa ngebantuin kita. Kalau kita pengen dibantuin sama kang **Grunt** ini, kita tinggal bikin aja **Gruntfile** buat dikasih ke si kang **Grunt** ini, nah nanti kalau udah tinggal dipanggil aja tuh.

Manggil akang grunt itu gampang, tinggal ketik aja **grunt** di command line, nanti dia juga datang sendiri udah gitu ngebantuin



```
grunt
$ grunt
Running "jshint:gruntfile" (jshint) task
>> 1 file lint free.

Running "jshint:src" (jshint) task
>> 1 file lint free.

Running "jshint:test" (jshint) task
>> 1 file lint free.

Running "qunit:files" (qunit) task
Testing test/tiny-pubsub.html....OK
>> 4 assertions passed (23ms)

Running "clean:files" (clean) task
Cleaning "dist"...OK

Running "concat:dist" (concat) task
File "dist/ba-tiny-pubsub.js" created.

Running "uglify:dist" (uglify) task
File "dist/ba-tiny-pubsub.min.js" created.
Uncompressed size: 389 bytes.
Compressed size: 119 bytes gzipped (185 bytes minified).

Done, without errors.
$ --
```

Yeoman (Yo)

Nah lain hal dengan akang **Grunt** dan mamang **Bower**, pak **Yeoman** ini Juragannya. Jadi pak **Yeoman** ini bisa dibilang bisa nyuruh si akang **Grunt** dan mamang **Bower** buat bantuin kita tanpa kita harus nyuruh satu-satu .

Kata si pak Yeoman juga dia bilang begini :

Tools

The Yeoman workflow comprises three types of tools for improving your productivity and satisfaction when building a web app: the scaffolding tool (yo), the build tool (Grunt, Gulp, etc) and the package manager (like Bower and npm).



yo scaffolds out a new application, writing your Grunt configuration and pulling in relevant Grunt tasks and Bower dependencies that you might need for your build.



The **Build System** is used to build, preview and test your project. [Grunt](#) and [Gulp](#) are two popular options.



The **Package Manager** is used for dependency management, so that you no longer have to manually download and manage your scripts. [Bower](#) and [npm](#) are two popular options.

Ini sebenarnya buat apa sih? Simple saja, untuk mempermudah ada dalam mengerjakan suatu project atau apps.

Belajar MV* Javascript Framework

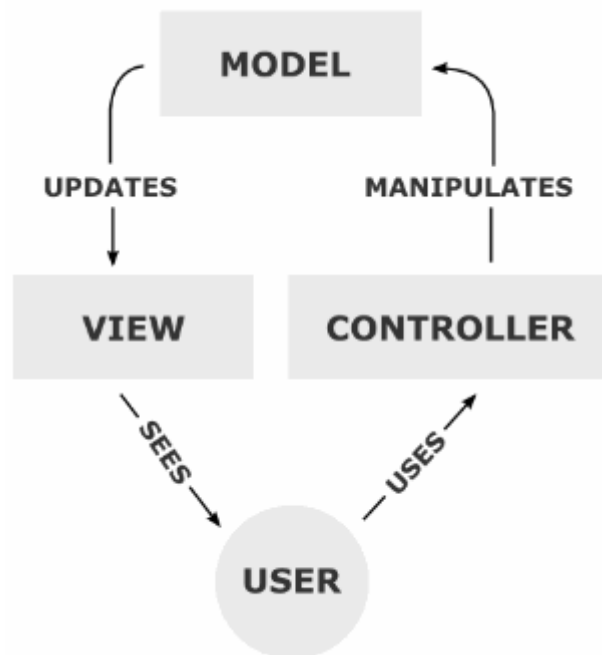
Apa lagi itu mas? Kepala saya pusing bacanya! MV MV

apa pula itu, Video Klip Korea maksudnya mas?

Hahaha, saya jelasin dulu ya. Sebenarnya di dalam dunia server-side programming, ada sebuah konsep yang bernama **MVC**, yaitu kepanjangan dari Model - View - Controller:

1. **Model**, yaitu bagian kode aplikasi yang berhubungan dengan basis data.
2. **View**, yaitu bagian kode yang berhubungan dengan tampilan ke pengguna.
3. **Controller**, yaitu bagian kode yang menghubungkan antara Model dan View.

(ngomong-ngomong, ini mestinya masuk materi Back-end hahaha, cuma saya bahas sedikit ya buat pengetahuan).



