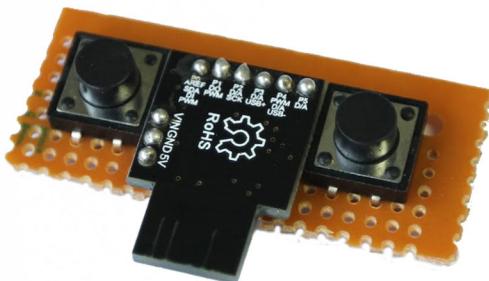


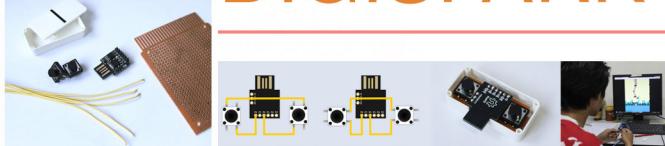
C<>dertalk

ONCOM ORIENTED

BIKIN TEMPLATE PORTFOLIO DENGAN BOOTSTRAP



BIKIN JOYSTICK DENGAN DIGISPARK



Ngobrol Santai, Bareng
Wenes Kushnadi
CTO sribulancer.com

Belajar Berkarya
dari DICODING.

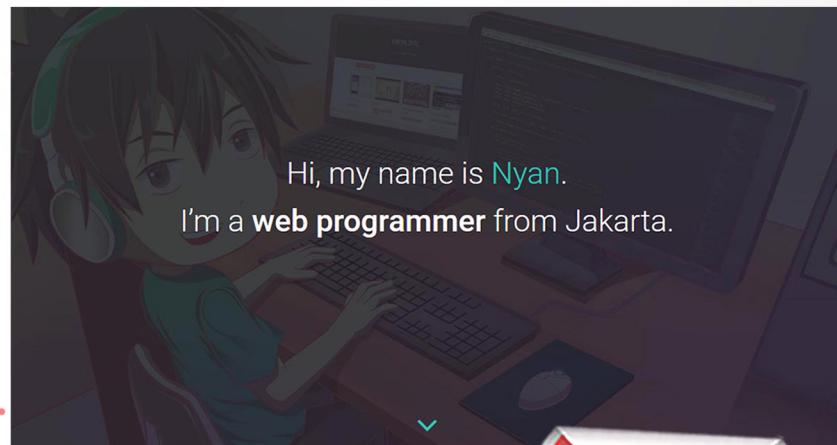
NARENDRA WICAKSONO
Nara Sumber & Co-Founder DICODING



- Penyusun: Tim Codepolitan -



"Include
Source Code"



Lorum ipsum dolor sit amet

Lorum ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercit ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.

Lorum ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Bonus
Wallpaper
Kece



My Work

Lorum ipsum dolor sit amet, consectetur adipiscing elit



Memprogram Dunia Fisik Bersama Arduino



Sapa Coders

Hallo kawan-kawan coders sebangsa dan setahan hair. Senang sekali rasanya kami bisa kembali menemani ruang baca teman-teman semuanya dengan sebuah karya. Semoga karya yang kami garap dengan cinta dan keseriusan ini bisa bermanfaat bagi teman-teman sekalian.

Buat temen-temen yang belum tau, jadi pada zaman dahulu kala, jauh sebelum CodePolitan lahir, kami pernah menerbitkan majalah elektronik (E-Magazine) yang secara umum membahas seputar pemrograman dengan penyampaian yang santai bernama nyankodMagz (nyankod.com). Majalah nyankodMagz kami terbitkan 1 bulan sekali dan kami bagikan secara gratis dengan format pdf buat teman-teman yang ingin belajar pemrograman. Hal yang menarik dari nyankodMagz adalah pembahasannya yang ringan dan mudah dicerna bahkan terkadang jenaka namun dengan konten yang dalam. Agak aneh memang untuk sebuah konten yang membahas tentang pemrograman yang notabene adalah hal teknis yang penuh dengan logika. Tapi walau demikian banyak orang yang menyukai nyankodMagz dan merasa terbantu sekali dalam belajar pemrograman. Bahkan beberapa orang menjadi terinspirasi untuk memulai belajar pemrograman, karena ternyata belajar pemrograman tidak sehoror yang dibayangkan dan bisa menyenangkan jika dibahas dengan lebih santai dan tidak kaku.

Dari edisi ke edisi kami terus senantiasa belajar untuk menyajikan konten yang terbaik serta sedikit demi sedikit memperkaya kontennya. Kami selalu berusaha memberikan konten yang terbaik yang kami bisa.

Setelah kurang lebih dua tahun menggarapnya, dalam perjalannya mulai muncul berbagai hambatan, mulai dari hambatan finansial, riset, sampai kepada resource pengisi konten. Keinginan hati untuk selalu berkarya dan memberikan yang terbaik, namun apalah daya beberapa hambatan tadi membuat kami tak berdaya. Yang pada akhirnya kami memutuskan untuk menghentikan penerbitan nyankodMagz di edisi 17. Dan kami fokus 100% pengembangan CodePolitan.

Dalam perjalanan kami mengembangkan CodePolitan, banyak sekali teman-teman coders yang menanyakan tentang nyankodMagz dan berharap kami menerbitkannya kembali. Sungguh merupakan harapan kami untuk terus berkarya dan kembali menerbitkan nyankodMagz, namun kami sadar bahwa jika kami hanya menerbitkannya saja sebagaimana sebelumnya, itu hanya akan mengulang cerita lama. Dari situ kami paham bahwa untuk jika ingin melanjutkan penerbitan majalah dan konsisten dalam berkarya kami harus mempersiapkan segala sesuatunya dengan matang, termasuk biaya operasional produksi. Oleh karena itu, mulai edisi ini, kami terpaksa membuatnya menjadi berbayar.

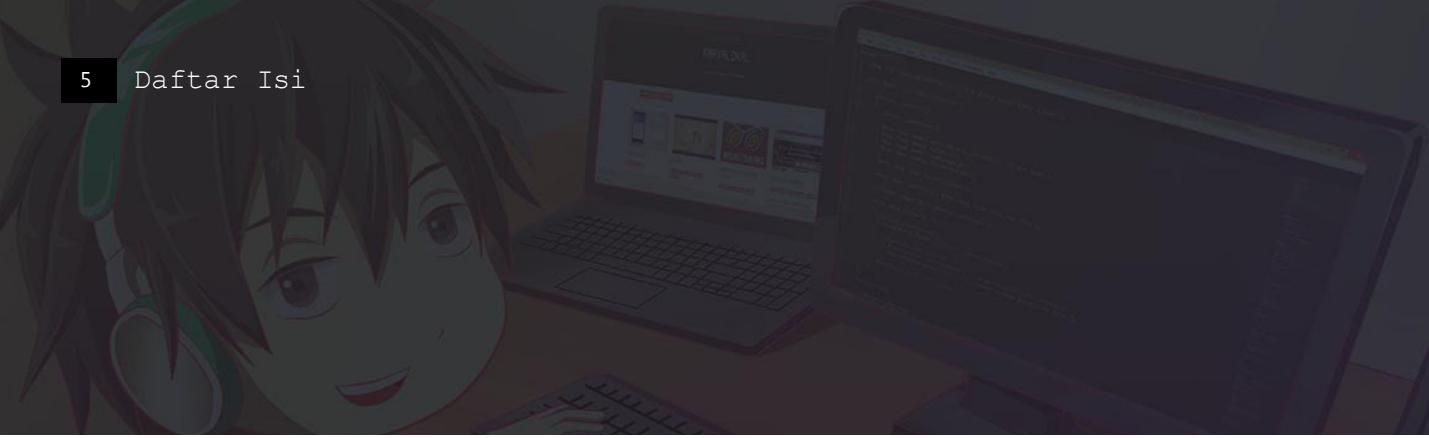
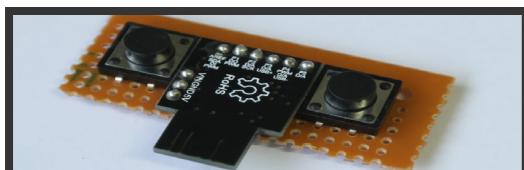
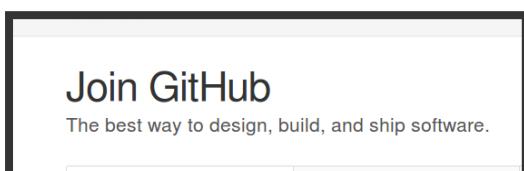
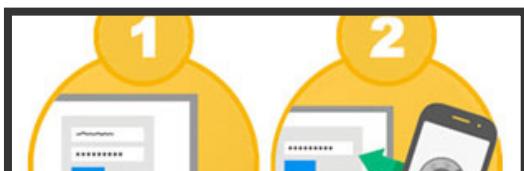
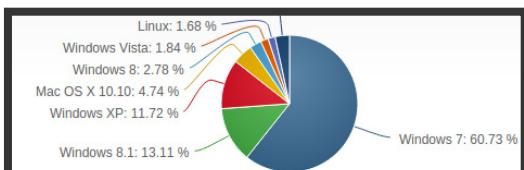
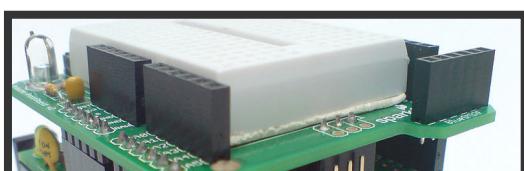
Sejatinya CoderTalk adalah lanjutan dari nyankodMagz. Sebuah estafet dengan visi sama namun tentunya dengan berbagai perbaikan, pembaharuan dan pengembangan. Itulah kenapa penamaan dan edisinya pun melanjutkan nyankodMagz. Perubahan nama ini kami lakukan untuk memperbaiki branding majalah ini dan juga terkait platform yang sedang kami kembangkan. Karena kedepannya CoderTalk bukanlah hanya sekedar majalah.

Ada banyak sekali pembahasan menarik di edisi kali ini, seperti kita akan belajar bagaimana cara membuat template web untuk profile pribadi kita. Juga ada pembahasan tentang bagaimana membuat joystick sederhana

menggunakan Digispark. Nggak Cuma itu, kita juga akan belajar langsung dari CTO Sribulancer dan CEO Dicoding, dan masih banyak pembahasan menarik lainnya.

Akhirnya, kami mengucapkan terima kasih atas segala dukungannya yang membuat kami bisa berkarya lagi. Semoga majalah ini dapat bermanfaat. Selamat menikmati jamuan Oncom Oriented ini :D

[CodePolitan Team]

**Main Artikel****Bootstrap**Membuat Halaman Portfoliao
Dengan Bootstrap 3**7****Weekend Project****Digispark**Membuat joystick
Menggunakan Digispark**51****Learn From Expert****Interview**Ngobrol Santai Bareng Wenes
Kusnadi, CTO sribulancer.com**60****Inspirasi****Dicoding**Belajar Berkarya Dari
Dicoding**66****Tool Reviews****Git**Efisiensikan Kerjaan Codingmu
dengan Memanfaatkan Tool
Bernama Git**72****Security****Web Programer**Hal-Hal Yang Harus Dihindari
web Programmer**86****Tool Reviews****Ubuntu Make**Ubuntu Make, Sahabat
Developer Pengguna Ubuntu**91****Info****Arduino**Memprogram Dunia Fisik Bersama
Arduino**99**

LOL

CODING

MAKAN

CODING

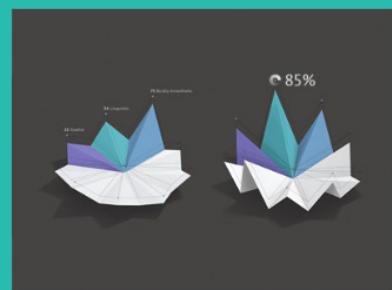
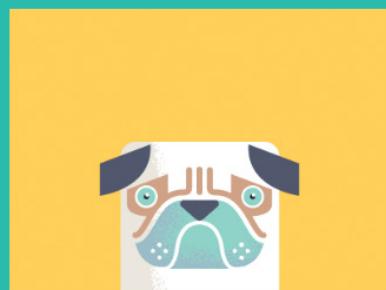
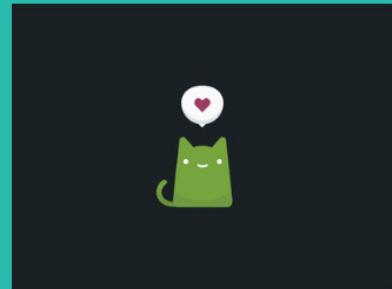
TIDUR

PROGRAMMER LIFE



My Work

Lore ipsum dolor sit amet, consectetur adipiscing elit



[MORE]

Membuat Halaman Portfolio dengan Bootstrap 3

Kresna Galuh D. Herlangga

“Kamu punya banyak karya, tapi cuma berceceran, terlupakan dalam seonggok folder, hanya kamu dan Tuhan yang tau? Mendingan publikasikan karyamu dengan halaman portfolio yang keren! Dengan Bootstrap 3, itu akan menjadi lebih mudah.”

Sebagai seorang coder, pasti kamu pernah menyelesaikan beberapa project kan, mungkin untuk pekerjaan, tugas kuliah atau hanya sekedar keisengan pribadi. Pertanyaannya adalah, dimana kamu biasa mendokumentasikannya? Github? Linkedin? Pernahkah kamu berpikir untuk membuat sebuah halaman khusus milikmu sendiri untuk menampilkan hasil karyamu? Kalau belum, mungkin inilah saatnya bagi kamu untuk kebetulan kamu sudah akan bermanfaat sebagai tertarik untuk revamp. Oriented kali ini kita sebuah halaman portfolio menggunakan Bootstrap

The Bootstrap logo consists of a dark purple rounded square containing a white stylized letter 'B'.

membuatnya. Dan jika punya, mungkin ini juga inspirasi, siapa tau kamu Karena pada Oncom akan mencoba membuat berbasis web dengan 3.

Kenapa saya harus membuat halaman portfolio online? Sederhananya dari pada karyamu berceceran, terlupakan dalam seonggok folder, hanya kamu dan Tuhan yang tau bahwa kamu pernah membuatnya, mendingan dipublikasi dengan format yang baik. Mungkin belum tentu akan bermanfaat sekarang, tapi boleh jadi di masa depan itu adalah asset yang berharga yang akan meningkatkan value yang kamu miliki. Paling tidak, ketika ada yang bertanya “Kamu kan programmer, apa yang pernah kamu buat?”, kamu bisa dengan keren menjawabnya, “silahkan buka

www.blablabla.blabla itu semua yang udah gue pernah kerjain". Kapan kira-kira pertanyaan itu akan keluar? Waktunya sih nggak nentu, bisa jadi ketika kamu melamar kerja atau mungkin ketika melamar gadis :D

Sebenarnya kita bisa saja membuat portfolio kita di Github atau Linkedin. Tapi tentunya tidak akan unik dan bebas. Jika kita membuatnya sendiri tentu tampilan dan konsepnya kita sendiri yang menentukannya.

Tapi terlepas dari itu, point inti dari apa yang kita buat dan pelajari kali ini bukanlah portfolionya. Tetapi bagaimana kita membuat interface halaman web dengan menggunakan Bootstrap 3, dan dalam hal ini kita akan membuat halaman portfolio.

So... kita di sini bukan membuat sebuah sistem portfolio. Melainkan kita akan mencoba belajar bagaimana membuat sebuah interface web dengan cara sebagaimana seharusnya.

Berkenalan dengan Bootstrap 3



Bootstrap adalah sebuah framework css yang dapat digunakan untuk mempermudah membangun tampilan web. Bootstrap pertama kali dikembangkan pada pertengahan 2010 di Twitter oleh Mark Otto dan Jacob Thornton. Bootstrap dikembangkan secara open source dengan lisensi MIT.

Bootstrap telah menyediakan banyak sekali class css dan plugin javascript yang bisa kita pakai untuk membantu mempermudah kita membuat halaman web. Karena kemudahan penggunaan, banyaknya komponen dan kelengkapan dokumentasinya, saat ini Bootstrap menjadi salah satu front-end framework yang paling banyak digunakan di dunia.

Sejatinya Bootstrap adalah sebuah framework, yaitu sebuah kerangka dasar kerja. So... selengkap dan sebanyak apapun class dan plugin yang ada di dalamnya, pasti akan ada kebutuhan yang tidak bisa terpenuhi dengannya. Sengaja saya sampaikan ini, karena banyak orang yang salah paham dalam memaknai framework, terutama Bootstrap. Banyak yang beranggapan jika kita menggunakan Bootstrap maka kita akan benar-benar bisa fokus pada core system dan melupakan stylesheet. Tidak sepenuhnya salah memang, karena itu merupakan salah satu tujuan Bootstrap dibuat. Tapi coba bayangkan jika kita menggunakan Bootstrap sebagaimana adanya (tanpa customize), maka akan banyak sekali web di dunia ini yang memiliki tampilan serupa karena menggunakan Bootstrap sebagai bahan dasarnya. So, kalaupun kita menggunakan Bootstrap, kita tetep harus menyesuaikannya dengan kebutuhan kita bahkan harus merubah theme dasarnya serta menambahkan stylesheet buatan kita agar tampilannya bisa unik sebagaimana yang kita inginkan.

Sebenarnya versi teranyar Bootstrap saat tulisan ini dibuat adalah Bootstrap v4.0.0-alpha. Hanya saja, karena dikhawatirkan versi alpha ini belum stabil dan memungkinkan akan banyak perubahan di versi stabilnya nanti, so di Oncom Oriented ini kita akan menggunakan versi lamanya yang sudah stabil, yaitu Bootstrap v3.3.5.

Cara Menggunakan Bootstrap

Agar pemahaman dan pembelajaran kita bulat, saya akan coba sedikit bahas tentang bagaimana cara menggunakan Bootstrap. Dan buat yang

sudah terbiasa dengan Bootstrap, mungkin bagian ini bisa dilewat.

Oh ya, sebelumnya saya asumsikan teman-teman pembaca sekalian sudah memahami konsep dasar dari HTML dan mengenal tag-tag HTML ya. Soalnya kita nggak akan bisa melangkah ke tahap selanjutnya tanpa pemahaman dasar akan HTML. So, buat kamu yang belum familiar dengan HTML, silahkan pelajari dulu, banyak kok tutorialnya di internet. Dan mungkin beberapa link berikut bisa membantu:

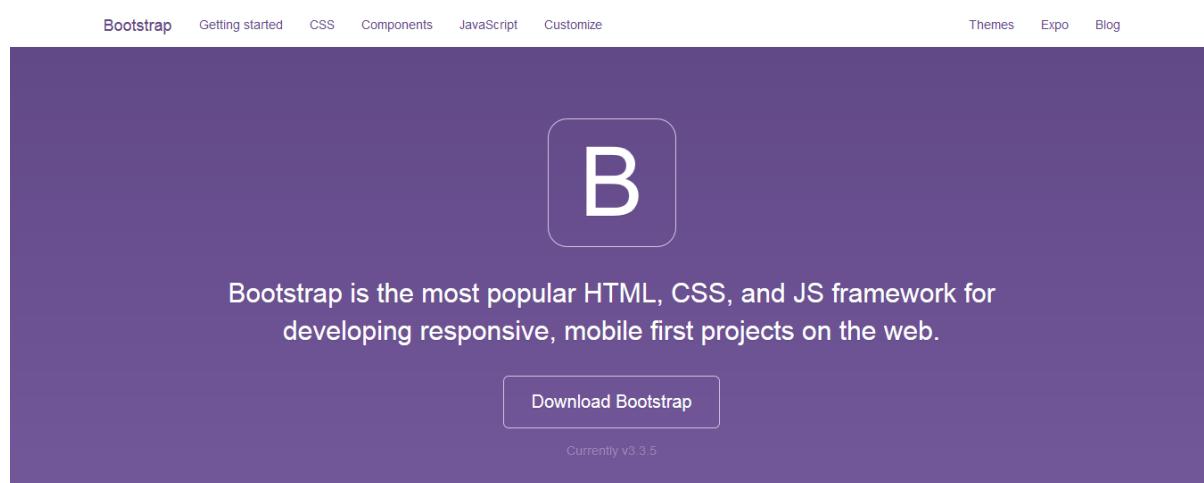
<https://id.wikipedia.org/wiki/HTML>

<http://www.w3schools.com/html/>

<http://www.tutorialspoint.com/html/>

<http://html.net/tutorials/html/>

Untuk menggunakan Bootstrap, pertama-tama kita harus mendownloadnya terlebih dahulu melalui web resminya di <http://getbootstrap.com>, kemudian klik tombol Download Bootstrap.



Selanjutnya pilih saja Bootstrap versi production yang siap pakai, yaitu yang paling sebelah kiri dengan tombol berlabel "**Download Bootstrap**".

FYI, yang tengah adalah bootstrap untuk development, berisi file Less, JavaScript dan font yang belum dibundel dan dicompile. Kalau kita mau mempelajari bagaimana Bootstrap dibuat untuk kemudian berkontribusi

dalam pengembangan Bootstrap, atau kita hendak memodifikasi struktur Bootstrap, kita bisa mendownload versi Source code ini. Sedangkan yang paling kanan adalah versi Sass. Sementara abaikan saja versi **Source code** dan **Sass**.

Download

Bootstrap (currently v3.3.5) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

[Download Bootstrap](#)

Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

[Download source](#)

Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

[Download Sass](#)

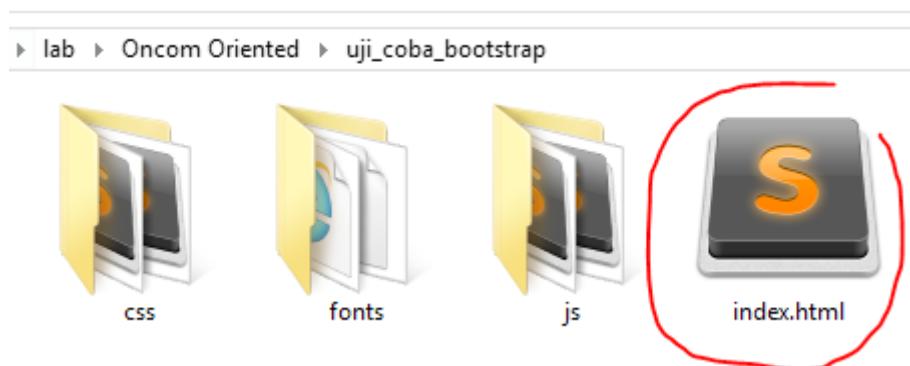
Setelah selesai mendownload kita akan mendapatkan sebuah file zip dengan nama **bootstrap-3.3.5-dist.zip**. Selanjutnya ekstraklah file tersebut! Maka akan muncul sebuah folder dengan nama **bootstrap-3.3.5-dist** yang memiliki 3 sub folder di dalamnya, yaitu css, fonts dan js. Kurang lebih kira-kira begini susunannya:

```
bootstrap-3.3.5-dist/
|--- css/
|   |--- bootstrap.css
|   |--- bootstrap.css.map
|   |--- bootstrap.min.css
|   |--- bootstrap-theme.css
|   |--- bootstrap-theme.css.map
|   |--- bootstrap-theme.min.css
|--- fonts/
|   |--- glyphicons-halflings-regular.eot
|   |--- glyphicons-halflings-regular.svg
|   |--- glyphicons-halflings-regular.ttf
|   |--- glyphicons-halflings-regular.woff
|   |--- glyphicons-halflings-regular.woff2
|--- js/
|   |--- bootstrap.js
|   |--- bootstrap.min.js
```

Setelah mendownload dan melihat struktur file Bootstrap, sekarang mari kita uji coba menggunakan Bootstrap pada halaman web.