Diagramnya kayak diatas. Pusing? Wajar, saya aja ngerti MVC butuh hitungan bulan (karena otak saya yang pas-pas an :p)

Intinya, MVC ini memisahkan fokus (SoC atau Separation of Concern) kepada bagiannya masing-masing, biar ga saling sikut antara satu bagian dengan bagian yang lain.

Nah, maksudnya dari MV* sendiri? Kebanyakan Framework Javascript itu mengimplementasikan sebagian dari konsep atau design pattern dari MVC ini sendiri, jadi tidak benar-benar terpaku dari MVC nya itu sendiri.

Seperti misalnya ada yang namanya **Model-View-Presenter(MVP)**, atau juga ada yang namanya **Model-View-ViewModel(MVVM)**

“Apa lagi itu pula, nyerah mas nyerah, saya angkat tangan \o_o/”

Hahaha, jangan terlalu dipikirkan serius. Pelan pelan saja untuk memahami hal ini, lama kelamaan nanti juga ngerti. Nih saya kasih artikel untuk pemahaman MVC itu sendiri

- [MVVM, MVC and MV* design pattern](http://leolanese.com/blog/?p=1377)
<http://leolanese.com/blog/?p=1377>

“Oke deh terserah mas-nya, lanjut aja deh p_p”

Siap haha! Nah, ada macem-macem framework JavaSript yang bisa kamu pakai. Ada beberapa Framework yang biasa digunakan oleh Front-end Developer



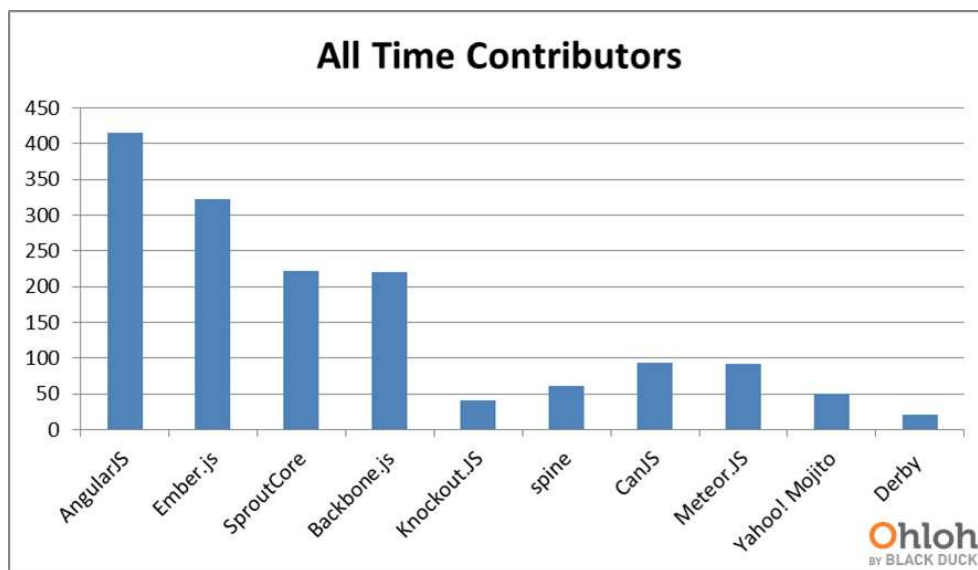
Framework-framework tersebut antara lain :

1. Ember.js
2. Knockout.js
3. Backbone.js
4. Angular.js
5. React.js

Saya kurang tahu apakah jQuery termasuk Framework atau Library, karena saat saya cari kebanyakan menyebutnya sebagai library.

“Terus saya harus ngerti semua gitu?”

Enggak perlu kok, pelajari saja yang menurut anda paling enak dipelajari atau yang banyak dipakai oleh orang lain.



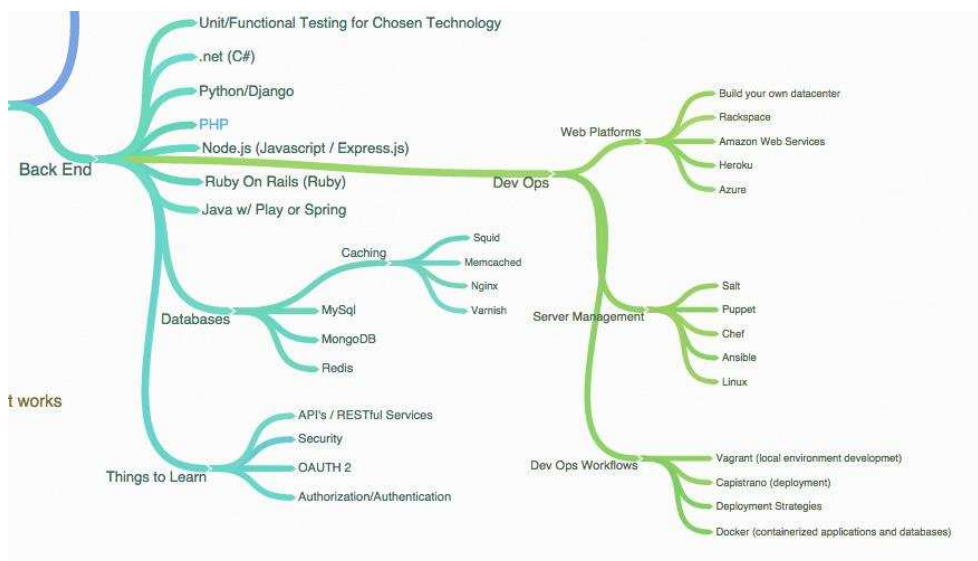
Kalau menurut diagram diatas sih, AngularJS dan EmberJS menduduki posisi paling atas kontributor terbanyak.

Back-end Web Developer

Untuk kamu yang tidak suka dengan desain (atau tidak mempunyai minat kesana) dan lebih senang berhadapan dengan pemrograman, kalian cocoknya menjadi Back-end Web Developer

Mungkin saya hanya menjelaskan secara singkat saja, karena saya pun sebenarnya tidak terlalu mendalami di bidang Back-end (hanya sebagian saja), karena masih banyak teman saya yang lebih jago seperti **Mulia Nasution** atau **Rahmat Awaludin** (yang punya buku **Seminggu Belajar Laravel**) yang bisa lebih menjelaskan di bidang ini.

Ada beberapa tahapan yang dapat kamu pelajari untuk menjadi seorang Back-end Web Developer :



Belajar Server-Side Programming Language

Hal pertama setelah kamu memahami dasar pemrograman (algoritma, struktur data, dll) yaitu dengan mempelajari bahasa pemrograman untuk web (server-side language).

Ada beberapa bahasa yang bisa kamu pelajari, seperti :

1. C# (.NET)

2. Python / Django
3. PHP
4. Node.js (Express.js)
5. Ruby (Ruby on Rails)
6. Java (Spring / Play)

Semuanya memiliki kekurangan dan kelebihan masing masing. Namun, saat ini bahasa pemrograman yang banyak digunakan oleh developer di dunia adalah PHP.

Adapun bahasa pemrograman yang kini mulai banyak digunakan seperti Ruby (Ruby on Rails) atau Node.js (Express.js)

Belajar Framework

Setelah kamu memahami dasar pemrograman salah satu bahasa, silahkan lanjutkan untuk mempelajari Framework.

Karena saya sendiri menggunakan bahasa PHP sebagai server-side language saya, saya sangat menyarankan untuk mempelajari framework **Laravel**. Sebelumnya saya sendiri menggunakan framework **CodeIgniter**.

Oh iya, ngomong2 kalau emang pengen belajar **CodeIgniter**, kebetulan temen ane **Nuris Akbar** pun punya DVD Tutorial yang bagus nih, langsung aja ke <http://belajarphp.net/kategori-52-tutorial-premium.html> kalau mau cek. Kalau gak coba tanya-tanya dulu aja orangnya di Facebook sekalian kenalan hehe.

Kalau **Laravel** sih saya rekomendasiin bukunya mas **Rahmat Awaludin**, bentuknya eBook kok! Bukunya kalau gak salah ada di Leanpub, coba dicari aja!

Belajar Database

Selain bahasa pemrograman, hal yang harus kamu pelajari (bahkan wajib) yaitu Database (basis data). Database ini yang akan kamu gunakan untuk menyimpan semua data yang ada di aplikasi yang kamu buat menggunakan bahasa pemrograman.

Banyak pilihan DBMS (Data Base Management System) yang bisa kamu gunakan, namun kebanyakan developer dan perusahaan menggunakan :

1. [MySQL \(paling banyak digunakan\)](#)
2. [MongoDB](#)
3. [Redis](#)

Belajar Database Caching

Database caching ini merupakan salah satu hal yang harus kamu pelajari. Untuk apa? Bayangkan sebuah aplikasi yang kamu buat, mempunyai pengguna yang terbilang besar (ribuan, ratusan ribu).