Pertama-tama ubahlah nama folder hasil ekstrak file Bootstrap tadi terlebih dahulu yang semula bernama **bootstrap-3.3.5-dist** menjadi **uji_coba_bootstrap** agar lebih mudah. Btw nama folder sebenarnya bebas aja, yang penting mempermudah.

Kemudian di dalam folder **uji_coba_bootstrap** buatlah sebuah file baru dengan nama **index.html**, yaitu sebuah file html tempat dimana kita akan mencoba menguji coba penerapan Bootstrap dalam halaman web.



Selanjutnya bukalah file **index.html** tersebut melalui texteditor semisal Sublime Text atau Notepad++, kemudian tuliskan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <title>Uji Coba Bootstrap</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
```

```
<body>
  <div class="jumbotron">
    <div class="container">
      <h1>Hello World!</h1>

      <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

      <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
    </div>
  </div>

  <!-- jQuery -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

  <!-- Bootstrap Js -->
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Kode di atas adalah kode HTML biasa untuk menampilkan halaman Hello World sederhana dengan style bergaya Bootstrap.

Teks berwarna kuning pertama adalah untuk mengatur agar tampilan web kita bisa responsive.

Selanjutnya teks berwarna hijau pertama adalah kode untuk memanggil file Bootstrap. Kita hanya perlu mengarahkannya pada file **bootstrap.min.css**. Path-nya tidak mesti seperti itu, file Bootstrapnya pada dasarnya bisa disimpan dimana saja. Karena file **index.html** kita sejajar dengan folder css, maka path Bootstrap kita adalah **css/bootstrap.min.css**.

Selanjutnya teks berwarna biru muda adalah kode utama dari halaman web kita. Itu adalah apa yang hendak kita tampilkan pada halaman web. Dalam hal ini kita akan menampilkan Jumbotron, yaitu sebuah kotak

besar yang berisi teks dan tombol di dalamnya.

Teks berwarna kuning yang kedua adalah untuk memanggil file jQuery. Karena Bootstrap memerlukan jQuery untuk JavaScriptnya. Agar sederhana kita ambil saja jQuery dari library-nya Google.

Dan teks berwarna hijau yang kedua adalah untuk memanggil plugin JavaScript-nya Bootstrap. Sama seperti CSS-nya, path-nya pun mengikuti posisi Bootstrap disimpan.

So, kurang lebih seperti berikut tampilan dari kode yang kita tulis tadi jika dijalankan melalui browser:



Itulah yang saya maksud dengan framework, kita bahkan belum menuliskan satu baris pun kode css, tapi tampilan Hello World kita sudah sangat rapih, bahkan sudah ada tombol yang ciamik. Jadi sederhananya Bootstrap telah menyediakan style standar yang bisa kita langsung pakai ketika kita membuat halaman web. Sebagai contoh pada gambar di atas kita menampilkan gambar tombol berwarna biru, itu adalah penggunaan class **btn btn-primary** pada tag **a**.

Kita baru saja belajar bagaimana menggunakan Bootstrap dan menggunakan jumbotron serta salah satu style button (**btn-primary**). Kurang lebih seperti itu cara penggunaannya. Kita tidak akan membahas penggunaan setiap komponen dan atau class yang ada di Bootstrap satu persatu dalam Oncom Oriented ini. So, buat kamu yang tertarik buat mengenal Bootstrap

lebih dalam kamu bisa langsung belajar dari dokumentasinya langsung di <http://getbootstrap.com>. Mudah-mudahan di lain kesempatan kita bisa membahas ini lebih dalam :D

Mulai Mendesign Halaman Portfolio

Setelah kita siap dengan toolsnya yaitu Bootstrap, selanjutnya kita akan mulai masuk ke tahap design halaman portfolionya.

Mendesign sejatinya adalah berbicara tentang seni, dan setiap orang pasti punya standar dan selera masing-masing. Jadi jika kamu punya selera yang berbeda dengan saya, itu sah-sah saja. Paling tidak nanti kita akan belajar bagaimana membuatnya, sehingga jika kamu mau membuat versi kamu sendiri nantinya akan lebih mudah ketika sudah memahami konsep dasarnya.

Apa yang kita bayangkan, pada dasarnya adalah interpretasi dari apa yang telah kita lihat, dengar atau rasakan. Kita tidak mungkin membayangkan sesuatu yang tidak pernah kita indrai. Semakin banyak yang kita indrai, maka semakin banyak pula yang mampu kita bayangkan. Dalam hal design, semakin banyak kita melihat-lihat design-design yang bagus, maka semakin banyak pula kita terinspirasi untuk membayangkan design-design yang akan kita buat. So, jika kita hendak mendesign sesuatu baik itu portfolio atau yang lainnya, maka perbanyaklah wawasan dan inspirasi dengan banyak melihat-lihat karya-karya terkait hal tersebut yang pernah orang lain buat. Saya jamin, itu pasti akan berdampak baik.

Beberapa web yang saya rekomendasikan untuk mencari inspirasi design adalah sebagai berikut:

<http://www.dribbble.com/>

<http://www.behance.net/>

<http://www.awwwards.com/>

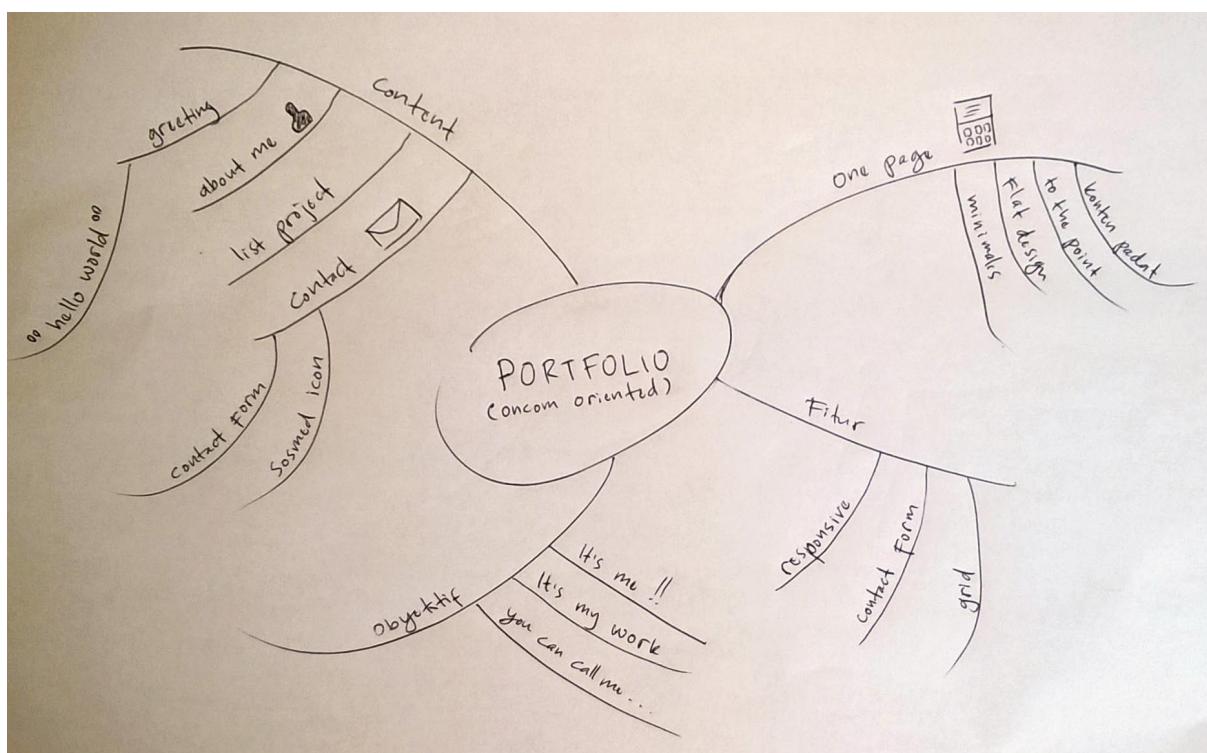
<http://flatdsgn.com/>

<http://www.makebetterwebsites.com>
<http://www.siteinspire.com>
<http://www.thebestdesigns.com>
<http://www.unmatchedstyle.com>
<http://www.kreavi.com>
<http://themeforest.net>

Analisa Kebutuhan Halaman

Oke, kita sekarang sudah punya banyak referensi design untuk portfolio. Karena sekarang saya sedang suka dengan flat design, maka pada kesempatan kali ini saya akan mencoba membuat halaman portfolio bernuansa flat design.

Setelah kita sudah tau design seperti apa yang akan kita buat. Selanjutnya adalah kita analisa terlebih dahulu kira-kira apa saja yang akan kita tampilkan dan kita butuhkan untuk halaman web portfolio kita.



Biasanya saya melakukan analisa sederhana dengan menggunakan mind mapping. Kamu boleh saja melakukannya dengan cara yang lainnya.

Oke sekarang semuanya menjadi lebih jelas. Apa yang saya butuhkan adalah sebuah halaman portfolio dengan konsep one page. Dengan tampilan minimalis ala flat design. Tentu saja responsive agar tetap rapih diakses melalui berbagai device. Adapun konten yang akan saya muat di dalam halaman tersebut adalah, greeting, about me, list project dan contact. Sedangkan tujuan dari halaman portfolio tersebut adalah untuk menunjukkan siapa saya, apa yang telah saya kerjakan dan cara menghubungi saya.

Membuat Wireframe

Setelah kita tau apa yang ingin kita buat dan punya gambarannya. Saatnya kita mencoba memvisualkannya ke dalam sebuah wireframe. Kenapa harus melalui tahap wireframe terlebih dahulu? Jawabannya yaitu agar kita nggak cape ngerubah kode html kita tanpa arah dan tujuan. Setidaknya dengan wireframe ini kita punya gambaran besar akan seperti apa jadinya nanti tampilan web yang ingin kita buat.

Dari analisa yang tadi kita telah buat, berikut ini wireframe halaman portfolio yang akan kita buat:

Hi, my name is Nyan.
I'm a web programmer from Jakarta.

Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

My Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit

[MORE]

GET CONNECTED

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna.

contact@codertalk.id
 +62 8999 129 220
 CodePolitan
 CodePolitan
 @CodePolitan
 +CodePolitan

Copyright © 2015 - All rights reserved.

Let's Code!

Atas berkat rahmat Allah Yang Maha Kuasa dan dengan didorongkan oleh keinginan luhur, akhirnya telah sampailah kita pada saat yang berbahagia, dengan selamat sentosa mengantarkan rakyat Indonesia ke depan pintu gerbang kemerdekaan negara Indonesia :D

Saatnya menulis kode guys!

Persiapan Lab

Sebelum memulai menulis kode, mari terlebih dahulu kita persiapkan lab kerja kita agar dalam prosesnya nanti menjadi lebih rapih dan lebih mudah untuk dipahami. Buatlah folder baru (bebas dimana saja) dengan nama **myportfolio**. Sebenarnya nama foldernya bebas saja, tapi jika kamu ingin lebih mudah mengikutinya kamu bisa membuat namanya sama dengan apa yang saya buat. Folder **myportfolio** ini akan kita jadikan folder lab kita untuk membuat halaman web portfolio kita.

Di dalam folder **myportfolio**, buatlah sebuah folder baru dengan nama **assets**. Folder **assets** ini akan kita jadikan sebagai tempat untuk menyimpan semua asset yang kita butuhkan dalam project kali ini.

Selanjutnya di dalam folder **assets**, buatlah 3 folder baru dengan nama **css**, **img** dan **vendor**. Folder **css** adalah untuk menyimpan file stylesheet buatan kita agar tampilan web kita seperti yang kita inginkan. Folder **img** untuk menyimpan semua file gambar yang kita butuhkan dalam project ini. Sedangkan folder **vendor** adalah untuk menyimpan semua kode plugin/framework/extension atau apapun itu yang bukan buatan kita. Salah satunya nanti kita akan gunakan folder vendor ini untuk menyimpan file Bootstrap. Yups, tepat karena Bootstrap bukan buatan kita. Tujuannya agar lab kita menjadi lebih rapih. Gambaran secara umum tentang folder lab kita adalah seperti berikut:

```
myportfolio/
|--- assets/
|   |--- css/
|   |--- img/
|   |--- vendor/
```

Persiapan Asset

Ada dua library yang kita pakai dalam project kali ini, yaitu Bootstrap dan Font Awesome. Mari kita persiapkan satu persatu dalam lab kita.

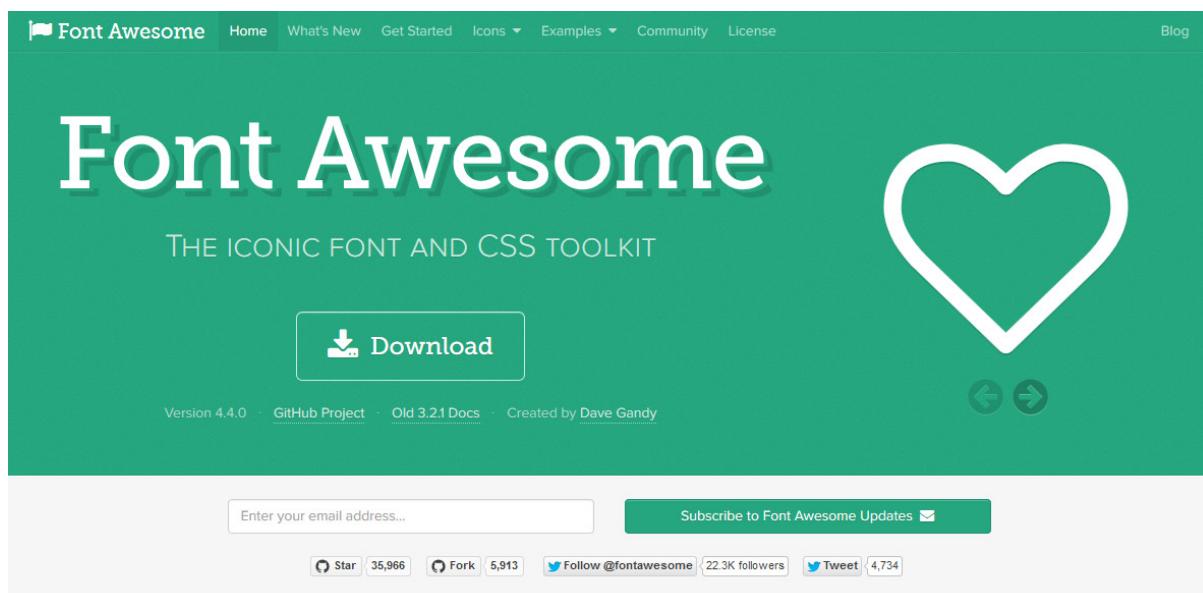
Bootstrap telah sedikit kita bahas sebelumnya tentang cara penggunaannya. Sekarang saatnya menggunakannya untuk kebutuhan yang lebih dari sekedar uji coba.

Pertama-tama, ekstrak kembali file bootstrap-3.3.5-dist.zip yang telah kita download sebelumnya, kemudian ubah nama folder bootstrap-3.3.5-dist hasil ekstrakannya menjadi “bootstrap” saja. Kemudian pindahkan folder bootstrap beserta isinya ke dalam folder vendor di dalam lab kita. Sehingga sekarang struktur foldernya seperti berikut:

```
myportfolio/
|--- assets/
|   |--- css/
|   |--- img/
|   |--- vendor/
|   |   |--- bootstrap/
|   |   |   |--- [file bootstrap: css, fonts, js]
```

Setelah kita selesai dengan Bootstrap, sekarang kita beralih ke Font Awesome. Sederhananya Font Awesome adalah sebuah library icon vektor dengan format font untuk memudahkan dalam menggunakan icon-icon standar dalam web. Sama seperti Bootstrap, Font Asewome juga telah menyediakan berbagai class yang bisa kita gunakan untuk memanggil/memakai icon tertentu. Versi teranyar Font Awesome saat tulisan ini

dibuat adalah v4.4.0. Pada versi ini sudah ada 585 icon yang siap pakai yang bisa kita gunakan untuk halaman web kita. Info lengkap dan cara penggunaannya bisa langsung dilihat melalui web resminya di <http://fontawesome.github.io/Font-Awesome/>. Nanti juga akan kita bahas sedikit penggunaannya dalam portfolio kita.



Untuk menerapkan Font Awesome, pertama-tama kita harus mendownload librarynya terlebih dahulu di <http://fontawesome.github.io/Font-Awesome/>.

Setelah selesai mendownload kita akan mendapatkan sebuah file zip dengan nama font-awesome-4.4.0.zip. Ekstraklah file zip tersebut. Maka akan muncul sebuah folder bernama font-awesome-4.4.0. Ubahlah namanya menjadi “font-awesome” saja agar lebih sederhana. Kemudian pindahkan folder font-awesome beserta isinya ke dalam folder vendor di lab kita.

Jika sudah, maka saat ini di dalam folder vendor kita akan ada dua folder yaitu bootstrap dan font-awesome. Struktur foldernya akan tampak seperti berikut:

```
myportfolio/
|--- assets/
|   |--- css/
|   |--- img/
```

```

|   |--- vendor/
|   |   |--- bootstrap/
|   |   |   |--- [file bootstrap: css, fonts, js]
|   |   |--- font-awesome/
|   |   |   |--- [file font asewome: css, fonts, less,
scss]

```

Setelah kita siap dengan dua library kita, persiapan terakhir adalah assets gambar. Silahkan download file gambar untuk project ini melalui tautan berikut: http://bit.ly/assets_gambar.

Ekstrak file_gambar.zip tersebut, kemudian copy semua file gambarnya ke dalam folder myportfolio/assets/img pada lab kita.

Oke, sampai sini kita sudah siap dengan persiapan project kita. Saat ini struktur folder kita akan seperti berikut:

```

myportfolio/
|--- assets/
|   |--- css/
|   |--- img/
|   |   |--- [file gambar]
|   |--- vendor/
|   |   |--- bootstrap/
|   |   |   |--- [file bootstrap: css, fonts, js]
|   |   |--- font-awesome/
|   |   |   |--- [file font asewome: css, fonts, less,
scss]

```

Mulai dengan HTML

Oke, setelah semua persiapan asset siap. Saatnya kita membuat halaman web portfolio kita dengan html. Buatlah sebuah file baru dengan nama index.html di dalam folder myportfolio. Kemudian tulislah kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0" />

  <title>Oncom Oriented</title>

  <!-- Bootstrap -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.
css" rel="stylesheet" />

  <!-- Font Awesome -->
  <link href="assets/vendor/font-awesome/css/font-awesome
min.css" rel="stylesheet" />

  <!-- STYLESHEET -->
  <link href="assets/css/mainstyle.css" rel="stylesheet"
/>

</head>
<body>

  <!-- MAIN WRAPPER -->
  <div class="main-wrapper">

    Hallo...

  </div>
  <!-- END: MAIN WRAPPER -->

  <!-- jQuery -->
  <script src="https://ajax.googleapis.com/ajax/libs/
jquery/1.11.3/jquery.min.js"></script>

  <!-- Bootstrap Js -->
```

```
<script src="assets/vendor/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

Kode di atas adalah kode html standar dan mirip dengan kode uji coba Bootstrap kita sebelumnya. Berbeda dengan file uji coba sebelumnya, kali ini pada bagian embed css Bootstrap pathnya adalah assets/vendor/bootstrap/css/bootstrap.min.css, hal itu dikarenakan sekarang file Bootstrapnya berada di folder assets/vendor, berbeda dari yang sebelumnya.

Selain itu, pada kode di atas juga kita telah memasang library Font Awesome. Bisa dilihat pada teks yang berwarna kuning. Instalasi Font Awesome persis sama dengan Bootstrap dan path-nya pun mengikuti tempat dimana ia disimpan, dalam hal ini di dalam folder assets/vendor.

FYI, teks berwarna hijau dengan tulisan “Hallo...”, di atas sebagai teks sementara saja. Di tahap selanjutnya kita akan timpa dengan kode yang baru. Ini hanya untuk memulai saja agar tidak kosong sama sekali tampilannya pada tahap ini.

Selanjutnya seperti yang saya sampaikan sebelumnya, bahwa Bootstrap adalah framework yang akan membantu kita membuat tampilan web kita, tapi kita tetap memerlukan customize dan tambahan kode css kita sendiri agar tampilannya bisa benar-benar seperti yang kita inginkan. So, buatlah sebuah file bernama mainstyle.css dan letakan di dalam folder assets/css. Kemudian pada file mainstyle.css tersebut tulislah kode berikut:

```
@import url(http://fonts.googleapis.com/
css?family=Roboto:400,100,300,500,700);

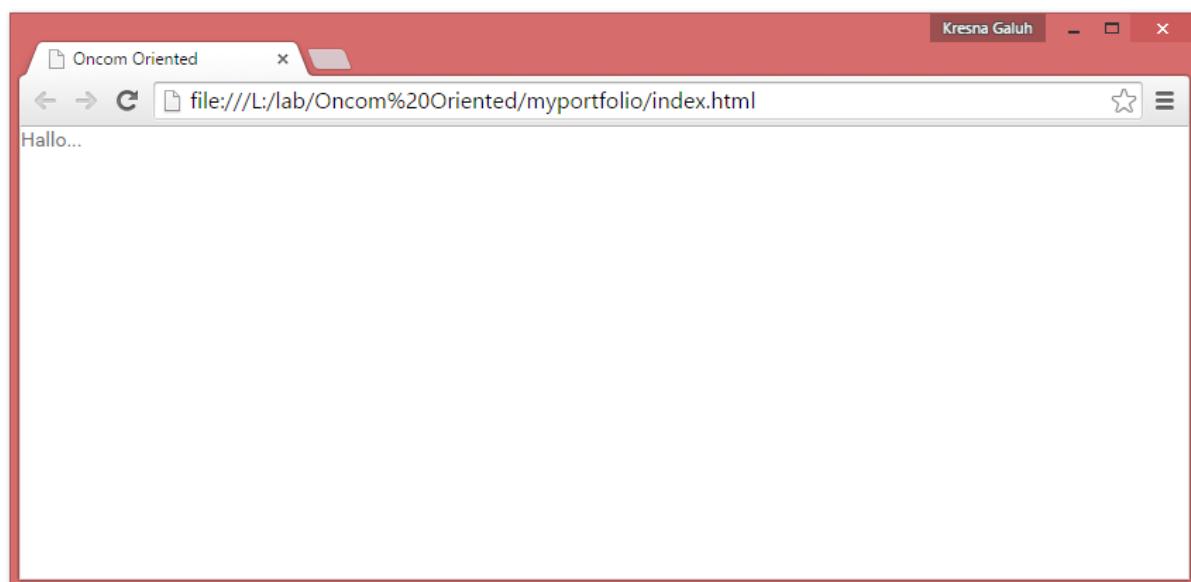
/*
** GENERAL
** -----
body {
    color: #656565;
    font-family: 'Roboto', sans-serif;
```

```
    font-size: 14px;
}

.container {
    max-width: 1060px;
}

/*
** TEXT
*/
.text-color-primary {
    color: #2abbae;
}
```

Untuk melihat progress-nya silahkan buka file index.html tersebut melalui browser.



Header

Selanjutnya kita akan membuat bagian header. Header di sini akan berisi konten greeting. Karena header inilah yang akan dilihat oleh pengunjung web portfolio kita pertama kali. Konsep header kita pada project kali ini

sederhana saja, yaitu memasang background gambar dan di dalamnya ada kalimat sapaan.

Kembali ke file **index.html** yang telah kita buat sebelumnya. Seperti pada bagian sebelumnya, saya telah katakan bahwa kita akan mengganti dan menimpa teks berwarna hijau, maka inilah saatnya. Kita akan menimpa teks hijau tersebut dengan header. Karena kedepannya kode kita akan panjang, oleh karena itu mulai saat ini saya tidak akan menulis semua kodennya setiap tahap, tapi hanya fokus menuliskan kode-kode yang fokus pada bagian tersebut saja. Namun saya akan tetap pandu dimana kode tersebut seharusnya ditulis.

Oke, hapuslah tulisan “**Hallo...**” pada file **index.html** kemudian tulislah kode berikut tepat di tempat tulisan “**Hallo...**” tadi:

```
<!-- HEADER -->
<header class="header-cover">
  <div class="container">
    <div class="header-inner">
      <h1>Hi, my name is <span class="text-color-primary">Nyan</span>.</h1>
      <h1>I'm a <strong>web programmer</strong> from
Jakarta.</h1>
    </div>
  </div>

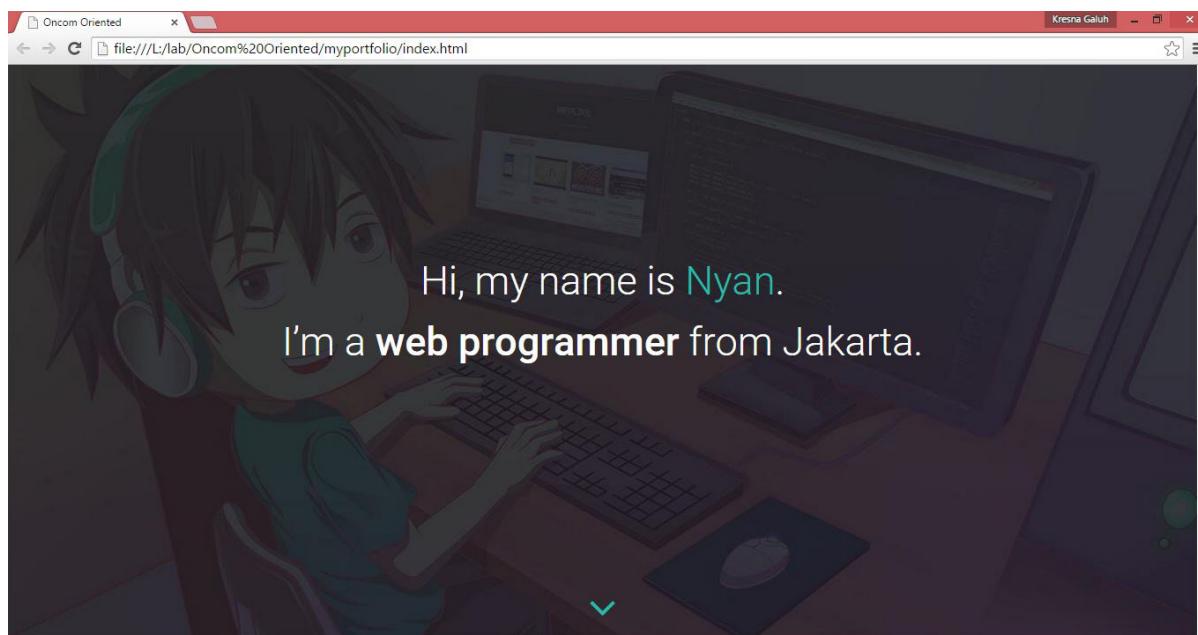
  <div class="button-area">
    <a href="#"><i class="fa fa-angle-down"></i></a>
  </div>
</header>
<!-- END: HEADER -->
```

Selanjutnya kita kembali file **mainstyle.css**. Setelah kode sebelumnya pada **mainstyle.css**, sekarang tambahkan kode berikut untuk menghias header kita.

```
/* -----
** HEADER
** ----- */
.header-cover {
    height: 650px;
    background: url(..../img/cover_header.jpg) no-repeat
bottom center;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;
    position: relative;
}
.header-cover .header-inner {
    padding-top: 200px;
    text-align: center;
    color: #ffffff;
}
.header-cover .header-inner h1 {
    font-weight: 300;
    font-size: 45px;
}
.header-cover .header-inner h1 strong {
    font-weight: 500;
}
.header-cover .button-area {
    position: absolute;
    bottom: 0px;
    left: 0px;
    right: 0px;
    width: 100%;
    text-align: center;
    font-size: 50px;
}
.header-cover .button-area a {
    color: #2abbae;
    text-decoration: none;
}
.header-cover .button-area a:hover,
```

```
.header-cover .button-area a:focus {  
    color: #209f94;  
    text-decoration: none;  
}
```

Jika sudah, pastikan hasil progressnya seperti gambar berikut:



Oh ya, background gambar itu bisa kamu ganti lho dengan gambar yang menurut kamu lebih cocok untuk profile kamu. Mungkin foto selfie kamu? Hehehehe :D

Section About

Bagaimana? Sejauh ini apakah aman? Jika hasilnya tidak seperti pada gambar saya harap kamu mau sedikit memeriksa ulang kodenya, siapa tau salah penempatannya. Tapi tenang, di akhir tulisan ini, saya akan sampaikan kode lengkapnya agar kamu bisa membandingkan dengan kode yang sedang kamu tulis.

Setelah selesai dengan section greeting pada header, sekarang kita akan membuat bagian selanjutnya yaitu section about. Pada bagian ini kita akan

membuat section untuk menjelaskan siapa kita (yang punya portfolio). Saya buat konsepnya sederhana saja, yaitu berbentuk paragraf saja. Namun tidak full dari kiri ke kanan halaman. Saya akan buat kira-kira 7/12 lebar halaman saja untuk teksnya. Di sini kita akan mencoba menerapkan sedikit konsep grid pada Bootstrap. Buat kamu yang ingin tahu lebih banyak tentang grid pada Bootstrap, kamu bisa pelajari di <http://getbootstrap.com/css/#grid>.

Dan untuk mempercantik halaman di bagian kanan bawah section about ini akan saya beri hiasan gambar kota berbentuk shadow berwarna hijau. Oke, kembali ke index.html. Setelah kode:

```
</header>
<!-- END: HEADER -->
```

Dan sebelum kode:

```
</div>
<!-- END: MAIN WRAPPER -->
```

Tulislah kode berikut (pastikan di antara kode tadi):

```
<!-- ABOUT -->
<section class="about">
  <div class="container">
    <div class="row">
      <div class="col-md-7">
        <h2><strong class="text-color-primary">Lorem ipsum</strong> dolor sit amet</h2>

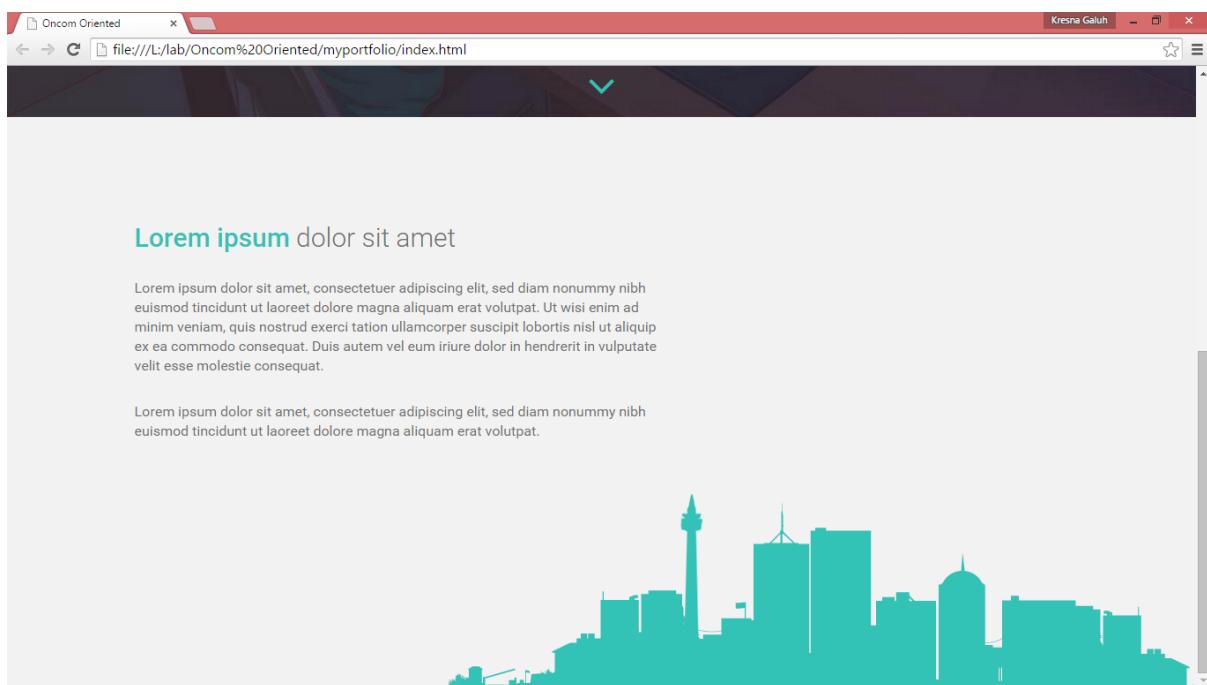
        <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.</p>
```

```
        <p>Lorem ipsum dolor sit amet, consectetuer  
adipiscing elit, sed diam nonummy nibh euismod tincidunt  
ut laoreet dolore magna aliquam erat volutpat. </p>  
    </div>  
    </div>  
    </div>  
</section>  
<!-- END: ABOUT -->
```

Selanjutnya kita ke file mainstyle.css. Setelah kode sebelumnya, tulislah kode css berikut:

```
/* -----  
** ABOUT  
** ----- */  
.about {  
    padding-top: 100px;  
    padding-bottom: 250px;  
    background: #f0f0f0 url(..../img/bg_about.png) no-repeat  
bottom right;  
    font-size: 16px;  
}  
.about h2 {  
    font-weight: 300;  
    margin-bottom: 30px;  
}  
.about h2 strong {  
    font-weight: 500;  
}  
.about p {  
    margin-bottom: 30px;  
}
```

Jika kita kembali buka hasilnya pada browser, sekarang kita akan melihat tampilan seperti berikut:



Sudah tampak bukan bentuknya?

Kamu bisa mengganti tulisan "Lorem ipsum"nya dengan deskripsi tentang kamu pribadi :D

Section List Project

Next. Saatnya kita membuat point inti dari apa yang dari tadi kita kerjain. Yaitu section untuk menampilkan list project yang pernah kita kerjain.

Di sini kita akan menerapkan grid style dan sistem grid-nya Bootstrap. Sekedar informasi buat kamu yang belum pernah pake Bootstrap, bahwa di Bootstrap ada sebuah fitur yang sangat keren yang berguna untuk membagi bagian tertentu dari halaman web menjadi beberapa bagian berdasarkan grid. Secara standarnya Bootstrap menerapkan sistem 12 grid, walaupun sebenarnya kita bisa ubah dan tentukan sendiri sesuai kebutuhan kita. Tapi memang pada kenyataannya jarang sekali sih kita mengubahnya, karena memang 12 grid biasanya sudah memenuhi kebutuhan. Sederhananya ketika kita membuat sebuah section atau element tertentu untuk halaman web kita, kita bisa membelahnya menjadi 12 bagian. Gambarannya seperti berikut:

1 col											
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Jadi satu baris (row), bisa dibagi menjadi 12 kolom (col). Misalnya kita ingin memotong halaman web kita menjadi 2 bagian sama besar. Berarti kita bisa memotongnya menjadi 6 col dan 6 col. Ilustrasinya seperti berikut:

6 col	6 col
-------	-------

Kita juga bisa membuat seperti ini, mungkin yang sebelah kiri untuk konten dan yang kanan untuk sidebar :D

8 col	4 col
-------	-------

Nah dalam project kali ini untuk section list project ini kita akan menggunakan 3 kolom seperti berikut:

4 col	4 col	4 col
-------	-------	-------

Btw buat kamu yang masih bingung dan penasaran dengan konsep grid di Bootstrap, kamu bisa pelajari lebih dalam di <http://getbootstrap.com/css/#grid>.

Oke sekarang saatnya kita melanjutkan kode kita. Silahkan buka kembali file **index.html**. Kemudian carilah kode berikut:

```
</section>
    <!-- END: ABOUT -->
```

Setelah ketemu, tulislah kode di bawah ini setelah kode tersebut di atas:

```
<!-- PORTFOLIO -->
<section class="portfolio">
    <div class="container">
        <h2>My Work</h2>
        <div class="text-big text-center">Lorem ipsum dolor
sit amet, consectetur adipiscing elit</div>

    <!-- PORTFOLIO LOOP -->
```

```
<div class="portfolio-loop">
  <div class="row">

    <div class="col-md-4 col-sm-6">
      <a href="#">
        <div class="portfolio-item">
          <div class="thumb">
            
          </div>
          <div class="caption">
            <div class="title">Lorem ipsum dolor sit
amet</div>
            <div class="desc">
              Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
            </div>
          </div>
        </a>
      </div>

      <div class="col-md-4 col-sm-6">
        <a href="#">
          <div class="portfolio-item">
            <div class="thumb">
              
            </div>
            <div class="caption">
              <div class="title">Lorem ipsum dolor sit
amet</div>
              <div class="desc">
                Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
              </div>
            </div>
          </a>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
</div>
</a>
</div>

<div class="col-md-4 col-sm-6">
    <a href="#">
        <div class="portfolio-item">
            <div class="thumb">
                
            </div>
            <div class="caption">
                <div class="title">Lorem ipsum dolor sit
amet</div>
                <div class="desc">
                    Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
                </div>
            </div>
        </a>
    </div>

    <div class="col-md-4 col-sm-6">
        <a href="#">
            <div class="portfolio-item">
                <div class="thumb">
                    
                </div>
                <div class="caption">
                    <div class="title">Lorem ipsum dolor sit
amet</div>
                    <div class="desc">
                        Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
                    </div>
                </div>
            </a>
        </div>
    </div>
</div>
```

```
</div>
</div>
</a>
</div>

<div class="col-md-4 col-sm-6">
<a href="#">
<div class="portfolio-item">
<div class="thumb">

</div>
<div class="caption">
<div class="title">Lorem ipsum dolor sit
amet</div>
<div class="desc">
    Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
</div>
</div>
</div>
</a>
</div>

<div class="col-md-4 col-sm-6">
<a href="#">
<div class="portfolio-item">
<div class="thumb">

</div>
<div class="caption">
<div class="title">Lorem ipsum dolor sit
amet</div>
<div class="desc">
    Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat
```

```
        </div>
    </div>
</div>
</a>
</div>

</div>
</div>
<!-- END: PORTFOLIO LOOP -->

<div class="portfolio-more text-center">
    <a href="#">[ MORE ]</a>
</div>
</div>
</section>
<!-- END: PORTFOLIO -->
```

Setelah itu kembali ke mainstyle.css, kemudian di akhir kode tuliskan kode berikut:

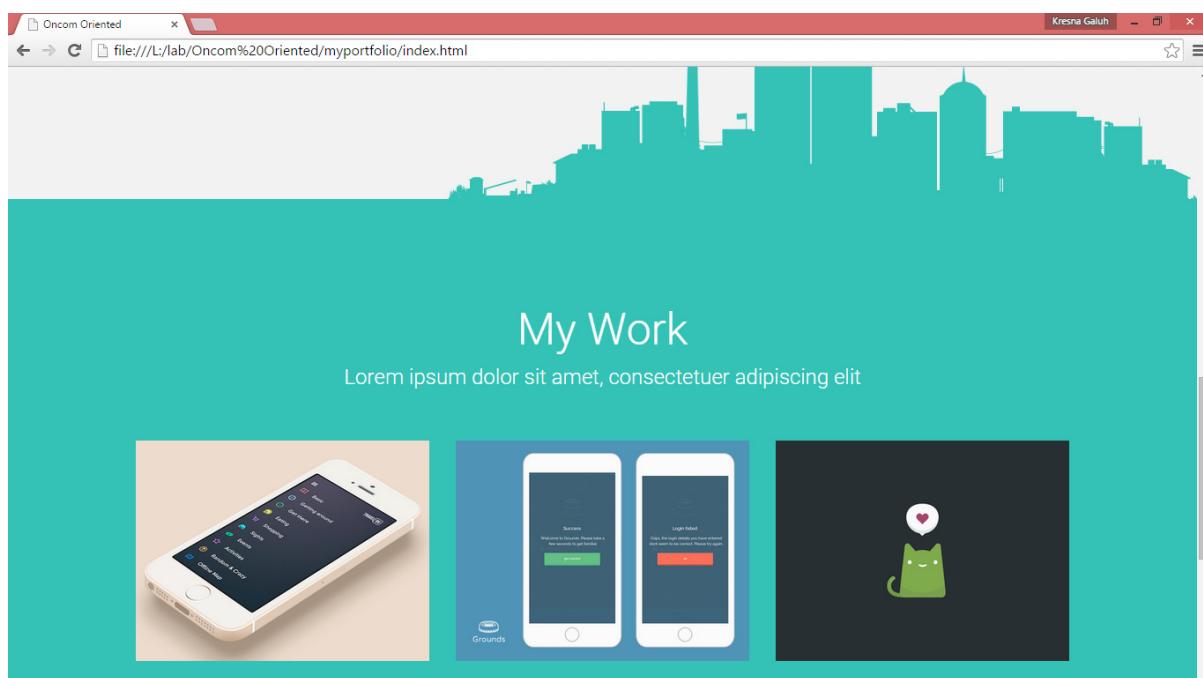
```
/* -----
** PORTFOLIO
** ----- */
.portfolio {
    background: #2abbae;
    padding: 100px 0px;
}
.portfolio h2 {
    font-size: 50px;
    font-weight: 300;
    text-align: center;
    color: #ffffff;
}
.portfolio .text-big {
    font-size: 24px;
    font-weight: 300;
    margin-bottom: 55px;
    color: #ffffff;
}
```

```
.portfolio-loop .portfolio-item {  
    position: relative;  
    overflow: hidden;  
    margin-bottom: 30px;  
}  
.portfolio-loop a,  
.portfolio-loop a:hover,  
.portfolio-loop a:focus {  
    color: #454442;  
    text-decoration: none;  
}  
.portfolio-loop .portfolio-item .thumb .thumb-img {  
    width: 100%;  
    height: auto;  
}  
.portfolio-loop .portfolio-item .caption {  
    position: absolute;  
    top: 0px;  
    left: 0px;  
    right: 0px;  
    width: 100%;  
    height: 100%;  
    background: rgba(255,255,255,0.9);  
    padding: 20px;  
    opacity: 0;  
    -webkit-transition: all 0.4s ease 0s;  
    -moz-transition: all 0.4s ease 0s;  
    -o-transition: all 0.4s ease 0s;  
    transition: all 0.4s ease 0s;  
}  
.portfolio-loop .portfolio-item:hover .caption {  
    opacity: 1;  
}  
.portfolio-loop .portfolio-item .title {  
    font-size: 18px;  
    font-weight: 500;  
}  
.portfolio-loop .portfolio-item .desc {  
    margin-top: 20px;
```

```
}

.portfolio-more {
    margin-top: 30px;
    font-size: 20px;
}
.portfolio-more a {
    color: #ffffff;
    text-decoration: none;
}
.portfolio-more a:hover,
.portfolio-more a:focus {
    color: #e7edec;
    text-decoration: none;
}
```

Jika sudah, maka sekarang tampilan lab kita akan seperti berikut:



Yups, sudah tampak bagian project dengan warna cerah bergaya flat design. Saya contohkan di sini hanya 6 item saja, pada prinsipnya kamu bisa tambahkan berapa saja sesuai kebutuhan kamu.

Section Contact

Bagian penting selanjutnya adalah kontak. Sebenarnya inilah pointnya dari apa yang sudah kita kerjakan sejauh ini. Untuk apa kita membuat dan merapihkan halaman profile kita bahkan membuatnya mudah diakses yaitu menjadi halaman web, untuk apa kemudian kita memberitahukan siapa kita dan memampang apa yang telah kita kerjakan jika kita tidak memberitahukan bagaimana caranya menghubungi kita. Ya, obyektifnya mungkin nanti akan ada orang yang merasa apa yang kita lakukan selama ini adalah sesuatu hal yang bagus, dan tertarik dengan proyek-proyek yang pernah kita lakukan, kemudian menghubungi kita untuk menawarkan project atau kerjasama :D

Konsep bagian kontak kita adalah membagi menjadi dua kolom dengan masing-masing 6 col grid Bootstrap. Bagian kiri kita akan isi dengan kontak kita secara manual dan beberapa link sosmed kita. Sedangkan bagian kanan kita akan bikin contact form dimana nanti pengunjung bisa langsung send pesan kepada kita dengan form tersebut. Tentu saja dalam hal ini form yang akan kita buat nanti hanya sebatas tampilan saja. Dan kalau kamu mau menjalankan fungsinya sebagaimana mestinya kamu bisa menggunakan JavaScript, PHP, ASP atau yang lainnya sebagai mesinnya :)

Sebelum masuk ke kode html-nya. Seperti utang saya di awal, bahwa saya akan sedikit menjelaskan tentang Font Awesome. Maka inilah saatnya. Karena di bagian ini kita akan berhadapan langsung dengan Font Awesome.

Sama seperti Bootstrap, Font Awesome juga telah menyediakan banyak class yang bisa kita gunakan. Bedanya adalah Font Awesome hanya fokus pada pemanfaatan icon. Saat ini, di versi 4.4.0 ini Font Awesome telah menyediakan 585 icon siap pakai.

Cara penggunaannya pun sangat sederhana, kita hanya perlu mengembed css Font Awesome kita di html kita seperti berikut:

```
<link rel="stylesheet" href="path/to/font-awesome/css/  
font-awesome.min.css">
```

Dan tentu saja itu sudah kita lakukan di atas, saat membahas Assets. Setelah itu kita sudah bisa memakai berbagai class yang ada di Font Awesome.

Sebagai contoh misalnya kita ingin menampilkan icon coffee seperti berikut:



Maka kita hanya perlu menggunakan kode seperti berikut:

```
<i class="fa fa-coffee"></i>
```

Class **fa** merupakan class pengenal bahwa kita akan menggunakan Font Awesome. Sedangkan class **fa-coffee** adalah untuk memanggil gambar coffee.