Tanpa optimasi dari sisi database (dan tentunya juga dari sisi programnya), aplikasi yang anda buat akan berjalan tidak sesuai dengan harapan (berat, lemot, kinerja server berat).

Teknologi yang bisa kamu pelajari antara lain :

1. [Squid](#)
2. [Memcached](#)
3. [Nginx](#)
4. [Varnish](#)

Hal lain yang harus dipelajari

Belajar Membuat API / RESTful Services

Hal ini merupakan hal yang sangat saya sarankan untuk kamu pelajari. Kenapa? Karena di era web development (khususnya sekarang), kebutuhan akan API sangat banyak.

Kegunaan untuk API tersebut untuk apa? Tentu saja untuk semuanya. Seorang Front-end Developer akan membutuhkan API itu untuk sistem yang anda buat, apabila seorang Front-end Developer tersebut menggunakan JavaScript Framework seperti Angular.js.

Mobile Developer (Android / iOS) pun pastinya akan membutuhkannya juga.

Belajar Security

Nah, security adalah hal yang sangat penting untuk dipelajari, karena security ini merupakan hal yang sangat vital didalam aplikasi yang akan kamu buat. Apabila aplikasi kamu mempunyai security yang buruk, siap siap saja aplikasi anda akan dibobol atau dirusak oleh pihak yang tidak bertanggung jawab.

Belajar OAUTH 2

OAUTH 2 ini merupakan teknologi autentikasi baru yang biasa digunakan web-web besar seperti Facebook, Twitter, atau yang lainnya.

Belajar Authorization / Authentication

Tentunya aplikasi yang kamu buat, hanya bisa diakses oleh orang yang kamu izinkan bukan?

Saya udah bisa nih, terus gimana cari duitnya?

Nah, ini pasti bagian favorit kalian kan? Pasti tujuan utama membaca ebook ini untuk menghasilkan duit kan? Hahaha sama kok dengan saya, toh siapa yang gak perlu duit? (tapi jangan lupa juga nyari pahala ya :p)

Cara-cara yang saya akan bahas disini merupakan beberapa pengalaman saya selama **banting tulang** dalam dunia web development, dan beberapa pengalaman teman yang sering saya amati di Facebook maupun di lingkungan sekitar saya.

Mungkin kamu punya ide atau saran lainnya yang ingin ditambahkan di eBook ini? Silahkan hubungi saya, siapa tau ilmunya bermanfaat untuk orang lain juga!

Cara Pertama : Berbagi

Salah satu untuk mendapatkan uang dari web development, salah satunya adalah berbagi.

“Lha, kok berbagi? Kan ceritanya saya mau cari duit nih gimana nih sih mas?”

Yap, berbagi merupakan salah satu cara untuk mendapatkan uang.

Sebenarnya, eBook ini pada awalnya ingin saya jual. Namun, saya percaya dengan eBook yang saya bagikan ini secara gratis, akan lebih mendapatkan berkah dan amal dibandingkan banyaknya uang yang akan saya dapatkan apabila saya menjual eBook ini. Tapi gak nolak kok kalau kamu mau donasi buat eBook ini :p #plak.

Berhubung saya beragama Islam (walaupun kadang solat masih kadang lupa XD), saya coba kutip salah satu ayat Al-Quran :

“Berimanlah kamu kepada Allah dan Rasul-Nya dan nafkahkanlah sebagian dari hartamu yang Allah telah menjadikan kamu menguasainya. Maka orang-orang yang beriman di antara kamu dan menafkahkan (sebagian) dari hartanya memperoleh pahala yang besar.” (QS. Al Hadiid: 7)

Mungkin untuk kamu yang beragama lain, pastinya mempunyai ayat atau hal lain yang mungkin hampir sama seperti di atas, karena semua agama mengajarkan hal yang baik ^ _ ^

Perlu diketahui, **harta** itu tak selamanya melulu soal uang lho! **Ilmu** yang kamu dapatkan juga termasuk harta kamu juga! Jadi, selagi kamu punya Ilmu yang bisa dibagikan, why not? ^ _ ^

Kenapa saya percaya bahwa banyak berbagi banyak rejeki? Saya coba ambil contoh teman saya yang sudah membuktikan hal tersebut. Ada t teman saya di Facebook yang sudah mempunyai banyak client luar. Dia pun salah satu orang yang membuat saya termotivasi untuk membuat eBook ini :D (terima kasih mas Bagus Fikri buat inspirasinya)

Saya sering melihat dia memposting pekerjaan-pekerjaan nya di Facebook, dan dia pun senang berbagi hal-hal mulai dari ilmu ataupun template gratis (istilah kerennya **freebies**). Dari situlah, dia mendapatkan client dari luar dan pekerjaan yang didapatkan gak main-main dari situ (keren-keren!).