Class yang lain bisa kamu lihat di <http://fortawesome.github.io/Font-Awesome/icons/>. Penerapannya sama saja kok. Misalnya kamu ingin menampilkan icon bergambar rumah maka yang kamu perlukan adalah menulis kode seperti berikut:

```
<i class="fa fa-home"></i>
```

Sederhana sekali bukan?

Oh ya, dua contoh di atas kita menggunakan tag i untuk memanggil Font Awesome-nya. Pada dasarnya kita bisa juga menggunakan tag html yang lain seperti span. Kalau saya justru lebih sering menggunakan tag span ketimbang i. Tapi sebenarnya tidak ada masalah sama sekali, bebas-bebas saja mau menggunakan tag yang mana, yang pasti tag kita harus memiliki display inline secara default.

Oke, setelah penjelasan tentang Font Awesome, sekarang saatnya kita menulis kode untuk section contact kita. Setelah kode section portfolio kita atau lebih tepatnya setelah kode:

```
</section>  
  
<!-- END: PORTFOLIO -->
```

Tulislah kode berikut pada **index.html**:

```
<!-- CONTACT -->  
<section class="contact">  
    <div class="container">  
        <h2>GET <strong>CONNECTED</strong></h2>  
  
        <div class="row">  
            <div class="col-md-6">  
                <div class="contact-social-media">  
                    <p>Lorem ipsum dolor sit amet,  
consectetuer adipiscing elit, sed diam nonummy nibh euismod  
tincidunt ut laoreet dolore magna.</p>  
  
                    <div class="contact-item">  
                        <span class="fa fa-envelope  
fa-fw contact-icon"></span> contact@codepolitan.com  
                    </div>  
  
                    <div class="contact-item">  
                        <span class="fa fa-phone fa-fw  
contact-icon"></span> +62 8999 129 220  
                    </div>  
  
                    <div class="contact-item">  
                        <span class="fa fa-linkedin  
fa-fw contact-icon"></span> CodePolitan  
                    </div>  
  
                    <div class="contact-item">  
                        <span class="fa fa-facebook  
fa-fw contact-icon"></span> CodePolitan  
                    </div>  
  
                    <div class="contact-item">  
                        <span class="fa fa-twitter fa-"
```

```

    fw contact-icon"></span> @CodePolitan
        </div>

            <div class="contact-item">
                <span class="fa fa-google-plus
fa-fw contact-icon"></span> +CodePolitan
                </div>
            </div>
        </div>

        <div class="col-md-6">
            <form class="contact-form">
                <div class="form-group">
                    <input type="email"
class="form-control" placeholder="Name">
                </div>
                <div class="form-group">
                    <input type="email"
class="form-control" placeholder="Email address">
                </div>
                <div class="form-group">
                    <input type="password"
class="form-control" placeholder="Subject">
                </div>
                <div class="form-group">
                    <textarea class="form-control"
rows="3" placeholder="Message"></textarea>
                </div>

                    <button type="submit" class="btn
btn-default btn-lg">SEND</button>
                </form>
            </div>
        </div>
    </div>
</section>
<!-- END: CONTACT -->

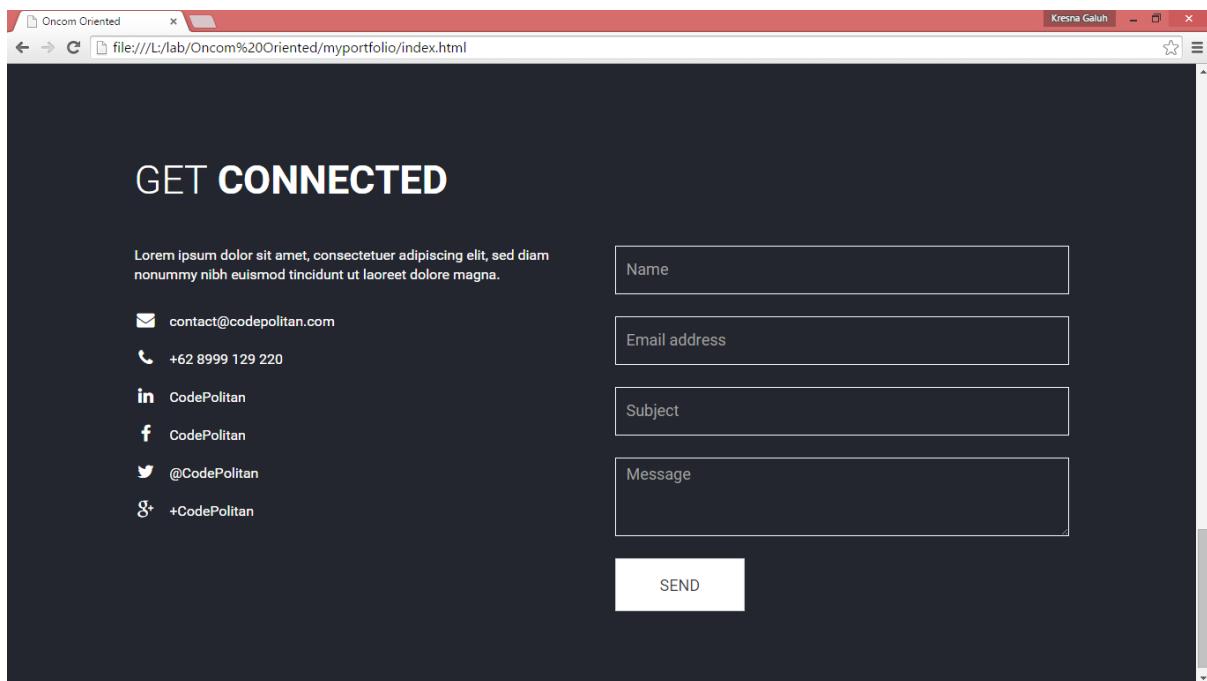
```

Selanjutnya pada file mainstyle.css tulislah kode berikut setelah kode-kode css sebelumnya (di paling bawah):

```
/* -----
** CONTACT
** ----- */
.contact {
    background: #1b1e25;
    color: #ffffff;
    padding: 100px 0px;
}
.contact .contact-social-media {
    margin-bottom: 40px;
}
.contact h2 {
    font-size: 45px;
    font-weight: 300;
    margin-bottom: 50px;
}
.contact h2 strong {
    font-weight: 600;
}
.contact p {
    font-size: 16px;
    margin-bottom: 30px;
}
.contact .contact-item {
    font-size: 16px;
    margin-bottom: 20px;
}
.contact .contact-item .contact-icon {
    font-size: 20px;
    margin-right: 10px;
}
.contact-form .form-group {
    margin-bottom: 25px;
}
.contact-form .form-control {
    background: transparent;
```

```
border: 1px solid #ffffff;
-webkit-border-radius: 0px;
-moz-border-radius: 0px;
-ms-border-radius: 0px;
-o-border-radius: 0px;
border-radius: 0px;
font-size: 18px;
color: #ffffff;
}
.contact-form input.form-control {
height: 55px;
line-height: 55px;
}
.contact-form .form-control::-webkit-input-placeholder,
.contact-form .form-control:-moz-placeholder,
.contact-form .form-control::-moz-placeholder
.contact-form .form-control:-ms-input-placeholder {
color: #ffffff;
}
.btn.btn-default {
-webkit-border-radius: 0px;
-moz-border-radius: 0px;
-ms-border-radius: 0px;
-o-border-radius: 0px;
border-radius: 0px;
padding-left: 50px;
padding-right: 50px;
}
.btn.btn-lg {
height: 60px;
line-height: 60px;
padding-top: 0px;
padding-bottom: 0px;
}
```

Sekarang jika kita melihat hasilnya pada browser, kita akan melihat sudah tampak section contact seperti gambar berikut:



Footer

Setelah semuanya selesai selanjutnya sebagai pelengkap kita tinggal menambahkan footernya saja. Kembali ke index.html, carilah kode berikut:

```
</section>
<!-- END: CONTACT -->

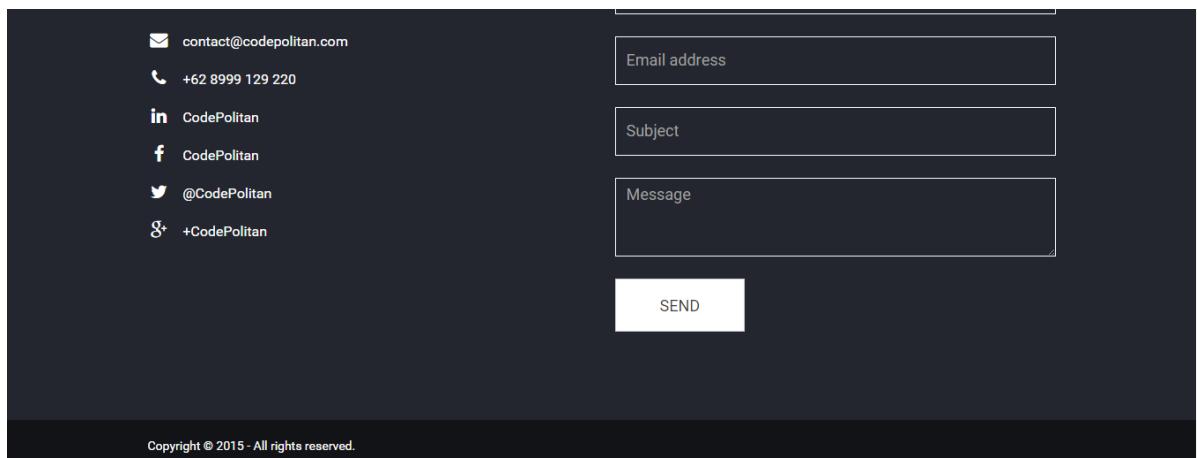
Kemudian tulis kode berikut:
<!-- FOOTER -->
<footer class="footer">
    <div class="container">
        Copyright © 2015 - All rights reserved.
    </div>
</footer>
<!-- END: FOOTER -->
```

Dan terakhir, tambahkan kode css berikut di akhir file mainstyle.css:

```
/*
** FOOTER
** -----
*/
.footer {
```

```
background: #0d0e11;  
color: #ffffff;  
padding: 20px 0px;  
}
```

Maka jika kita membukanya melalui browser saat ini web portfolio kita sudah memiliki footer seperti berikut:



Hasil Akhir

Sampai sini selesai sudahlah proyek kita kali ini. Tampilannya sudah sempurna seperti yang kita inginkan. Berikut tampilan totalnya:



Sebenarnya kamu bisa mengembangkan banyak hal lagi setelahnya. Mungkin dengan menambah animasi seperti efek parallax seperti di web portfolio saya di <http://www.kresnagaluh.com>. Intinya kita bisa bikin sekreatif mungkin selama kita mau ngulik :D

Kamu bisa melihat hasil akhir projectnya di tautan berikut: <https://dl.dropboxusercontent.com/u/117499135/myportfolio/index.html>.

Kamu juga bisa mendownload file projectnya di: <https://dl.dropboxusercontent.com/u/117499135/myportfolio.zip>.

Akhirnya, demikianlah pembelajaran kita pada Oncom Oriented kali ini. Semoga bermanfaat :D

Jika ada yang kurang jelas atau kamu punya pertanyaan terkait project ini, silahkan ditanyakan melalui kresnagaluh@gmail.com, selama saya bisa menjawabnya insya Allah akan saya bantu. Dan terakhir jika kamu berhasil mengikuti materinya dengan baik, upload juga dong hasil karya kamu. Buat kamu yang portfolionya kreatif akan kami tampilkan di web kami :D

Selamat bersenang-senang!

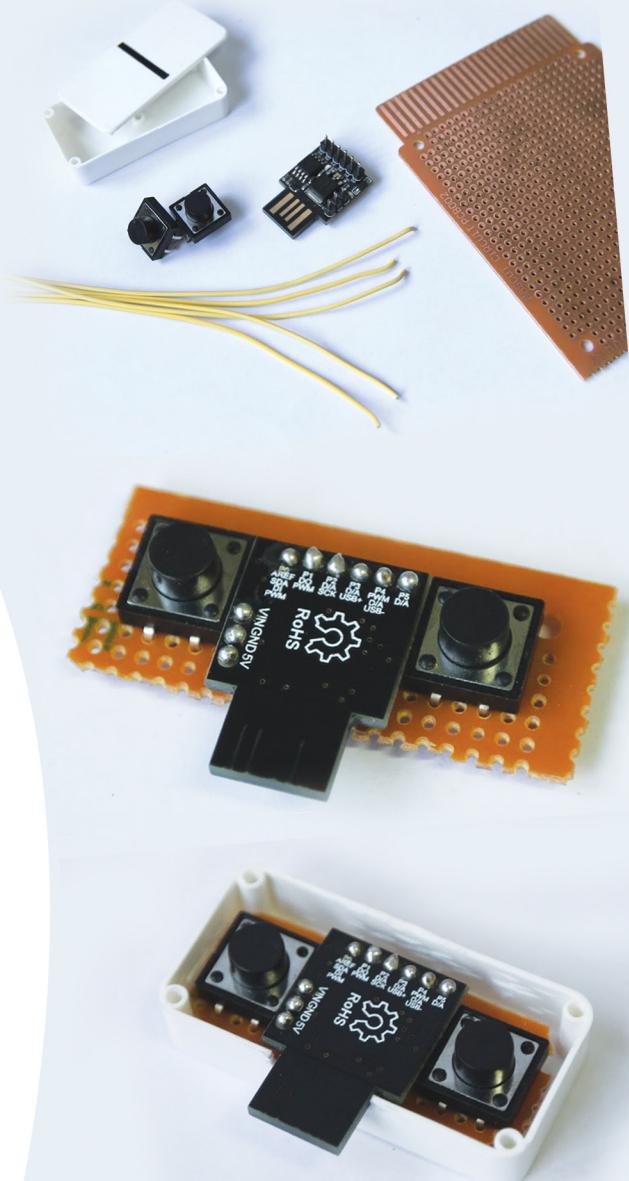
Ayo,...DAPATKAN DIGISPARK BOARD **GRATIS!!**

Caranya:

Bikin web portfolio pribadimu sekeren mungkin, terus kirim url dan designya ke contact@codepolitan.com sebelum **31 Oktober 2015**.

Portfoliomu akan kami publikasikan di web CoderTalk dan channel CodePolitan lainnya.

Buat dua orang dari kamu yang portfolionya paling keren, akan mendapatkan **Digispark Board** secara **GRATIS!!**



{• CodePolitan


Learn Programming. Together.



Membuat Joystik Menggunakan Digispark

Siapa bilang hanya pabrik dan perusahaan besar saja yang bisa membuat joystick? Dengan menggunakan Digispark Board dan mengikuti tutorial ini, kita bisa membuat joystick sendiri untuk game yang kita buat.

TONI HARYANTO

Digispark adalah development board berbasis mikrokontroler Attiny85, seperti halnya board Arduino tapi dengan harga yang lebih murah, ukuran yang lebih kecil dan kemampuan yang lebih terbatas. Digispark pertama kali dibuat oleh Erik Kettenburg, seorang programmer asal Portland, Amerika Serikat, melalui proyek yang dipublikasikan di Kickstarter pada bulan Agustus 2012 dan berhasil mengumpulkan dana sebesar 313 ribu dollar US.

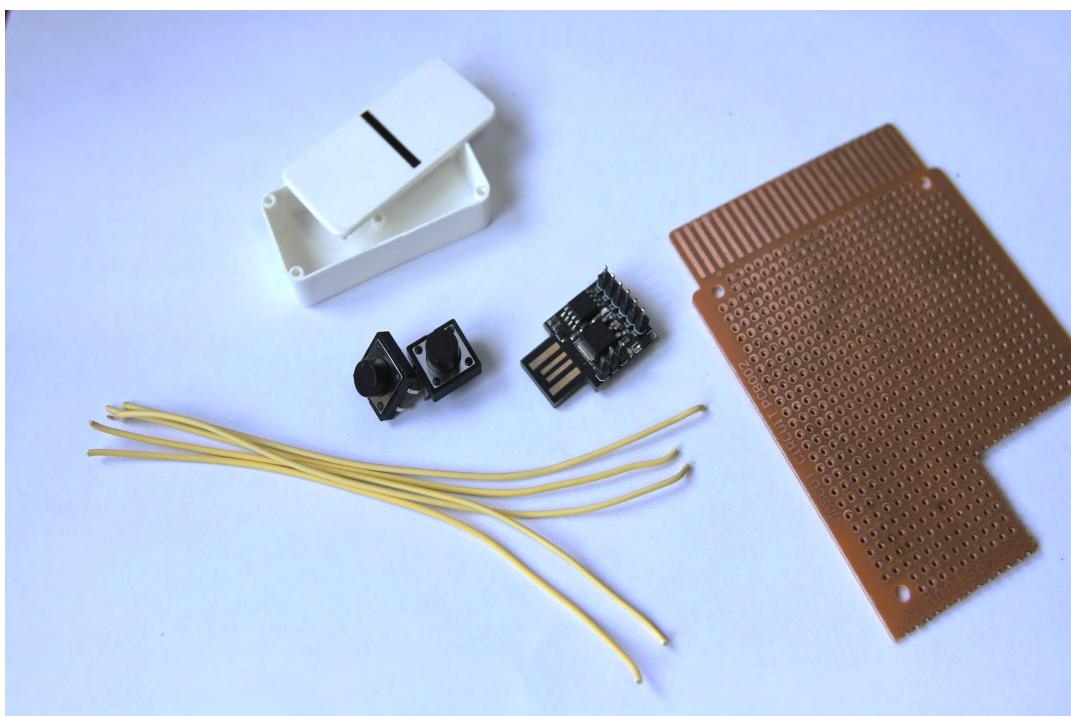
Awalnya Erik biasa menggunakan board Arduino untuk setiap proyeknya, baik proyek yang besar maupun kecil. Namun dia sangat menyayangkan setiap kali membuat proyek menggunakan board Arduino dan meninggalkan resource Arduino tersebut untuk proyek-proyek kecil yang sebetulnya hanya memerlukan beberapa pin saja. Hal tersebut membuat biaya produksi lebih besar untuk keperluan yang sederhana. Dari situlah dia akhirnya membuat board dengan ukuran yang lebih kecil dengan biaya produksi yang lebih murah. Menggunakan mikrokontroller Attiny85 yang hanya memiliki 6 pin input/output, Digispark lahir dan beredar di pasaran dengan harga \$12 atau sekitar 150 ribu rupiah. Selain itu Erik juga mempublikasikan Digispark sebagai perangkat Open Source di bawah lisensi Creative Common Attribution-ShareAlike (CC BY_SA 3.0) sehingga siapapun boleh mengubah, memodifikasi, membuat sendiri dan bahkan memproduksi dan menjualnya.

Pada weekend project kali ini, saya akan menunjukkan bagaimana memanfaatkan Digispark pada proyek sederhana yang mudah dan menarik, cocok buat refreshing akhir pekan Kamu melepas penat mengerjakan pekerjaan koding harianmu hehe.. Saya menggunakan Digispark karena memang proyek yang akan Kamu lihat ini tidak memerlukan banyak pin. Selain itu, Digispark juga dapat ditemui dengan mudah di toko-toko online di Indonesia dengan kisaran harga 50-60 ribu rupiah. Harganya bisa murah karena biasanya Digispark yang beredar di Indonesia adalah clone buatan Cina. Meski demikian kualitasnya tetap bisa diandalkan, paling tidak works fine buat project kecil kita, dibanding bila mesti beli ke luar negeri yang selain memerlukan biaya pengiriman yang lebih besar, juga waktu pengiriman yang relatif lebih lama. But it's up to you guys.

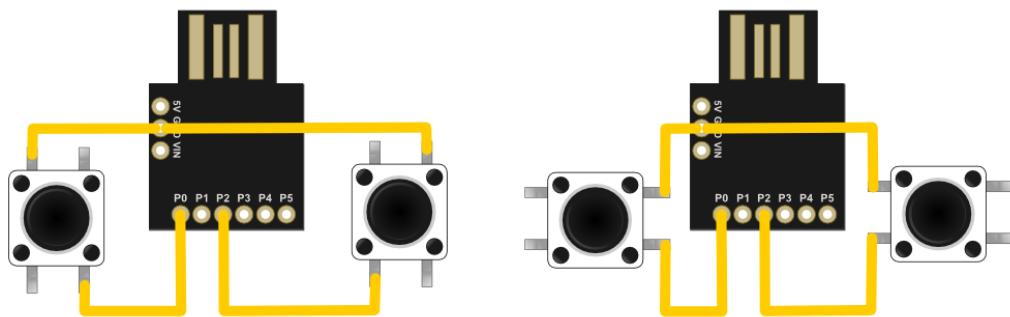
Digispark
juga dapat
ditemui dengan
mudah di toko-
toko online
di Indonesia
dengan
kisaran
harga 50-60
ribu rupiah".

Ada dua jenis board yang beredar di pasaran. Ada yang menggunakan port micro USB seperti yang sering kita temui pada smartphone pada umumnya. Ada juga yang menggunakan konektor USB tipe A on board dan ini adalah versi official. Kelebihan dari jenis dengan konektor on board USB adalah board tersebut dapat langsung dipasang ke colokan USB yang ada di komputer, tapi saya sarankan buat beli yang versi micro USB karena kita dapat menggunakan kabel data micro USB dan itu lebih memudahkan untuk development. Akan tetapi pada proyek ini saya menggunakan jenis board dengan on board USB dan alhasil saya tetap harus beli kabel ekstensi USB female dan itu lebih sulit dicari dibanding kabel micro USB.

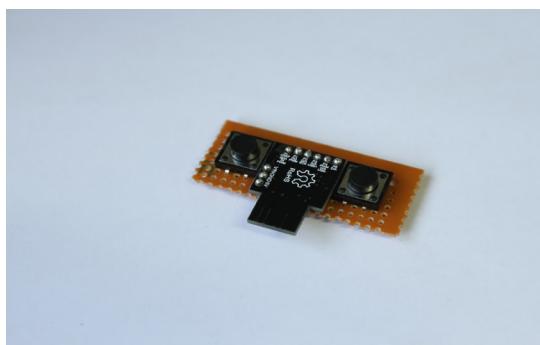
Untuk proyek kali ini saya tidak banyak menggunakan bahan, hanya dua pushbutton dan prototype PCB. Selain itu Kamu bisa manfaatkan kabel untuk menyambung pin Digispark dengan kaki pushbutton, atau langsung menghubungkannya dengan timah melalui lubang-lubang pada PCB. Saya juga memanfaatkan kotak kecil bekas mainan domino punya anak saya sebagai cangkang joysticknya. Ini tidak wajib. Kamu bisa pakai kotak korek api atau kotak lainnya atau membiarkannya tanpa kotak, tidak masalah.



Sekarang saatnya untuk merangkai komponen. Masing-masing tombol dihubungkan ke pin P0 dan P2 dengan mode *active-low* yang artinya kaki lain dari tombol tersebut dihubungkan ke *ground*. Dengan demikian ketika kita menekan tombol tersebut, pin 0 dan pin 2 akan membaca sinyal LOW. Kita juga akan menggunakan *internal pull-up resistor* dari mikrokontroller sehingga ketika tombol tidak ditekan, sinyal pin akan dibaca HIGH. Gambar berikut adalah ilustrasi bagaimana menghubungkan kaki-kaki tombol ke pin pada Digispark.



Kamu dapat langsung memasangnya di proto PCB. Saya memasangnya dengan susunan seperti pada gambar di atas. Kamu dapat menggunakan salahsatu dari dua cara di atas. Pastikan Kamu tidak salah menghubungkan kaki tombolnya. Dan karena saya punya kotak yang tidak terlalu besar, maka saya memotong beberapa bagian dari sisi protoPCB supaya bisa muat ke dalam kotak, yang penting tidak memotong jalur listriknya.



Now, it's time for code! Digispark sudah menanamkan *bootloader* “Micronucleus Tiny85” pada Attiny85 sehingga ia dapat berlaku sebagai USB device dan dapat diprogram menggunakan Arduino IDE. Buka Arduino IDE di komputermu. Saya asumsikan Kamu sudah menginstall digispark library di Arduino IDE. Kalo belum, silakan install terlebih dahulu dengan cara-cara seperti yang dijelaskan di halaman Wiki Digispark (<http://digistump.com/wiki/digispark/tutorials/connecting>) atau baca panduan instalasi library Digispark pada artikel “Memprogram Dunia Fisik Menggunakan Arduino” di halaman 99. Adapun kode-nya seperti berikut:

```
1 // load library keyboard
2 #include "DigiKeyboard.h"
3
4 // set variable pin
5 // tombol pin 0 untuk fungsi enter dan
6 // tombol pin 2 untuk fungsi spasi
7 int tombolA = 0;
8 int tombolB = 2;
9
10 // set variable status tombol ditekan
11 char pressed = 0;
12
13
14 void setup() {
15     // set mode pin
16     pinMode(tombolA, INPUT);
17     pinMode(tombolB, INPUT);
18
19     // set default value pin sebagai HIGH
20     digitalWrite(tombolA, HIGH);
21     digitalWrite(tombolB, HIGH);
22 }
23
24
25 void loop() {
26     // panggil fungsi ini untuk menjaga koneksi dengan
27     // komputer
28     // jika tidak, komputer akan mengira digispark ber-
29     // henti bekerja
30     // dan memunculkan error
```

```
31 DigiKeyboard.sendKeyStroke(0);  
32  
33 // selama tombol fungsi enter ditekan  
34 if(digitalRead(tombolA) == LOW){  
35     // kirim karakter enter ke komputer  
36     DigiKeyboard.sendKeyStroke(KEY_ENTER);  
37     DigiKeyboard.sendKeyStroke(0);  
38 }  
39  
40 // bila tombol fungsi spasi ditekan  
41 if(digitalRead(tombolB) == LOW){  
42     // bila tombol baru ditekan  
43     if(pressed == 0) {  
44         // kirim karakter spasi ke komputer  
45         DigiKeyboard.sendKeyStroke(KEY_SPACE);  
46         DigiKeyboard.sendKeyStroke(0);  
47         // set status tombol sudah ditekan  
48         pressed = 1;  
49     }  
50 }  
51  
52  
53 // apabila tombol fungsi spasi dilepas  
54 if(digitalRead(tombolB) == HIGH){  
55     // kembalikan status ditekan ke 0  
56     pressed = 0;  
57 }  
}
```

Pada dasarnya cara kerja aplikasi joystick ini sama seperti keyboard. Kita mensimulasikan tombol keyboard ditekan dengan mengirimkan sinyal melalui USB sedemikian sehingga komputer menerjemahkannya sebagai penekanan tombol keyboard. Seperti kita lihat pada kode di atas kita menggunakan library DigiKeyboard bawaan Digispark.

Pada proyek ini kita hanya menngunakan dua buah tombol saja, maka pada baris 7 dan 8 kita inisiasikan variabel kedua tombol dengan nomor pin, yakni pin 0 kita beri nama tombolA dan pin 2 kita beri nama tombolB.

Pada fungsi `setup()` kita set mode pin menjadi INPUT dan set nilai default ke HIGH.

Pada baris 32 kita cek apakah tombolA ditekan. Apabila ditekan (value LOW) maka kita kirim karakter enter ke komputer. Demikian juga pada baris 39 kita cek bila tombolB ditekan maka kita kirimkan karakter spasi ke komputer. Perbedaan dari kedua blok kode tersebut terletak pada aksi pengiriman karakternya. Pada tombolA program akan terus menerus mengirimkan karakter enter selama pin 0 bernilai LOW atau dengan kata lain selama tombol masih ditekan. Berbeda dengan tombolB. Sebelum karakter spasi ditekan, kita cek terlebih dahulu apakah variabel *pressed* bernilai 0 (baris 41). Bila variabel *pressed* bernilai 0 itu artinya tombol tersebut baru saja ditekan, sehingga program boleh mengirimkan karakter spasi. Di baris 46 setelah program mengirim karakter spasi, kita set variabel *pressed* menjadi 1, dan pada baris 51 kita cek apakah tombol sudah dilepas sehingga variabel *pressed* mesti direset terlebih dahulu. Hal ini supaya untuk *loop* selanjutnya (selama tombol ditekan) karakter spasi tidak akan dikirim sampai pengguna melepas tombol tersebut terlebih dahulu. *That's all about the code.*

Silakan dicompile dan diupload kodennya ke digispark, dan kamu bisa mencoba hasilnya langsung. Untuk tes, kamu bisa coba program notepad dan tempatkan kursor disitu. Apabila karakter spasi dan enter muncul ketika tombol-tombol ditekan, itu artinya proyek akhir pekan Kamu berhasil! Yeeaaah!! :D

Aaaand, supaya joystick Kamu lebih asyik dipakai, Kamu bisa coba bikin game sederhana menggunakan Construct 2, atau Kamu bisa langsung coba performa joystick Kamu dengan memainkan game FlappyNyan yang sudah kami siapkan dan dapat diunduh di Chrome web store disini <https://chrome.google.com/webstore/search/flappynyan>.

Selamat berakhir pekan! Salam karya!

model diperankan oleh Hendri Lubis :v



*Demo aplikasi Digispark Board joystick pada permainan
FlappyNyan*

Don't worry
if it doesn't
work right.

If
everything
did,
you'd be
out of a job.

Mosher's Law of
Software Engineering

c<>dertalk

NGOBROL SANTAI BARENG PAK WENES KUSNADI, CTO SRIBULANCER.COM

Oleh Ahmad Oriza



Halo teman-teman Coders, kali ini kita akan ngobrol langsung dengan Pak Wenes Kusnadi, Chief Technology Officer & Co-Founder dari Sribulancer.com. Namun sebelum itu saya ingin mengenalkan singkat terlebih dahulu apa itu Sribulancer dan siapa Pak Wenes Kusnadi. Bagi yang sudah tahu bisa skip bagian ini.

Sribulancer.com merupakan salah satu online platform untuk mencari pekerja freelance terbesar di Indonesia baik untuk kebutuhan pribadi maupun bisnis. Sribulancer.com bisa dibilang mirip dengan online platform lain seperti Freelancer.com dan Projects.co.id (software job). Sribulancer launch pada bulan September 2014, tercatat sampai saat ini sudah beragam job post yang published. Seperti website development, graphic design, photography, data entry sampai kebutuhan translasi. Merujuk pada website officialnya sudah ada 2.336 job post 37.537 freelancer terdaftar dengan nilai transaksi proyek mencapai Rp. 5,502,863,245.

“Banyak-banyaklah menambah ilmu diluar keahlian kamu sendiri dan jangan idealis.”

Pak Wenes Kusnadi sendiri merupakan seorang experienced di bidang teknologi. Beliau adalah programmer Ruby dan PHP. Selain khatam di ranah backend beliau juga berkompeten dalam ranah frontend seperti menguasai HTML, CSS, Javascript dan Jquery (Javascript library). Beliau telah bekerja di beberapa

perusahaan sebelum akhirnya berlabuh menjadi peracik teknologi di Sribulancer.com. Pak Wenes pernah bekerja sebagai assistant IT manager untuk PT. Mount Scopus Indonesia juga pernah bekerja sebagai programmer di PT. Exertainment Indonesia.

Teman-teman dapat berinteraksi dengan beliau di profile linkedin nya pada tautan berikut <https://www.linkedin.com/pub/wenes-kusnadi/52/93b/b5b/en>.

Selanjutnya yuk simak wawancara kami dengan Pak Wenes, bagaimana Pak Wenes men-delivery platform nya kepada customer, teknologi dibelakang layar, hingga masalah dan solusi yang dihadapi seorang CTO Sribulancer.com. Yuk disimak ☺

Hallo Pak Wenes. Boleh gak kita ngobrol ngobrol tentang perjalanan Pak Wenes sebagai CTO Sribulancer.com ☺. Banyak teknisnya yah Pak, pembaca kita geek semua?

Hi Riza, boleh – boleh. Kalo ada pertanyaan yg saya bisa jawab pasti saya share ☺

Pertanyaan perdana nih Pak, mengapa Bapak tertarik join dengan Sribulancer.com dan akhirnya meninggalkan pekerjaan lama dan memilih dunia startup?

Oh sebenarnya saya dari awal bergabung di sribu.com dulu pada 2011, baru setelah 3 tahun jalan kita (Ryan Gondokusumo dan Saya) rencanakan untuk membuat produk kedua yaitu sribulancer.com.

Awalnya mulanya sebelum Saya memutuskan bergabung, Ryan menceritakan ide & visi dia kedepan dan juga pada waktu itu belum ada pemain lokal yg besar. Industri startup saja masih sedikit pemainnya.

Jadi Saya merasa ada peluang di bisnis ini & Ryan bisa meyakinkan saya untuk full time bergabung bersamanya.

Ceritain teknologi di belakang layar Sribulancer.com dong Pak. Jika terlalu panjang mungkin bisa jelaskan bagian programming language, framework, database engine, dan server yang digunakan saja? Juga mengapa memutuskan untuk memilih teknologi tersebut?

Oke, Saya coba ceritakan dari awal Sribu.com ya. Awalnya Saya menggunakan PHP tanpa framework dan MySQL. Setelah hampir setengah tahun berjalan dan ada beberapa developer bergabung, technical advisor Saya menyarankan untuk mencoba Ruby on Rails & MongoDB. Karena PHP yang saya tulis tersebut susah untuk scaling kedepannya & benar-benar tidak ada standarnya.

Selain itu development dengan Rails bisa cukup menghemat waktu dan ditambah lagi Rails pada waktu itu sedang rame-ramenya. Jadi kami putuskan untuk mendevelop Sribu versi kedua dengan Rails & MongoDB, sementara versi PHP – MySQL masih berjalan sambil menunggu versi keduanya selesai.

Dari segi server, awalnya kita masih Pakai share hosting di kenalan Ryan. Tapi karena kita menggunakan Rails, maka kita putuskan untuk pindah ke server VPS yang bisa dimaintain sendiri supaya installasi framework & plugin dll bisa dilakukan sendiri. Kebetulan yang kami Pakai adalah Linode. Sebenarnya masih banyak alternatif lain yang bagus seperti digital ocean, heroku, rackspace dll.

Lalu lanjut ke Sribulancer, kita pakai teknologi yang kurang lebih sama seperti Sribu. Tapi kita pakai versi Rails yang lebih baru dari Sribu.

Kalau boleh tahu ada berapa rata-rata post job per hari di Sribulancer.com?

Untuk di Sribulancer, rata-rata kami memiliki job posting 30+ per hari.

Apa sih fitur yang paling sulit atau yang memakan waktu lama untuk didevelop dalam Sribulancer.com?

Sebetulnya fitur itu tidak ada yang terlalu sulit dan secara teknis semua pasti bisa direalisasikan, tapi yang memakan waktu lama & sering berubah ditengah jalan itu adalah bisnis proses/alurnya.

Tapi kalau diminta sebut 1 fitur yang sulit, Saya bisa sebutkan fitur realtime chat di Sribulancer. Karena fitur itu kolaborasi antara rails, worker, redis, dan node.js. Detilnya bisa di cek di blog developers sribu: <http://developers.sribu.com/2015/02/14/how-do-we-make-sribulancer-as-a-realtime-freelancing-platforms/>

Sampai sekarang ada berapa orang dalam tim developer Sribulancer.com? detilnya boleh gak Pak :p, seperti berapa sys admin, dba, back end programmer, front end magician dan analyst nya?

Di kita ada 3 front-end developers, 3 back-end developers, 1 Project Manager/QA. Kalau urusan sysadmin masih saya handle sendiri dan analyst dihandle Ryan.

Untuk kolaborasi dan management task. Tim Pakai tools dan metode apa saja ya Pak?

Kolaborasi tools, awalnya kita Pakai bitbucket tapi sekarang Pakai github dengan konsep gitflow.

Kemudian untuk project/task management, awalnya kita Pakai basecamp. Kemudian kita pindah ke atlassian jira dengan module agilennya. Sekarang kita menggunakan github issue. Tapi tidak menutup kemungkinan saya akan coba tools seperti trello atau pivotal trackers.

Untuk metode kita gunakan SCRUM tapi belum benar-benar diimplementkan menyeluruh. Yang kami gunakan adalah konsep sprint dan level issue (high priority & low priority)

Apa sih kendala teknis terbesar yang pernah di hadapi di Sribulancer.com? dan bagaimana Bapak dan tim mengatasinya? Misal kena serangan di server, atau pernah down, dan lain lain.

Yang pernah terjadi akhir-akhir ini adalah dimana server redis kami sempat mati jam 1 dini hari dan baru notice setelah jam 6 pagi ditelpón Ryan ☺, ini kendala yang besar karena availability adalah yang terpenting di bisnis website.

Sebagai seorang CTO pasti harus berpikir keras ya Pak gimana caranya platform online terbesar Indonesia ini supaya lancar, fast diakses. Boleh dong diceritain mengenai apa aja yang tim lakukan untuk dalam optimasi Sribulancer.com? misal caching atau konfigurasi server, optimasi front end.

Mengenai optimasi kita ada lakukan google pagespeed, caching di beberapa page yang tinggi trafficnya dan optimasi beberapa fitur lainnya agar sesuai dengan apa yang user kami cari misalnya fitur pencarian/browse freelancer. Fitur ini kami implementasikan dengan ElasticSearch.

Untuk kedepannya tidak menutup kemungkinan kami bisa beralih ke teknologi baru apabila lebih bagus dari yang sekarang.

Kita tinggalin teknisnya nih Pak sekarang, menurut Bapak kriteria developer yang baik itu seperti apa? Mungkin bisa share juga kriteria yang Bapak pegang ketika nerima pelengkap/pengganti pekerja dalam tim? Mudah mudahan berguna untuk teman-teman calon pekerja dan fresh grad.

okay, kalo developer yang saya suka itu yang full stack. Jadi dia mengerti soal front-end, back-end dan DevOps. Dulu memang agak susah mencari developer seperti itu, tapi karena sekarang sudah banyak startup dan di lingkungan startup sendiri umumnya resource terbatas, jadi mau tidak mau seorang developer harus belajar menangani hal – hal diluar keahliannya.

Misalnya back-end developer belajar menstyling layout halaman atau belajar memanage server sendiri. Begitu juga sebaliknya untuk front-end developer. Tidak perlu terlalu detil mempelajarinya.

Lalu jangan terpaku dengan satu teknologi meskipun populer dan sudah ada sekal lama, dulu saya lama berkutat di PHP & MySQL. Sekarang saya rasakan benefitnya mempelajari hal lain seperti Ruby & MongoDB.

Terakhir, menjadi seorang developer bukan berarti hanya menerima pekerjaan mentah – mentah tanpa tahu maksud dan tujuan suatu module dibuat. Sebisa mungkin kritis dan selalu ingin tahu.

Last question Pak, adakah pesan-pesan yang ingin disampaikan kepada teman-teman pembaca? Seperti wejangan gitu Pak :p

Banyak-banyaklah menambah ilmu diluar keahlian kamu sendiri dan jangan idealis.

Belajar Berkarya dari Dicoding

Kresna Galuh D. Herlangga



NEWS ACADEMY CHALLENGE SHOWCASE EVENT LEADERBOARD REWARD [Log In](#)

Bergabunglah Bersama Developer Terbaik Indonesia

Dicoding sebagai Developer Hub #1 di Indonesia menawarkan Challenge, Event & Akademi bagi para Developer dan Komunitas IT untuk berkarya dan berinovasi dalam membuat Aplikasi untuk beragam platform.

Inilah kesempatan untuk terhubung ke Developer terbaik di Indonesia.

[Daftar Sebagai Developer](#)

[Daftar Sebagai Partner](#)

[Daftar Menggunakan Akun Dicoding](#)

Apa perbedaan Developer & Partner?

“Ketika Anda

menemukan sebuah ide brilian yang sesuai dengan passion Anda, maka kejarlah. Jangan berhenti di tengah jalan. Kejarlah dengan segenap kemampuan, ambisi, dan talenta yang Anda miliki. Karena yang membedakan kesuksesan seseorang terletak persistensi dan keyakinan yang ia miliki untuk terus memperjuangkan idenya menjadi sebuah kenyataan.

Belum genap 1 tahun memang, tapi sudah berhasil memberikan dampak yang cukup baik bagi dunia IT di Indonesia. Dicoding, berdiri sejak Januari 2015, digawangi oleh Narenda Wicaksono dan Kevin Kurniawan, merupakan sebuah developer hub yang menawarkan challenge, event & akademi bagi para developer dan komunitas IT untuk

berkarya dan berinovasi dalam membuat aplikasi untuk beragam platform. Sebagai sebuah developer hub, hingga saat ini sudah ada 5025 developer yang bergabung bersama Dicoding dan telah ada 720 aplikasi yang diikutsertakan dalam challenge Dicoding.

Pada kesempatan kali ini, Narendra Wicaksono, CEO Dicoding akan berbagi cerita dan pengalamannya dalam membangun Dicoding kepada Codertalker sekalian. Siapa tau kita bisa kecipratan inspirasi serta bisa turut berkarya membangun peradaban IT di Indonesia.

Narendra Wicaksono adalah seorang sarjana S1 ITB jurusan Teknik Informatika. Sebelum membangun “kendaraannya sendiri” di Dicoding, Narendra pernah bekerja di Microsoft Indonesia (2007-2010) sebagai Technical Advisor dan di Nokia Indonesia (2010-2014) sebagai Developer Manager. Codertalker bisa mengenal Narendra lebih jauh melalui Linkedin-nya di <https://www.linkedin.com/in/narendra>.



Narendra mengutarakan bahwa saat ini, Indonesia memiliki banyak sarjana lulusan program studi ilmu komputer dan informatika, namun hal ini tidak serta merta memudahkan para pemegang kepentingan industri IT untuk menemukan pengembang aplikasi yang handal. Hal ini karena tidak seluruh lulusan ilmu komputer yang memutuskan untuk meneruskan karir menjadi pengembang aplikasi memiliki portofolio dan keahlian yang benar-benar sesuai dengan kebutuhan pasar. Berangkat dari pemikiran sederhana tersebut, Dicoding hadir menjembatani para pengembang aplikasi muda maupun profesional untuk bertemu dengan tantangan-tantangan yang disediakan oleh para pemilik proyek dengan imbalan berupa poin yang dapat ditukar dengan berbagai rewards yang menarik. Harapannya dengan Dicoding akan lahir lebih banyak developer bertenaga dan berbudaya saing untuk Indonesia.

Sering kali kita mendengar di berbagai media bahwa lulusan IT di Indonesia tidak siap terjun ke dunia kerja, banyak lulusan IT yang tidak bisa menyesiakan dengan kebutuhan industri, itulah sebabnya mengapa banyak sekali perusahaan-perusahaan sering kali kesulitan dalam merekrut tenaga kerja IT-nya terutama programmer. Menurut Narendra, sebenarnya permasalahan industri kesulitan dalam menyerap tenaga IT di Indonesia bukan karena banyak lulusan IT di Indonesia yang tidak siap terjun ke dunia kerja atau tidak memiliki skill yang mumpuni, melainkan karena masih terkendala dari segi jarak dan komunikasi. “Sebagian besar lulusan IT yang berpotensi emas berkedudukan jauh dari ibukota, dan sebagian lain masih terkendala dalam bagaimana menyampaikan dan mempertemukan keahlian mereka dengan kebutuhan pasar yang dinamis. Melihat

D
e
C
o
d
e
r
T
a
l
k

kenyataan di atas, permasalahan utama para developer di Indonesia adalah ‘discoverability’. Itu yang berusaha kami jembatani melalui Dicoding.” ungkap pria yang menguasai pemrograman C# ini.

Narendra juga menambahkan bahwa berdasarkan pengalamannya lebih dari 8 tahun menggeluti profesi sebagai developer relations dia telah bertemu dengan beragam developer yang tersebar di seluruh penjuru tanah air. Mulai dari calon-calon developer yang masih menempuh pendidikan tinggi di bidang ilmu komputer hingga mereka yang sudah sangat senior dan mumpuni dengan skill pemrograman kelas dunia. “Meski jumlah yang betul-betul menggeluti profesi sebagai pengembang software tidak banyak, tapi mereka ada dan tersebar di berbagai pelosok Indonesia. Mereka memiliki potensi yang begitu besar!” tambahnya lagi.

Melihat potensi developer Indonesia yang begitu besar namun banyak diantara mereka yang terpaksa mengubur keahliannya untuk melakoni profesi lain karena kendala-kendala yang disebutkan diatas, Narendra dan Kevin merasa terpanggil untuk melakukan “sesuatu”. Itulah akhirnya di penghujung tahun 2014, Narendra memilih untuk berhenti dari pekerjaannya yang sudah bisa dibiliang menjanjikan (bekerja di perusahaan IT multinasional) untuk kemudian mengembangkan Dicoding bersama Kevin.