Percaya deh sama saya, kalau kamu banyak berbagi, mudah-mudahan rezeki yang kamu dapatkan lancar dan yang pastinya barokah (bukan varokah ya)!

Cara Kedua : Ikutan Kopdar dan Group-Group

Cara kedua untuk mendapatkan uang yaitu dengan ikutan kopdar

“Ah si mas ini bercanda deh, saya kan mau nyari uang kok malah disuruh ngumpul-ngumpul?”

Eh serius, saya gak bercanda lho :p Kopdar itu salah satu sarana untuk kalian ketemuan dan ngobrol dengan orang lain.

Misalnya nih, kalau kalian seorang freelancer, ngumpul-lah dengan teman seperjuangan kamu. Nah, terus hubungannya kopdar dengan cari duit itu apa? Nih saya kasitau ya!

Dengan kopdar, kalian bisa tau teman-teman seperjuangan kamu. Nah, siapa tau teman kamu tersebut ternyata membutuhkan *skill* kamu. Misalnya nih, si A seorang **Web Designer**, dan si B seorang **Back-end Developer**. Nah si C ini ternyata dia itu seorang **Front-end Developer** dan punya bisnis di **Themeforest** dan sedang mencari seorang Web Designer dan Back-end Developer.

Nah, kalau kopdar begitu kan nanti bisa saling ketemuan antara yang butuh dan membutuhkan, saling untung lah intinya!

Jangan sampai kalian kerja atau jadi freelancer, tapi **kerjaannya di depan komputer terus**, gak bersosialisasi. Sosialisasi itu penting lho! Mau kamu seculun apapun atau sekece apapun, ikut aja kalau ada kopdar! Toh gak ada ruginya kan, banyak untungnya? ^_^

Cara Ketiga : Mencari Pekerjaan Secara Online

Nah kalau yang ini cocoknya untuk kamu yang kerja freelance atau remote-worker. Banyak cara yang bisa kamu ambil untuk mencari kerja di Online.

Cari Kerjaan di oDesk (atau situs lainnya)

Banyak teman saya yang nyari kerja di oDesk, dan beberapa diantara mereka udah punya penghasilan yang **wow**. Bahkan para pekerja yang tiap hari kerja dari jam 9 sampe jam 5 pun kalah kalau soal earning.

Kerjaan di oDesk itu pembayaran nya pake Dollar alias \$, jadi kamu mesti persiapan untuk punya PayPal, Rekening Bank, dan kalau bisa punya Kartu Kredit (atau VCC/VCN) juga.

Saya sendiri belum pernah bekerja lewat oDesk (belum pernah dapet client soalnya :p), tapi kalau misalnya kalian mau tanya-tanya seputar oDesk, atau ingin gabung komunitasnya, Coba saja cari di Facebook, pasti ketemu kok!

Bagi kamu yang ingin mencari uang selain dari oDesk, kamu bisa coba Elance, Freelancer.net, FreelanceSwitch, Envato Studio atau situs-situs lainnya.

Cari Kerjaan di Group Facebook

Nah, akhir-akhir ini suka banyak nih lowongan pekerjaan (mau full-time ataupun part-time) yang sering diposting di Facebook.

Pokoknya, coba kamu cari-cari deh. Sering pantengin aja itu group biasanya memang suka ada kok! Tapi inget, yang sesuai dengan kemampuan kamu ya!

Cara Keempat : Membangun Portfolio dan Personal Branding

Nah, ini, salah satu poin yang menurut saya sangat sangat penting. Perusahaan aja punya Portfolio dan Brandingnya, masa kamu kalah?