Jangan sangka dalam perjalanan sukses dan berkembang suatu startup akan mulus dan tanpa hambatan. Selalu saja ada hal-hal tidak terduga yang merintangi. Pun demikian dengan perjalanan Dicoding. Terlebih lagi tim Dicoding berdomisili di 4 kota yang berbeda. Narendra mengatakan bahwa hambatan dan tantangan terbesar yang dihadapinya ketika menjalankan Dicoding adalah transisi mindset dari pegawai menjadi mindset entrepreneur. Ketika menjadi pegawai kita hanya perlu memikirkan jobdesc kita sendiri dan tenang-tenang saja karena pasti setiap akhir bulan akan menerima gaji dari perusahaan. Namun ketika menjadi entrepreneur terlebih menjalankan sebuah early startup terkadang kita harus menjalankan peran sebagai CEO sekaligus sekretaris, admin

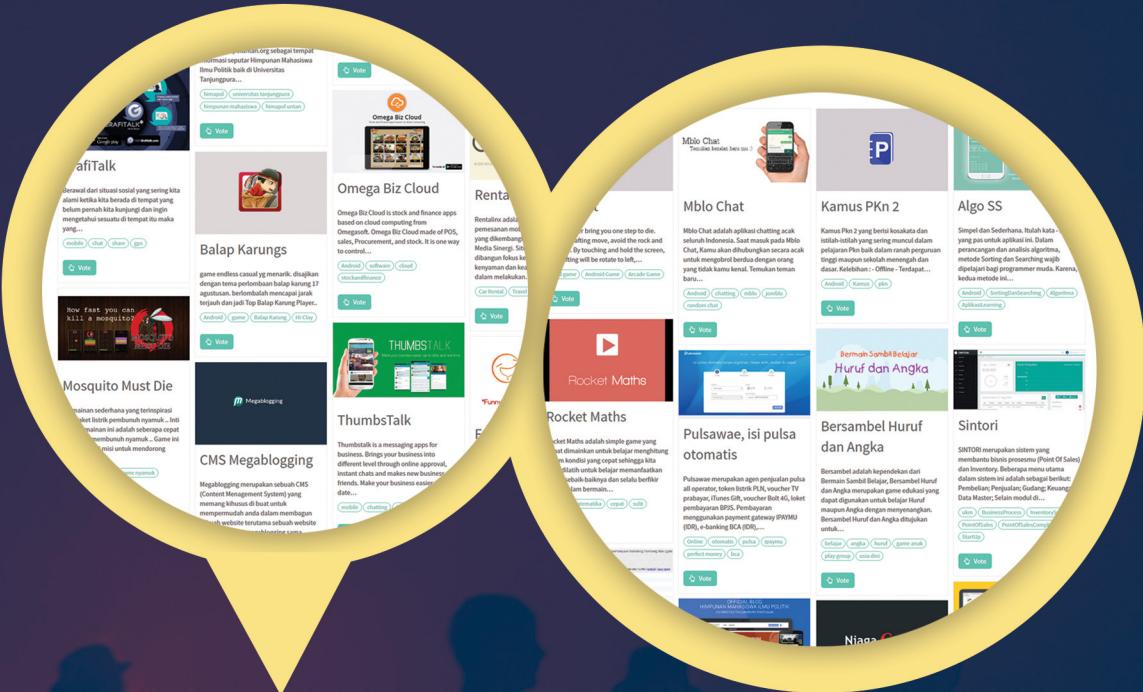
dan marketingnya sekaligus sampai bisa membangun tim yang lebih besar lagi. Yang dipikirkan juga bukan berapa banyak gaji yang bisa kita terima, melainkan bagaimana kita bisa memberikan remunerasi yang layak untuk tim kita.

Harapannya Dicoding mampu melahirkan lebih banyak developer bertalenta dan berdaya saing untuk masyarakat Indonesia. Dan sebagai penutup, mari kita dengar pesan dari co-founder Dicoding ini:

Teruslah semangat dalam berkarya. Ketika Anda menemukan sebuah ide brilian yang sesuai dengan passion Anda, maka kejarlah. Jangan berhenti di tengah jalan. Kejarlah dengan segenap kemampuan, ambisi, dan talenta yang Anda miliki. Karena yang membedakan kesuksesan seseorang terletak persistensi dan keyakinan yang ia miliki untuk terus memperjuangkan idenya menjadi sebuah kenyataan.

Semoga bermanfaat :D

SHARE & VOTE ! KARYAMU!



KARYALOKAL

EXPLORE PRODUK DIGITAL, BUATAN LOKAL

Follow & Dapatkan Info Terbarunya



Visit Us at www.karyalokal.com

Efisiensikan Kerjaan Codingmu dengan Memanfaatkan Tool Bernama Git

- Ridwan Fajar -

Kenalan Dulu!!

Git merupakan sebuah sistem *revision control* yang dikembangkan pertama kali oleh Linus Torvald yang ditujukan untuk memantau pengembangan kernel Linux. Ironis memang, karena motto Git sendiri adalah “The Stupid Content Tracker” namun kenyataanya, penyedia layanan Git saat ini mempunyai jumlah pengguna yang banyak dari berbagai segmen. Git merupakan *revision control* dengan perintah yang lengkap dan menyediakan operasi *high-level* dan akses penuh hingga ke tingkat internal.

Git membuat sejumlah orang dapat bekerja pada dokumen yang sama pada satu waktu dan meminimalkan menghapus bagian orang lain. Tidak seperti sistem lainnya, setiap Git *working directory* merupakan repositori yang lengkap bersama riwayatnya. Dan dapat terpisah dari server pusat. Terinspirasi dari BitKeeper dan Monotone, Git menjadi sebuah *version control system* yang bersifat terdistribusi.

Git memiliki sejumlah perintah yang terbagi menjadi dua bagian:

- ▶ *High level (porcelain)*, terdiri dari *main porcelain command*, *ancillary command*, dan *interacting with other command*.
- ▶ *Low level (plumbing*, terdiri dari *manipulation command*, *interrogation command*, *synching repositories*, dan *internal helper command*.

Memilih Layanan Git

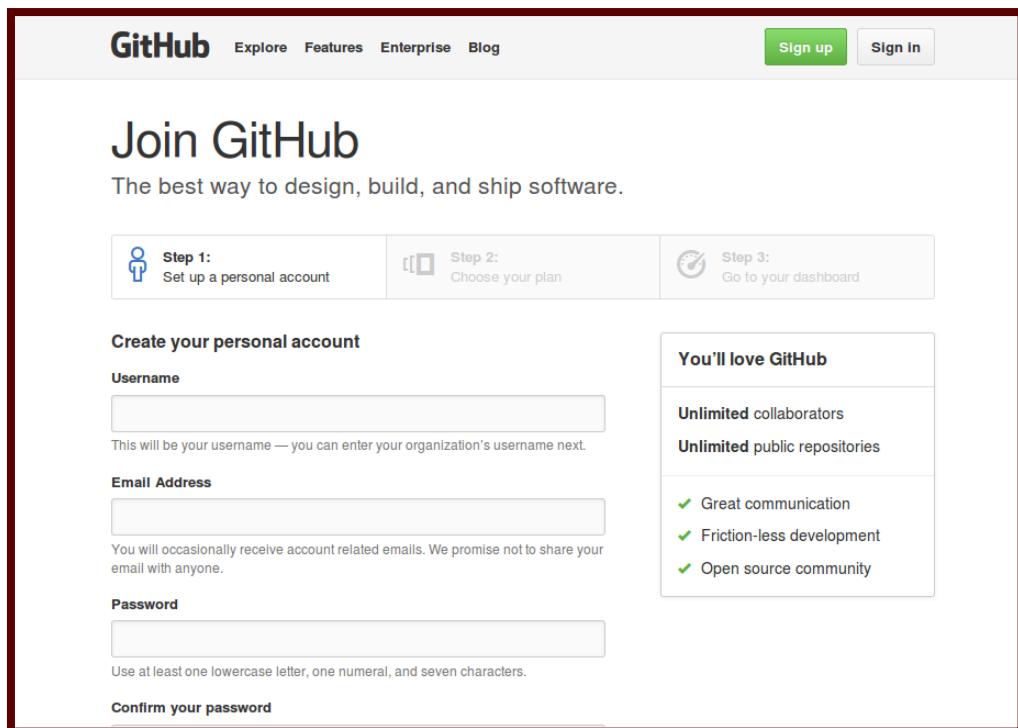
Ada beberapa perusahaan yang menawarkan sejumlah layanan Git untuk para *developer*. Diantaranya ada Github, Bitbucket, dan Gitorious yang menawarkan jasa Git gratis dan berbayar. Ada pula yang sepenuhnya berbayar seperti Assembla. Layanan Git ini biasanya hanya gratis untuk beberapa orang yang terlibat dalam proyek yang disimpan di layanan mereka seperti Bitbucket, ada juga yang membatasi layanan gratis jika proyek tetap dibuka sebagai proyek publik seperti Git. Tentu saja bagi *developer* yang masih sebagai *soloist* ataupun sudah punya tim namun masih terkendala biaya, Anda bisa memilih Gitorious, Github, atau Bitbucket. Jika mempunyai dana, Anda dapat memilih layanan *enterprise* dari Github dan Bitbucket, atau membayar layanan Git kepada Assembla.

Bahkan Anda pun dapat membangun layanan Git Anda sendiri dengan menggunakan Gitlab (<https://about.gitlab.com/>) atau GoGS (Go Git Service, <https://github.com/gogits/gogs>). Dengan memasangnya di *server* kantor atau instansi Anda, dan melakukan sedikit konfigurasi, Anda sudah dapat membangun layanan Git untuk keperluan Anda sendiri. Tentunya Anda pun harus mengelolanya secara mandiri.

Konfigurasi Git untuk Github

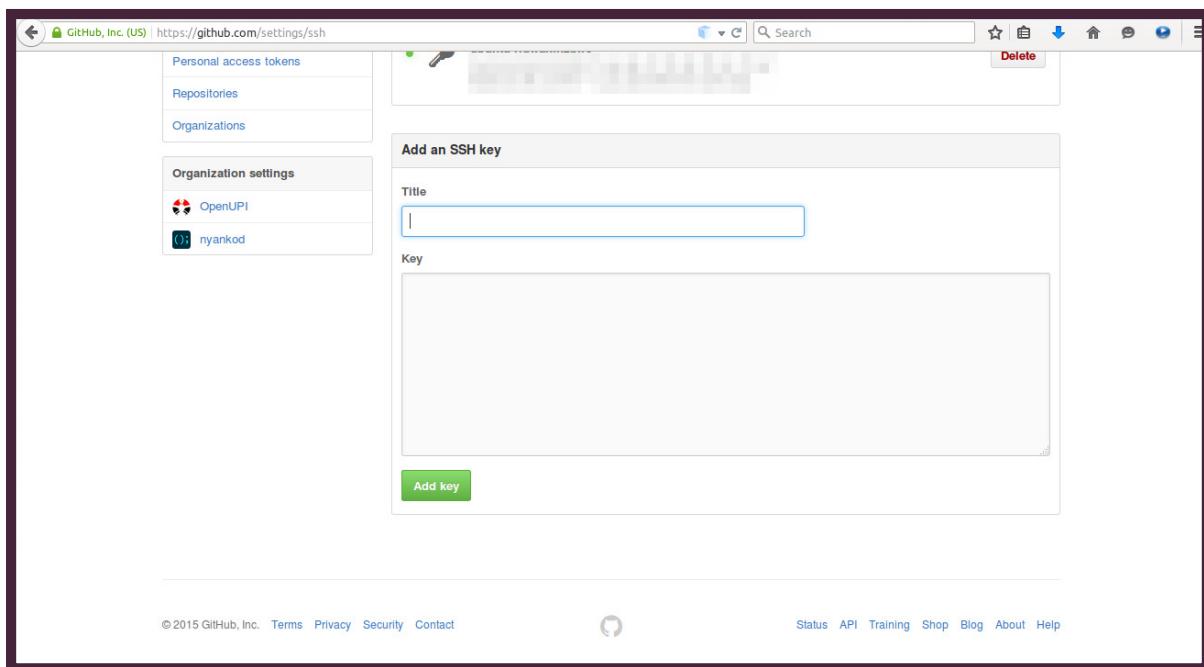
Sebagai perkenalan, kita akan menggunakan layanan Github dan konfigurasinya di Linux (Ubuntu) untuk mencoba Git. Jadi sudah saatnya, khususnya bagi Anda yang masih mahasiswa, untuk tidak menggunakan lagi flashdisk sebagai media pertukaran *source code* untuk tugas besar Anda :D. Berikut adalah langkah awal untuk menggunakan Github:

Daftar terlebih dahulu sebagai *member* di Github (<https://github.com/join>). Kemudian jangan lupa konfirmasi akun Anda melalui e-mail verifikasi yang dikirimkan oleh Github. Setelah melakukan verifikasi silahkan *login* di Github dan mari rasakan sensasinya :D



1. Sesudah *login*, Anda akan melihat halaman depan Github yang menampilkan informasi terkini, aktivitas *commit* dari rekan Anda, repositori yang Anda miliki dan ikuti, serta repositori yang Anda ikut kontribusi di dalamnya. Ada juga tombol “New Repository” yang dapat membantu Anda untuk membuat proyek baru di Github.

2. Hal berikutnya adalah Anda harus melakukan pengaturan *SSH Key* yang terdapat di mesin Anda untuk ditempelkan ke Github, agar Github dapat mengenali mesin Anda. Melalui terminal Linux Anda (atau *command line* sistem operasi lainnya) harus mengeksekusi **ssh-keygen -t rsa -b 4096 -C “email_anda@websitenyaapa.com”**. Selanjutnya Anda akan ditanya dimanakah *key* akan disimpan, biarkan saja secara *default*. Lalu Anda akan ditanya juga apakah akan menggunakan *passphrase* atau tidak. Biasanya *file key* akan tersimpan di /Users/you/.ssh/id_rsa dan /Users/you/.ssh/id_rsa.pub. Jika di Linux biasanya akan tersimpan di /home/username/.ssh atau ~/.ssh (**disini ada gambar artikel-git-3.png, artikel-git-4.png**)

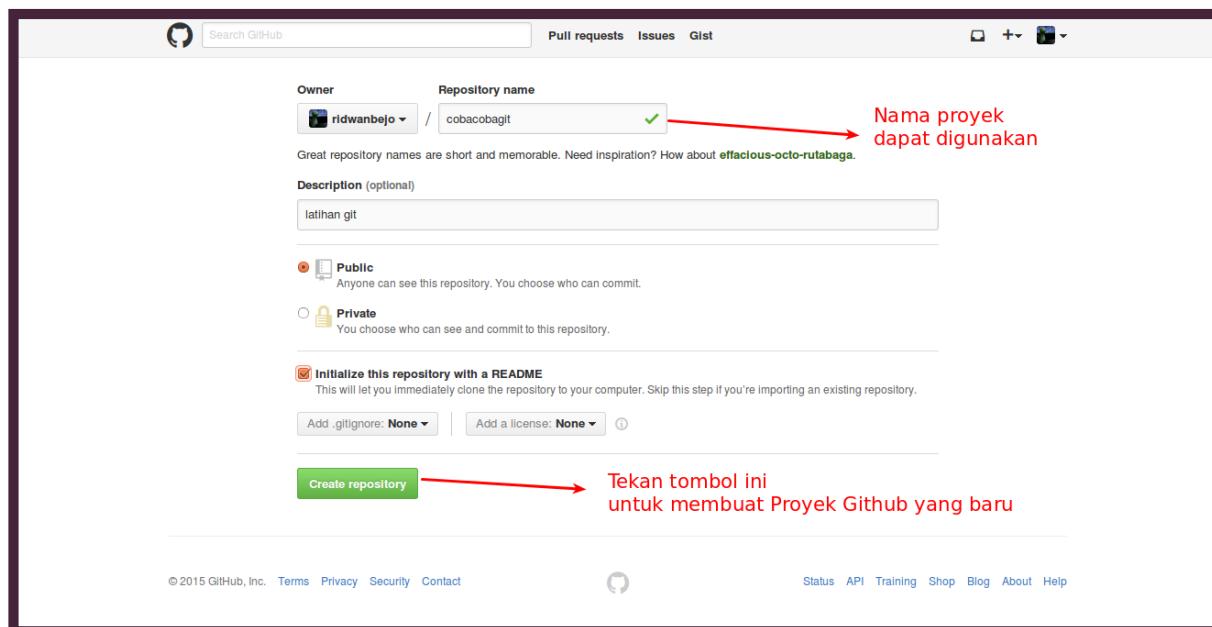


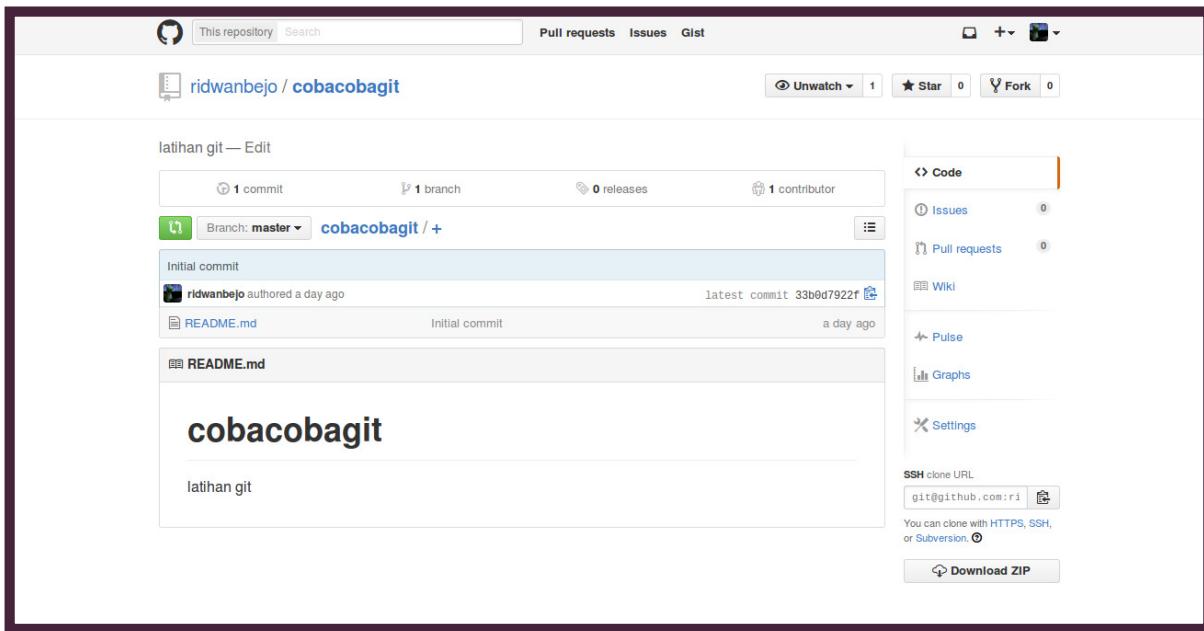
Step 5: Test the connection

To make sure everything is working, you'll now try to SSH into GitHub. When you do this, you will be asked to authenticate this action using your password, which is the SSH key passphrase you created earlier.

- 1**
- 2** You may see this warning:
- 2** Verify the fingerprint in the message you see matches the following message, then type `yes`:
- 2** If the username in the message is yours, you've successfully set up your SSH key!

3. Selanjutnya salin isi *file id_rsa.pub* kemudian buatlah SSH *key* baru untuk Github melalui menu *Setting -> SSH Key*. Tekan tombol “Add SSH Key” beri nama *key* tersebut, dan tempel isi *file id_rsa.pub* yang telah digenerasi
4. Selanjutnya mari kita coba apakah SSH kita sudah terhubung dengan Github atau tidak, dengan menggunakan perintah **ssh -T git@github.com** di terminal Linux Anda. Jika tidak ada masalah, Anda akan melihat informasi seperti pada gambar berikut.
5. Apabila ada masalah, cobalah untuk ikuti instruksi yang ada di halaman ini <https://help.github.com/articles/generating-ssh-keys/>
6. Terakhir kita akan mencoba membuat sebuah proyek di Github. Tekan tombol “New Repository” yang ada di halaman depan Github sesudah Anda *login*. Kemudian isi nama repositori Anda. Apabila nama repositori belum ada yang menggunakan, Anda akan dapat membuat repositori. Kemudian Anda dapat mengisi deskripsi untuk proyek tersebut. Karena Anda menggunakan versi gratis maka Anda hanya dapat membuat proyek Github Anda menjadi **Public**. Anda pun dapat menginisialisasinya dengan *file README*.



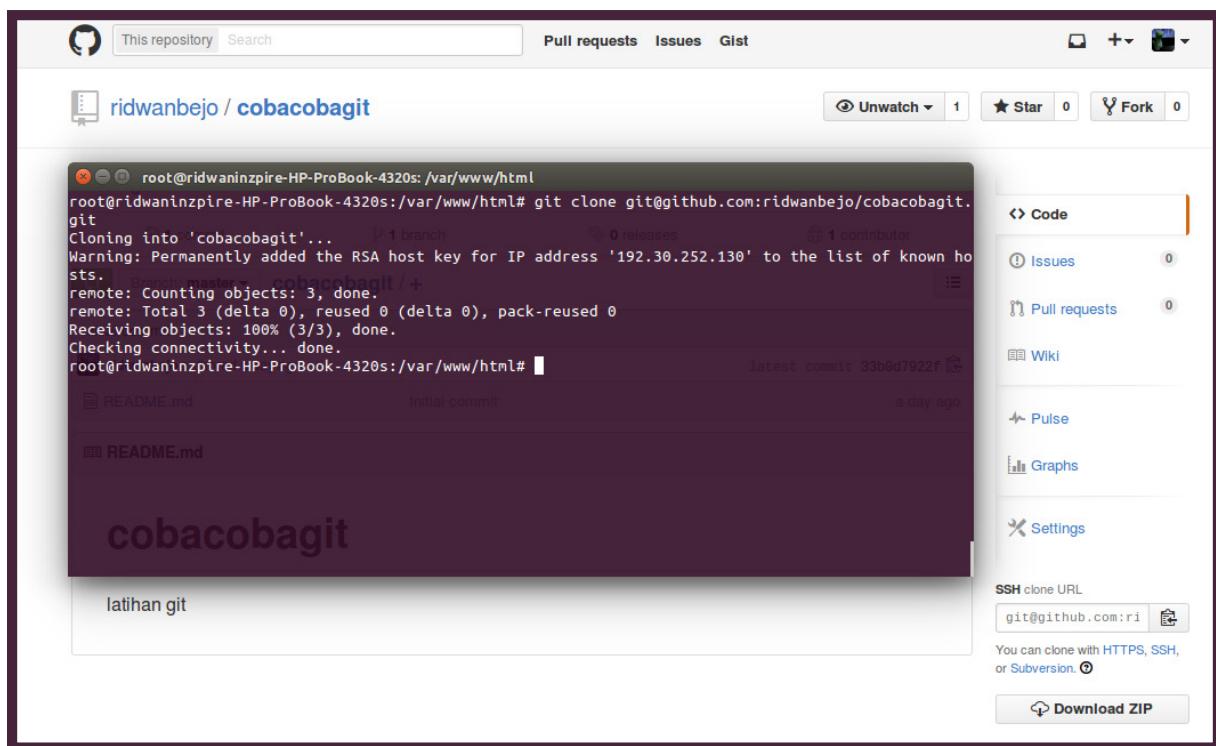


7. Proyek Github pun sudah dapat mulai dikerjakan, dan mari kita pindah ke pembahasan berikutnya :D.

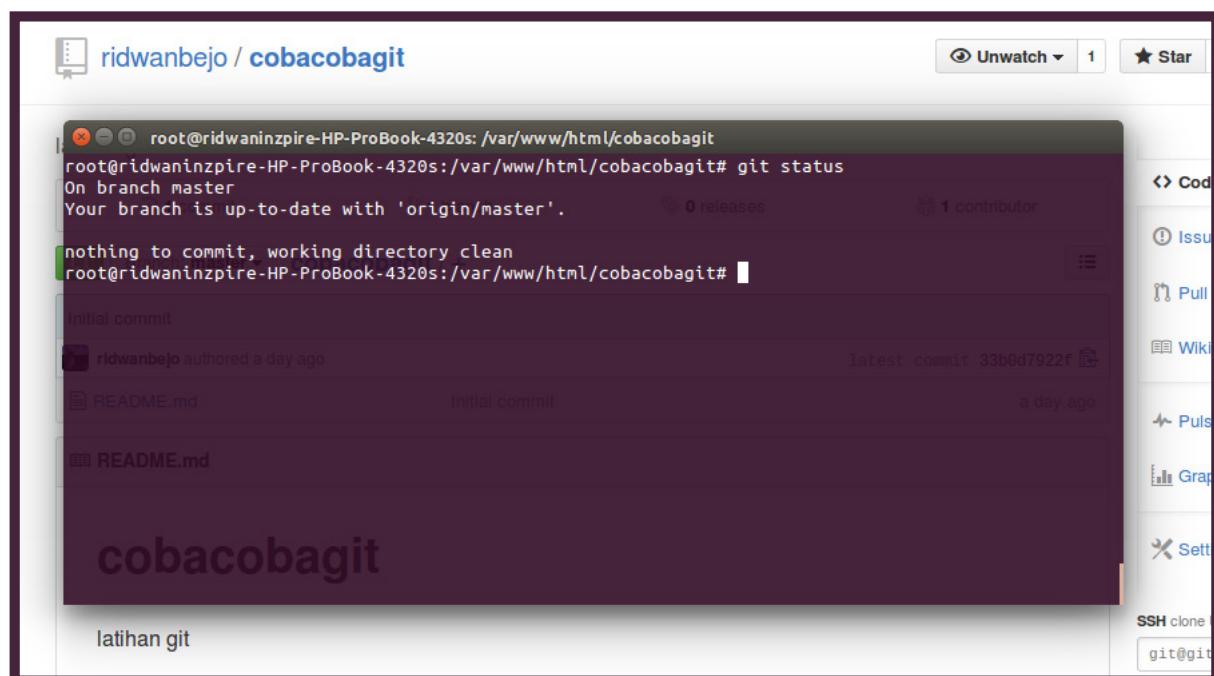
Perintah Dasar Git

Agar menguasai teknik paling dasar dari Git, mari kita coba perhatikan dengan sek-sama simulasi berikutini:

- Sekarang kita akan mulai mengenal beberapa perintah dasar yang dapat Anda gunakan untuk beroperasi dengan Git di Github. Sebagai langkah awal mari kita *clone* proyek yang sudah dibuat sebelumnya dengan perintah **git clone git@github.com:ridwanbejo/cobacobagit.git**, tempatkan di direktori yang Anda inginkan, misalnya di **/var/www/html**. Jika berhasil maka Anda akan melihat informasi seperti pada gambar berikut:



- ▶ Sekarang mari kita masuk ke dalam direktori yang baru saja kita *clone* tersebut lalu eksekusilah perintah **git status** untuk melihat status dari proyek Github yang baru saja di-*clone*. Anda akan melihat informasi seperti pada gambar berikut:



- ▶ Sekarang kita *edit file README.md* yang terdapat dalam proyek tersebut. Simpan hasil *edit*-nya kemudian kita coba perintah **git diff** untuk melihat perubahan

yang terjadi terhadap proyek tersebut. Anda akan melihat perubahan yang terjadi dalam **README.md** tersebut. Jika Anda melihat teks berwarna merah maka baris tersebut telah dihapus, sedangkan jika melihat baris warna hijau maka baris tersebut sudah ditambahkan.

The terminal window shows the command `git diff` output:

```

root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit
git diff --git a/README.md b/README.md
old mode 100644 < 1 commit
new mode 100755
index 1fid031..ddfe41
--- a/README.md      ~~~~~ master
+++ b/README.md      ++++++
@@ -1,2 +1,5 @@
 # cobacobagit
 # latihan git
+Latihan Git untuk mengenal Git bagi pemula
+Salam Nyankod ;)

```

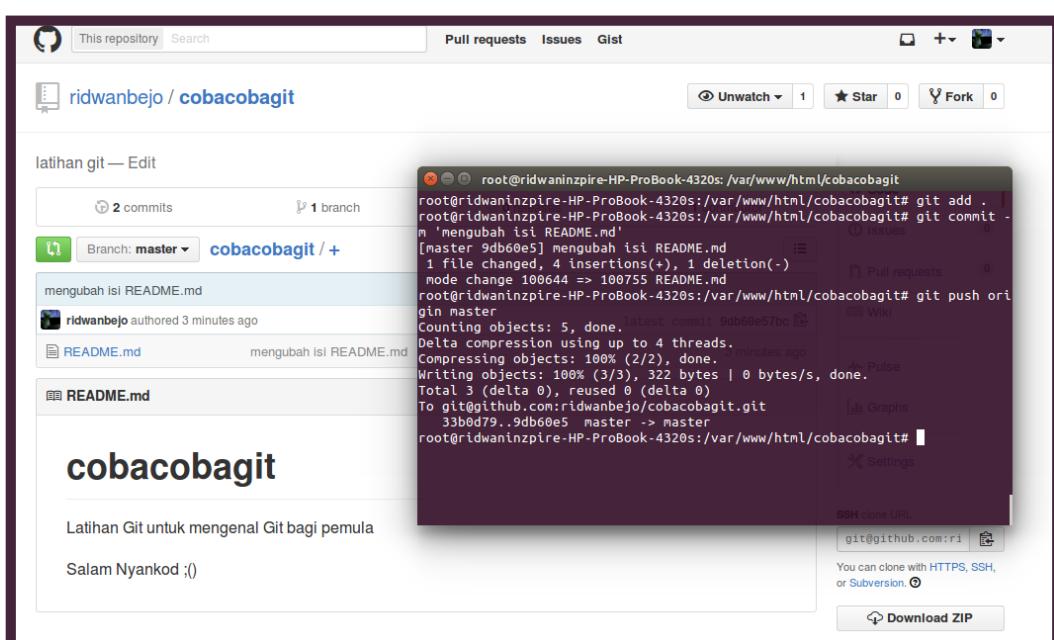
The text editor window shows the README.md file with the following content:

```

# cobacobagit
Latihan Git untuk mengenal Git bagi pemula
Salam Nyankod ;)

```

- ▶ Setelah membuat perubahan pada *file* tersebut, kita akan kirimkan perubahan ke Github. Eksekusi perintah **git add .** untuk memastikan perubahan yang akan dikirim lalu melakukan persetujuan yang diubah dengan perintah **git commit -m "mengubah isi README.md"**, kemudian kirim perubahan tersebut dengan perintah **git push origin master**. Dan coba lihat halaman proyek Anda di Github, dan lihat bagian *commit*. Anda akan melihat perubahan yang telah Anda kirimkan sebelumnya.

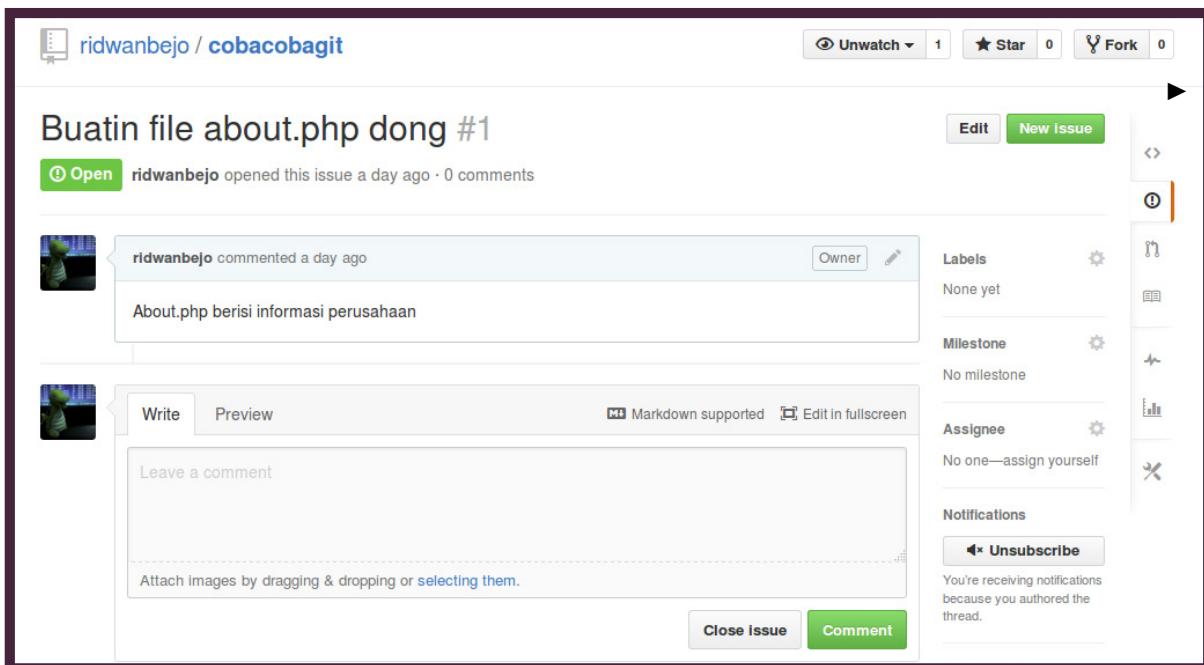


- ▶ Selamat :D, Anda sudah mengetahui teknik paling dasar menggunakan Git. Sekarang kita coba lagi membuat sebuah *file* baru yaitu **index.php** kemudian isi dengan kalimat Hello World :D. Lakukan hal yang sama seperti langkah - langkah diatas, maka *file* baru tersebut akan terkirim ke Github.

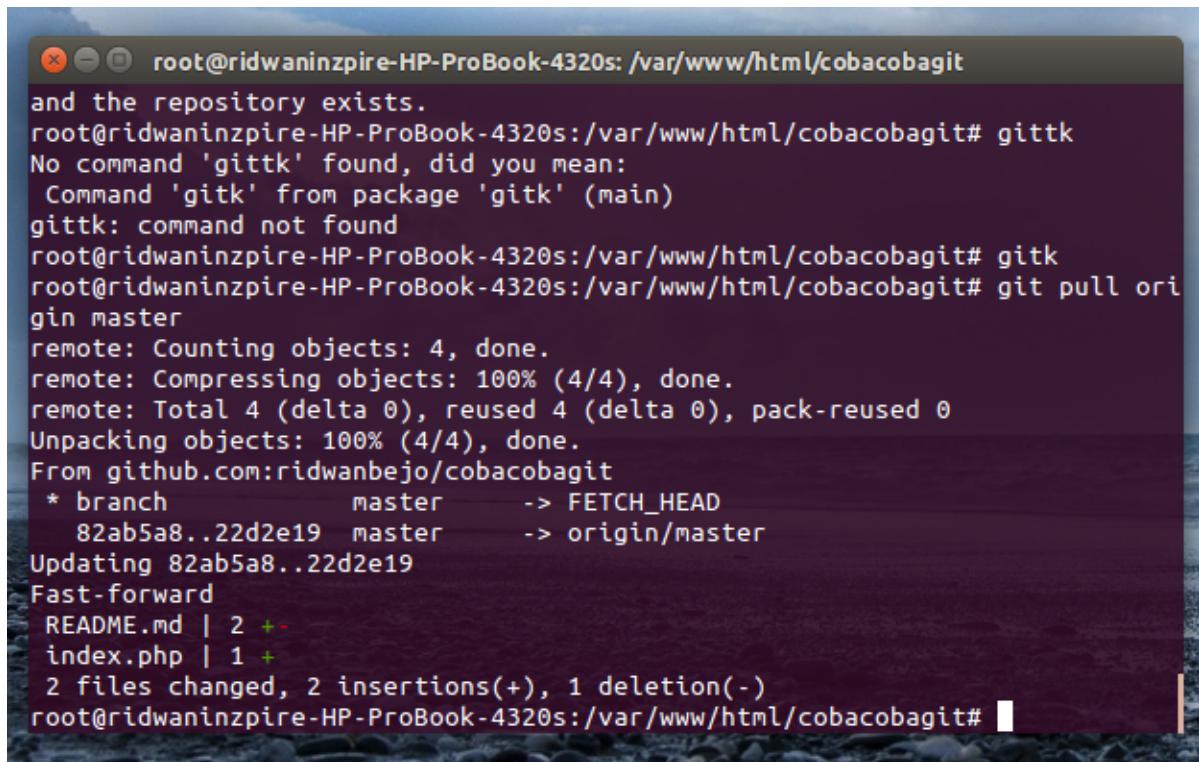
```

root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit
root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit# git commit -m 'menambahkan file index.php'
[master 92ab5a8] menambahkan file index.php
 1 file changed, 5 insertions(+)
 create mode 100644 index.php
root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit# git push origin master
Branch master -> cobacobagit/+Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:idwanbejo/cobacobagit.git
 9db60e5..82ab5a8 master -> master [mengubah isi README.md]
root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit# git status
On branch master
  README.md
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit# git diff
root@ridwaninzipre-HP-ProBook-4320s:/var/www/html/cobacobagit#

```



- ▶ Apabila terjadi perubahan dari kontributor, Anda dapat mengeksekusi perintah **git pull origin master** untuk mendapatkan perubahan. Tentunya kalau perubahan tersebut sudah disetujui oleh pengelola proyek.



The screenshot shows a terminal window with the following command history:

```
root@ridwaninzpire-HP-ProBook-4320s:/var/www/html/cobacobagit
and the repository exists.
root@ridwaninzpire-HP-ProBook-4320s:/var/www/html/cobacobagit# gittk
No command 'gittk' found, did you mean:
  Command 'gitk' from package 'gitk' (main)
gittk: command not found
root@ridwaninzpire-HP-ProBook-4320s:/var/www/html/cobacobagit# gitk
root@ridwaninzpire-HP-ProBook-4320s:/var/www/html/cobacobagit# git pull origin master
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From github.com:ridwanbejo/cobacobagit
 * branch           master      -> FETCH_HEAD
   82ab5a8..22d2e19  master      -> origin/master
Updating 82ab5a8..22d2e19
Fast-forward
 README.md | 2 ++
 index.php | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
root@ridwaninzpire-HP-ProBook-4320s:/var/www/html/cobacobagit#
```

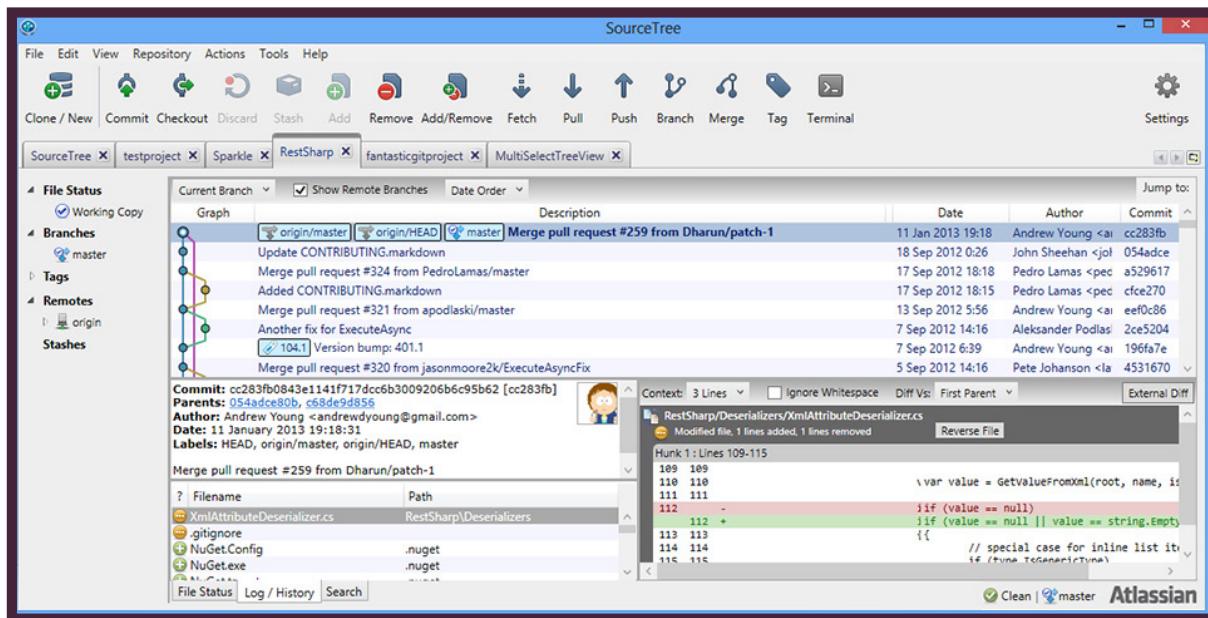
- ▶ Github pun memiliki *issue tracking*, yaitu berupa sistem untuk pengajuan tiket. Anda dapat mengajukan tiket untuk Anda sendiri, atau kepada pemilik sebuah proyek di Github.

Salah satu alasan sederhana kenapa menggunakan layanan Git seperti Github ataupun yang lainnya, Anda tidak perlu khawatir kehilangan *source code* tugas besar dari kuliah pemrograman Anda atau kerjaan Anda. Karena ketika *laptop* atau *komputer* Anda rusak, Anda sudah mempunyai *back up* yang tersimpan di Layanan Git. Selanjutnya, Anda hanya tinggal melakukan *clone* ulang sesudah terjadi insiden.

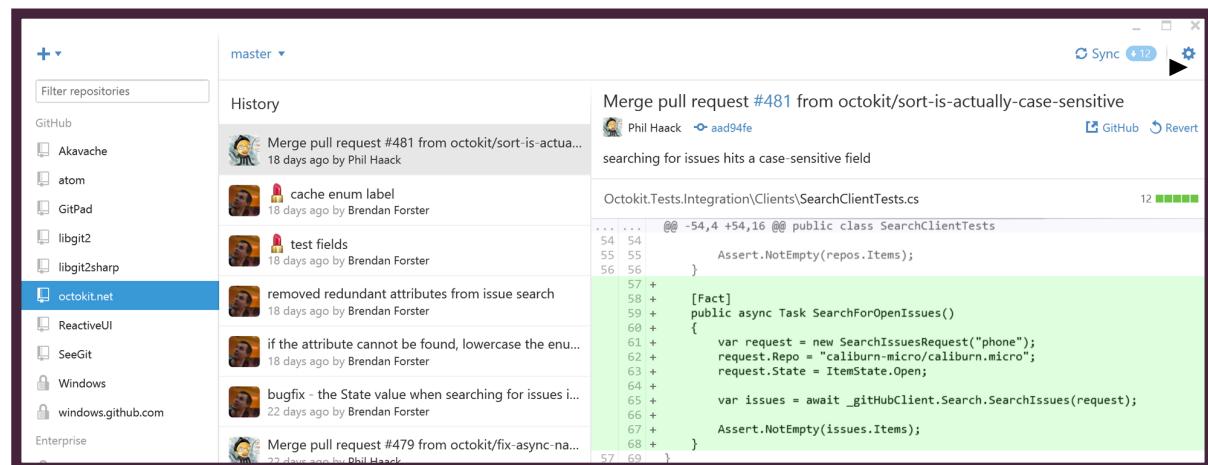
Aplikasi Pendukung Git

Selain menggunakan Git versi *command line* yang sebenarnya sudah lengkap akan fitur - fiturnya. Anda juga dapat memilih beberapa aplikasi *desktop* untuk Git sesuai dengan selera Anda. Berikut ini adalah beberapa aplikasi *deskstop* untuk Git yang dapat Anda pasang di mesin Anda.

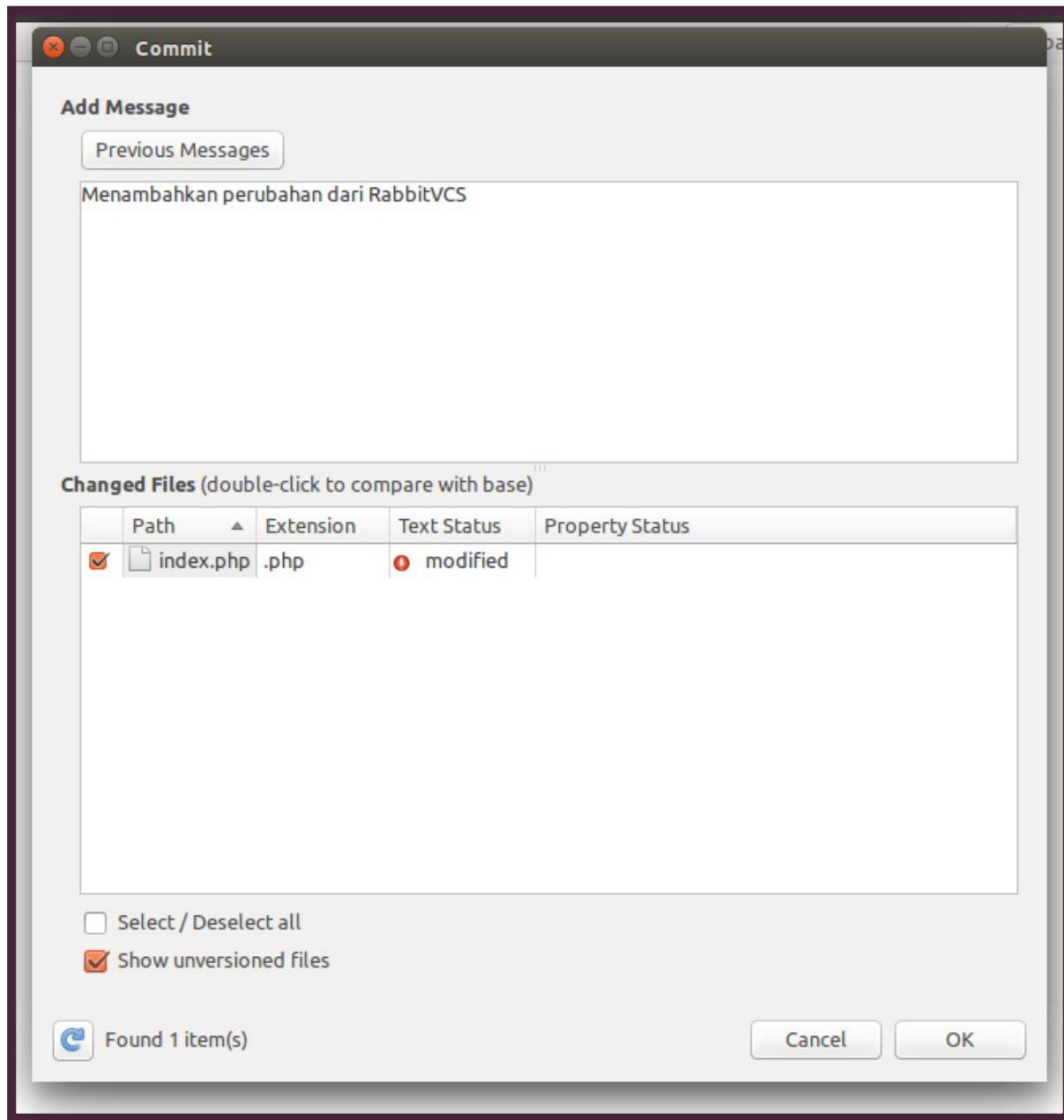
- SourceTree, ditujukan untuk pengguna Windows dan OSX, SourceTree merupakan aplikasi yang dapat mengelola proyek yang dipantau oleh Git atau Mercurial. Aplikasi ini dirilis oleh Atlassian untuk memudahkan penggunaan BitBucket.