Saya coba kasih contoh nih, ada teman saya di Facebook bernama **Rahmat Awaludin**. Dia sering memposting sesuatu atau berkontribusi di Group seperti Laravel Indonesia, PHP Indonesia dan Kami Kerja

Remote. Dia pun membuat eBook berjudul **Seminggu Belajar Laravel**. Dia mempunyai keahlian di **Laravel**, salah satu framework PHP yang sekarang ini sedang booming.

Lain hal dengan teman saya **Bagus Fikri Yuliono**, dia merupakan seorang **Freelance Web Designer**. Hal tersebut saya amati dari dia yang sering memperlihatkan hasil karyanya yang kebanyakan ia buat di Adobe Photoshop. Dia pun salah satu owner **Group Facebook** bernama **Desainer Web Indonesia**. Selain itu, dia pun mempunyai akun **Dribbble** dan **Behance**, yang memperlihatkan portfolio pekerjaan web designer-nya.

Ada pula salah satu teman saya yang agak “nyentrik”, yaitu mas **Yadi Afriadi** atau yang mungkin anda mengenalnya sebagai **Matthew Fariz**. Dia bekerja sebagai **Freelancer Web Designer** (sepengetahuan saya), namun yang sering saya amati di **Facebook**-nya malah kebanyakan **meme-meme** yang keren dan kreatif! Dan kebanyakan meme yang ia buat memang mempunyai makna yang bagus (dan kadang aneh juga haha).

Nah, kalau saya sendiri? Silahkan sebut saya apa saja, yang pasti sih saya senang bikin Video Tutorial, Ebook, senang juga ngeDesain dan bikin Web. Pokoknya saya senang yang berhubungan dengan Teknologi ~~dan makan yang banyaaaak!!!~~

Kamu juga harus bangun portfoliomu ya, biar orang lain mengakui keahlianmu. Secara otomatis, personal brand kamu akan terbentuk dan orang-orang inget dengan kamu juga! Oh ya inget satu lagi, jangan lupa **buat kartu nama buat kamu sendiri**. Nanti kalau ditanyain calon client ternyata gak ada nanti kamu juga yang repot :p

Pertanyaan yang sering ditanyakan

Kalau jadi web developer, harus punya Macbook?

Enggak, sama sekali enggak perlu **Mac**, apalagi **Macbook**. Toh kamu bisa ngerjain kerjaan web di sistem operasi manapun, mau di **Windows** ataupun di **Linux**. Everything is possible. Saya aja dari dulu belajar web pake Windows kok!

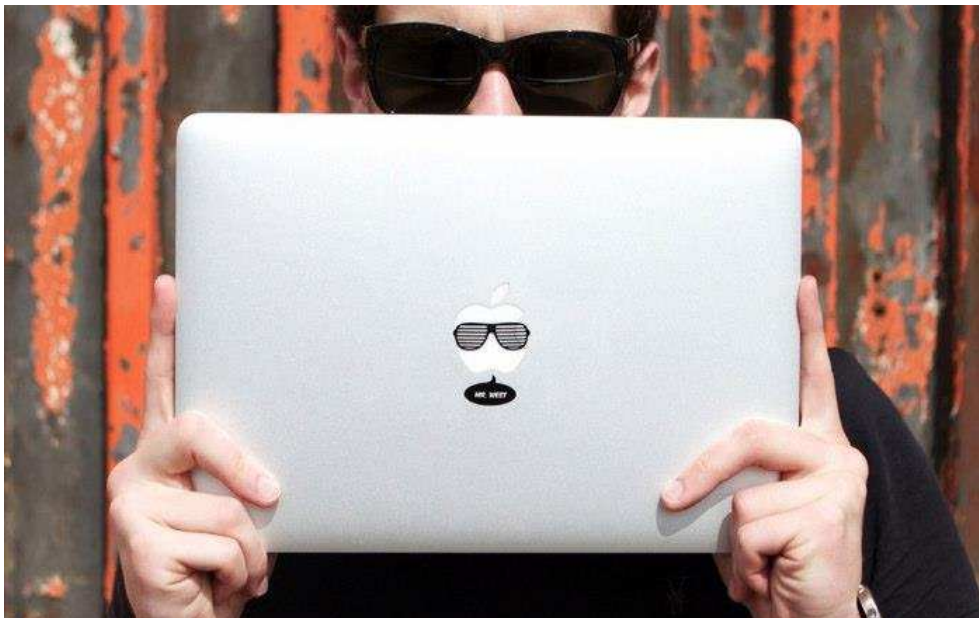
Tapi, ada sebagian client menganggap (dan kadang menentukan) harga dari device yang kamu gunakan. Hah maksudnya gimana?



Cini gini saya cerita sedikit.. Saya pernah diceritakan oleh teman saya di facebook (yang saya lupa lagi siapa, kalau ga salah dia Internet Marketer). Dulunya dia adalah seorang graphic designer yang ngerjain kerjaan desain ke client-client.



Nah, sebelum dia pakai Mac untuk meeeting dengan clientnya (meeting ketemuan), dan doi pake laptop standar (harga 3-4 jutaan lah) sang clientnya tersebut hanya memberikan fee yang bisa dibilang **average** atau standar lah.



Setelah dia menabung sampe kebeli laptop Mac (Macbook), dia coba mencari client lagi, dan dia meeting dengan membawa Macbook barunya. Ternyata hasilnya diluar dugaan.

Dia bisa menarifi sang client tersebut **dengan harga lebih tinggi** dibanding sebelumnya. Kok bisa gitu?

Ternyata sang client berpikir bahwa **“Wah ini orang sepertinya professional, kayaknya kerjaan nya bagus nih. Gak bisa kalau saya tarif murah ini orang”**. Artinya sang client ini **menghormati** sang calon designer ini, hanya dengan melihat device yang ia miliki.

Sama seperti seorang businessman yang mencoba untuk **melobby** seseorang calon partnernya, dan dia membawa dan menggunakan mobil mewah sekelas **BMW, Limousine** atau sejenisnya (saya ga hapal merek mobil xD). Mereka menganggap orang tersebut adalah **orang penting**. (Saya juga pernah diceritakan oleh mentor saya soal ini)

Dibilang miris? Tidak juga. Toh kalau misalnya kalian ingin PDKT sama cewek/cowok idaman kalian, pasti kalian dandan super wangi dan super rapih. Gak beda jauh kan? :p

Istilahnya buatlah kalian terlihat “sepenting” mungkin, agar sang calon client kamu percaya bahwa kamu benar-benar serius dan memang orang penting.

Namun, hal tersebut kembali ke diri kalian masing masing. Kalau misalnya kalian udah keren banget tapi giliran ngelobby ke client gak bisa kan percuma :p Apalagi kalau kalian gak punya skill yang mumpuni untuk mengerjakan kerjaan tersebut. Istilahnya **“Man Behind the Gun”**

Temen saya pun banyak tuh yang **keren-keren** namun dengan device yang **sederhana**, dan bisa menghasilkan dengan earning yang terbilang wow.

Ada gak komunitas yang mewadahi web developer?

Banyak, banyak banget! Ada banyak group yang bisa kamu join, untuk menjalin silaturahmi ataupun bertanya-tanya. Salah satu group yang saya ikuti antara lain:

1. PHP Indonesia
2. Kami Kerja Remote
3. Desainer Web Indonesia
4. Web Design Indonesia
5. Laravel PHP Indonesia
6. Codeigniter Indonesia

Kalau mau bisa Web Development harus kuliah ga?

Tidak harus, tapi apabila kamu mempunyai waktu dan uang yang cukup untuk kuliah, apa salahnya kuliah?

Toh saya aja masih ambil kuliah kok di Widyatama ngambil Sistem Informasi, walaupun saya lebih suka ke bagian **Front-end Web Developer** dan **UX/UI Designer**.

Tools yang digunakan biasanya apa aja sih?

Ya jelas, kalau kamu seorang Web Developer, musti punya :

1. Laptop / Komputer
2. Internet
3. Smartphone
4. Kopi :p

Saya mau nanya yang lain mas..

Sejujurnya, saya bingung apa saja yang harus saya tuliskan disini hahaha xD, tapi kalau kamu ada pertanyaan, silahkan bisa email saya di hi@galihpratama.net atau coba tanya-tanya di group **Kami Kerja Remote**, saya biasa mantengin disitu soalnya.

Makasih ya!

Hahaha, saya bingung sebenarnya mau nulis kata penutup, tapi sekali lagi saya berterima kasih sekali buat kamu yang udah baca **eBook saya yang aneh ini** sampe habis xD

Semoga bisa bermanfaat buat kamu yang ingin terjun di dunia persilatan **web development** ini. Saya doakan agar kamu bisa cepat dapet ilmu dan mudah-mudahan dapet rezeki juga :D



“Thanks for Reading this Book!”