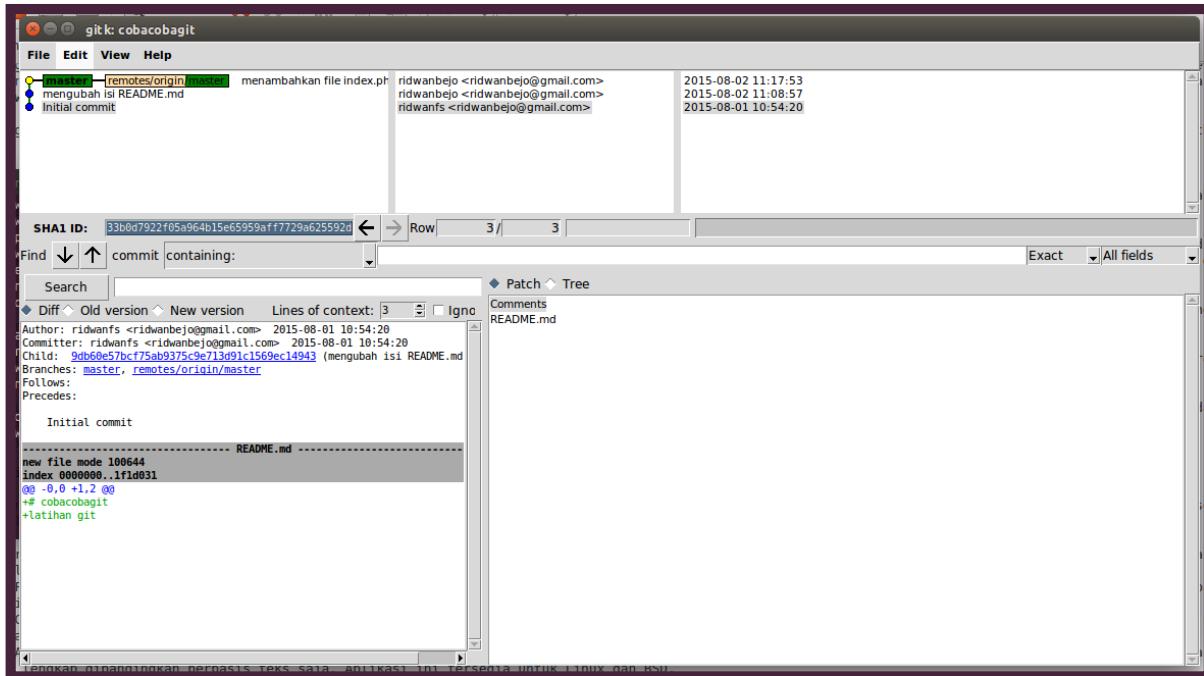
- Github For Windows, ditujukan hanya untuk pengguna Windows, Github For Windows merupakan aplikasi yang dapat mengelola proyek yang dipantau oleh Git. Anda dapat mengurangi penggunaan perintah Git *command line* hanya dengan menekan tombol pada Github For Windows. Aplikasi ini dirilis oleh Github



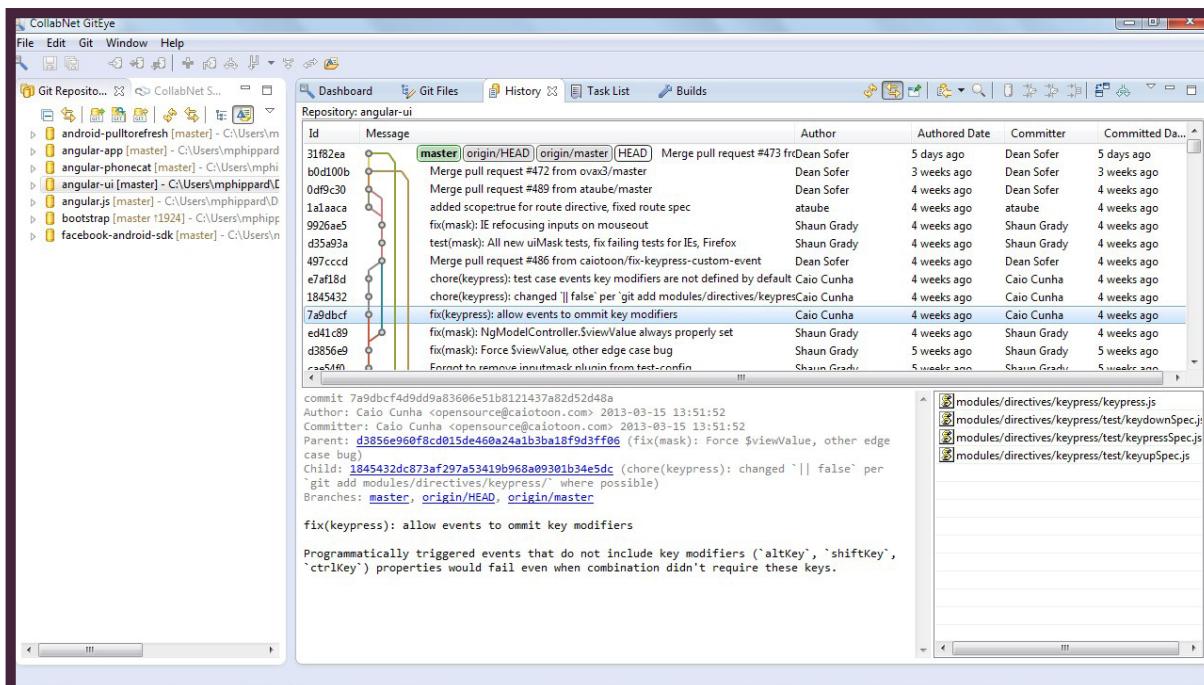
- RabbitVCS, ditujukan untuk pengguna Linux dan BSD. RabbitVCS merupakan aplikasi *deskstop* yang dapat diintegrasikan dengan *file manager* untuk Linux dan BSD seperti Thunar dan Nautilus. Tidak hanya Git, RabbitVCS dapat digunakan juga untuk proyek yang dipantau menggunakan SVN dan Mercurial.



- GitTK, Aplikasi sederhana ini digunakan untuk melihat perubahan yang terjadi dalam sebuah proyek Git. Digunakan untuk memudahkan Anda melihat perubahan secara visual dan lebih lengkap dibandingkan berbasis teks saja. Aplikasi ini tersedia untuk Linux dan BSD.



- GitEye, Aplikasi *desktop* berbasis Eclipse ini dirilis oleh CollabNet. Anda dapat mengelola proyek yang dipantau Git menggunakan GitEye. Aplikasi ini tersedia untuk Windows, Linux, BSD, dan OSX.



Referensi

1. Git Book (<https://git-scm.com/book/en/v2/>)
2. Help Github (<https://help.github.com/>)
3. Bitbucket Documentation (<https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+Documentation+Home>)
4. Try Github (<https://try.github.io/levels/1/challenges/1>)
5. Free Course Git on CodeSchool (<https://www.codeschool.com/courses/try-git>)

Hal-Hal Mendasar yang Harus Dihindari Web Programmer

- Yana Permana -

Mengembangkan sebuah aplikasi web adalah tugas seorang web programmer. Didalamnya banyak komponen-komponen yang harus dikembangkan. Belum lagi aplikasi web harus dikerjakan dengan waktu deadline yang mepet.

Kebanyakan web programmer menginginkan aplikasi web yang dikembangkan mempunyai proses yang cepat dan andal. Namun dalam beberapa hal, proses yang cepat tersebut bisa menjadi senjata makan tuan. Karena tidak mempertimbangkan keamanan di setiap komponen-komponen web seperti validasi input, enkripsi, database, otentifikasi pengguna dan masih banyak lagi.

Mengembangkan aplikasi web dengan proses yang cepat memang penting namun yang lebih penting lagi adalah menghindari kesalahan koding terutama yang menyangkut keamanan dan data sensitif didalam web tersebut. Berikut hal-hal mendasar yang harus web programmer hindari:



Penggunaan Password Hard-Coded

Penggunaan password hard-coded masih menjadi kesalahan yang sering dilakukan oleh banyak programmer. Peletakkan password yang biasanya diletakkan pada koding atau kolom komentar ternyata bisa menjadi hal yang berbahaya. Walaupun biasanya password tersebut hanya digunakan ketika pengujian suatu fungsi yang diinginkan.

```
1  <form name="login" action="login.php" method="post"
2    onSubmit="return email_Submit();">
3      <b>Log in Address: </b>
4      <input type="text" name="email" size="29" value="">
5      <b>Password: </b>
6      <input type="password" name="password" size="29" value="">
7
8
9    <!-- Debugging credentials: admin@bankminion.com:adminGanteng -->
10
11  </form>
```

Kelihatannya sepele memang, namun jika programmernya lupa untuk menghapus komentar tersebut, bisa-bisa game over itu hehe. Apakah sudah ada yang sistem yang menjadi korban? Banyak diantaranya Bank [1], CCTV [2] dan lain-lain.

Tidak Memperhatikan Otentifikasi Pengguna



Two Factor Authentication

Ini terjadi ketika, programmer tidak memperhatikan fitur mana yang harus dibolehkan oleh pengguna biasa dan pengguna admin. Kadang otentifikasi yang diberikan programmer biasanya sama untuk kedua pengguna. Hal ini sebenarnya

merepotkan, karena apabila attacker tahu otentifikasi yang diberikan sama untuk keduanya. Attacker dengan mudah menembus otentifikasi tersebut.

Perlu adanya perlakuan yang berbeda untuk otentifikasi pengguna admin. Kadang yang sering terjadi adalah ketika seseorang menjadi pengguna admin di sebuah perusahaan, kemudian dia keluar. Setelah keluar ternyata dia masih bisa mengakses sistem tersebut, hal ini juga berbahaya.

Solusinya yang mungkin untuk ini adalah penggunaan two-factor otentifikasi. Biasanya menggunakan faktor geografis, data karyawan yang berhak mengakses, nomor HP, email, dan lain-lain. Semua bisa diatur sesuai kebutuhan.

Validasi Input

Input yang tidak tervalidasi bisa merepotkan aplikasi web yang hendak dibangun. Attacker biasanya melakukan *fuzzing*, untuk mencoba seberapa jauh aplikasi web tersebut merespon berbagai isian input yang biasanya berisi *shellcode* dan injeksi seperti SQL, XSS, Xpath, dan lain-lain.



Solusi yang mungkin untuk ini adalah penggunaan RegEx. RegEx merupakan kependekan dari Regular Expression. Penggunaannya memang tidak mudah, tapi kalau sudah terbiasa dengan RegEx, hal tersebut akan menguntungkan kita.

Penggunaan Enkripsi Lemah



DES dan lain-lain [3].

Ketika mengembangkan aplikasi web, programmer yang kurang memahami kriptografi biasanya menggunakan enkripsi lemah. Enkripsi tersebut sudah dicap lemah oleh para komunitas kriptografer dunia. Diantara enkripsi lemah yang sering digunakan adalah MD5, RC4,

Mengijinkan Pengguna Mendaftar Dengan Password Lemah

Pengguna suka dengan password yang mudah diingat dan pendek. Namun jangan biarkan hal itu terjadi, walaupun kita tidak punya tanggung jawab untuk itu. Lebih baiknya, beri syarat password yang ketat.

Jika password lemah, attacker dengan mudah melakukan bruteforce dengan wordlist yang tidak terlalu besar. Apa yang terjadi jika attacker sudah mendapatkan password tersebut? Hal selanjutnya adalah mereka akan melakukan fuzzing pada setiap input ketika mereka sudah login ke sistem tersebut. Ini akan berbahaya jika poin-poin sebelumnya tidak diperhatikan.



Proses pengembangan aplikasi web dengan proses cepat dan andal tidaklah jadi masalah ketika keamanan juga diperhitungkan. Walaupun sepele, namun jangan diremehkan. Karena human weakness adalah penyebab terbesar tembusnya keamanan sebuah aplikasi web.

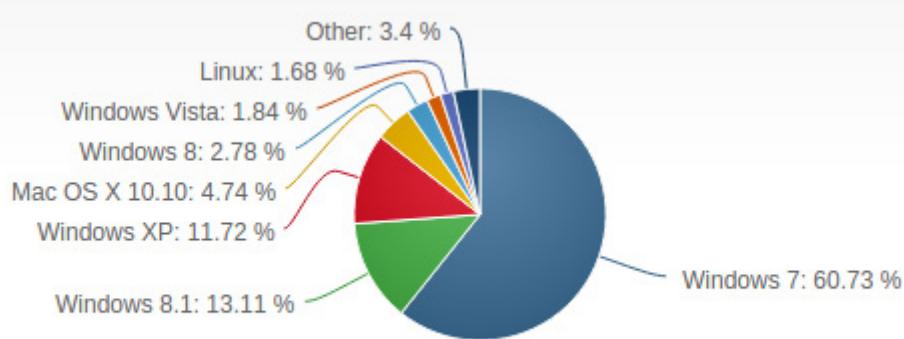
- [1] <http://o.canada.com/news/bmo-atm-winnipeg-460526>
- [2] <https://media.blackhat.com/us-13/US-13-Heffner-Exploiting-Network-Surveillance-Cameras-Like-A-Hollywood-Hacker-WP.pdf>
- [3] <https://gist.github.com/tqbf/be58d2d39690c3b366ad>

UBUNTU MAKE, Sahabat Para Developer Pengguna UBUNTU

- Bagus Aji Santoso -

Komputer desktop yang menggunakan sistem operasi Linux khususnya Ubuntu memang jauh lebih sedikit dibandingkan sistem operasi seperti Microsoft Windows maupun Mac OS X. Meskipun begitu, tak sedikit pengguna Ubuntu yang juga memiliki pekerjaan sebagai seorang developer. Bagi para developer aplikasi web, developer aplikasi desktop, maupun developer-developer lainnya, Ubuntu seringkali menjadi pilihan utama sebagai sistem operasi untuk membantu mereka menyelesaikan pekerjaannya.

Illustration 1: Sumber: itsfoss.com



Ada berbagai alasan mengapa para developer ini memilih Ubuntu dibandingkan sistem operasi lain. Diantaranya adalah Ubuntu dapat digunakan secara gratis tanpa harus membayar sepeser pun, Ubuntu bebas untuk dimodifikasi sesuai dengan kebutuhan kita, dan Ubuntu dapat menjadi lingkungan pengembangan aplikasi yang menyenangkan.

Hampir tak ada bahasa pemrograman yang tidak dapat berjalan di Ubuntu. Baik itu C/C++, Java, Python, Perl, PHP, bahkan programmer .NET Framework dapat bekerja di Ubuntu dengan menggunakan [Mono](#). Canonical, perusahaan pengembang Ubuntu sadar bahwa sebagian besar pengguna mereka merupakan seorang developer. Oleh karena itu mereka mengembangkan satu aplikasi yang akan membantu pengguna Ubuntu untuk mengunduh *developer tools* terbaru beserta dependensinya dengan jauh lebih mudah. Aplikasi ini diberi nama Ubuntu Make.

Ubuntu Make, yang dulunya bernama [Ubuntu Developer Tools Center](#) memiliki beberapa prinsip dasar dalam pengembangannya. Prinsip-prinsip tersebut:

1. Ubuntu Make akan selalu mengunduh dan mendukung versi terbaru dari bahasa pemrograman yang bersangkutan.
2. Ubuntu Make akan mendukung Ubuntu versi LTS (Long Term Support). Artinya Ubuntu Make di Ubuntu 14.04 akan mendapat perbaruan terus menerus hingga 2019.
3. Karena Ubuntu Make akan selalu mengunduh versi terbaru dari bahasa pemrograman yang mereka dukung, maka Ubuntu Make akan selalu menjalankan *testing* secara berkesinambungan sehingga bahasa pemrograman yang didukung oleh Ubuntu Make akan dapat bekerja dengan baik

Ubuntu Make tidak hanya dapat digunakan di Ubuntu, namun distribusi Linux lain yang merupakan turunan dari Ubuntu juga dapat

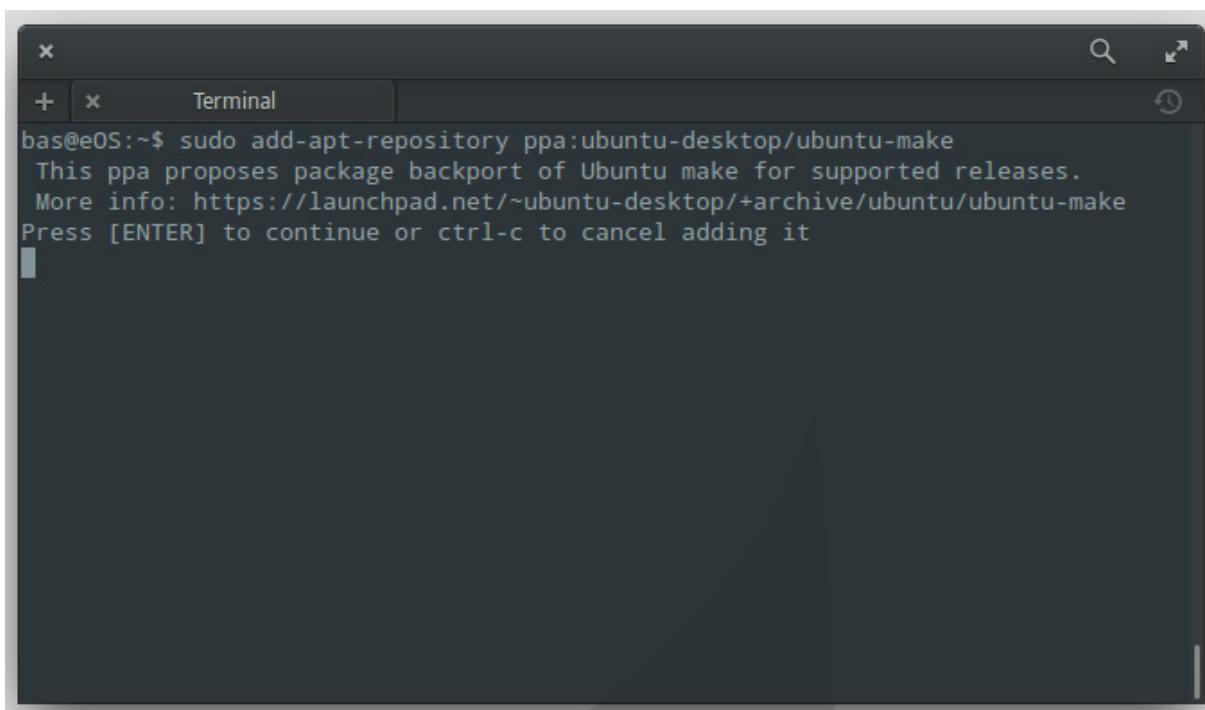
ikut menggunakannya. Memasang Ubuntu Make pun tidak lah sulit. Berbekal pengetahuan dasar akan penggunaan aplikasi Terminal, kita sudah dapat memasang dan menggunakan Ubuntu Make.

Untuk memasang Ubuntu Make, pastikan pembaca menggunakan sistem operasi Linux dengan distribusi Ubuntu atau turunannya (Linux Mint, Elementary, Zorin, Netrunner, dll.). Selain itu, pastikan bahwa versi yang digunakan berbasis Ubuntu 14.04 atau yang lebih baru.

Pertama, buka aplikasi terminal dan jalankan perintah di bawah ini untuk menambahkan *repository* Ubuntu Make. Masukkan password apabila diminta:

```
$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
```

Illustration

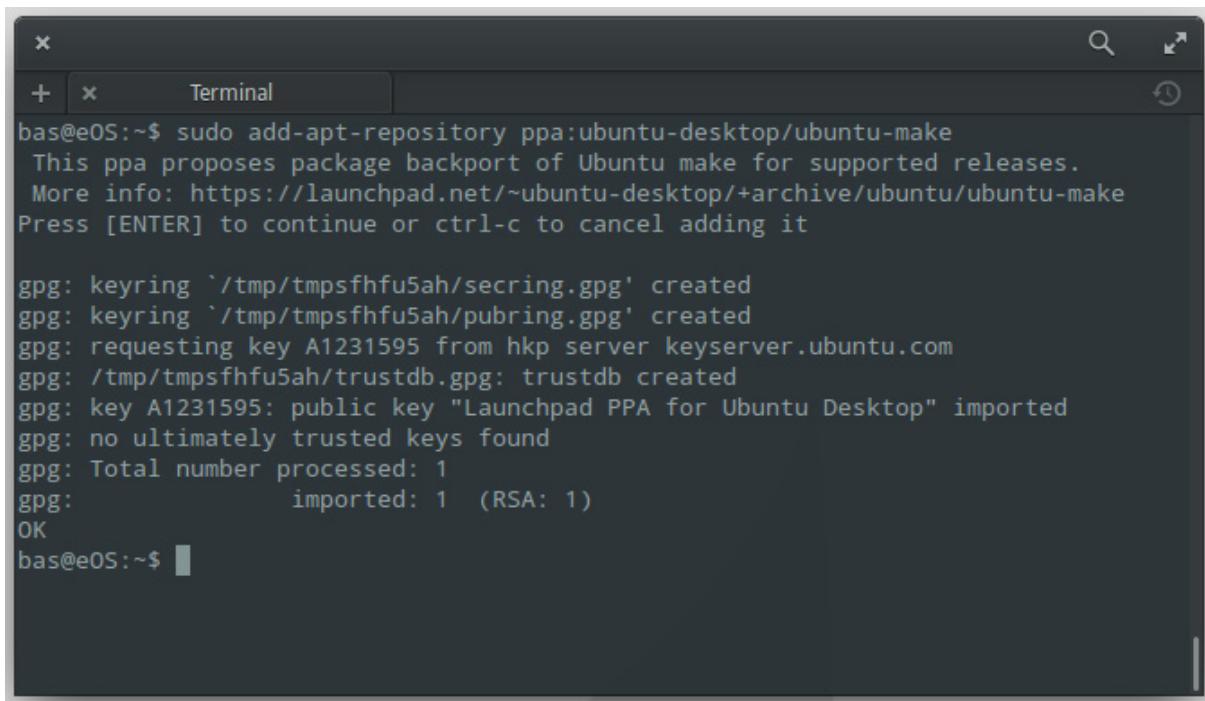
A screenshot of a terminal window titled "Terminal". The window contains the following text:

```
bas@eOS:~$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
This ppa proposes package backport of Ubuntu make for supported releases.
More info: https://launchpad.net/~ubuntu-desktop/+archive/ubuntu/ubuntu-make
Press [ENTER] to continue or ctrl-c to cancel adding it
```

The terminal window has a dark background and light-colored text. It includes standard terminal icons at the top: a close button, a plus sign for new tabs, and a search icon.

Jika aplikasi terminal pembaca sudah mengeksekusi perintah di atas, tekan tombol ENTER untuk melanjutkan.

Illustration



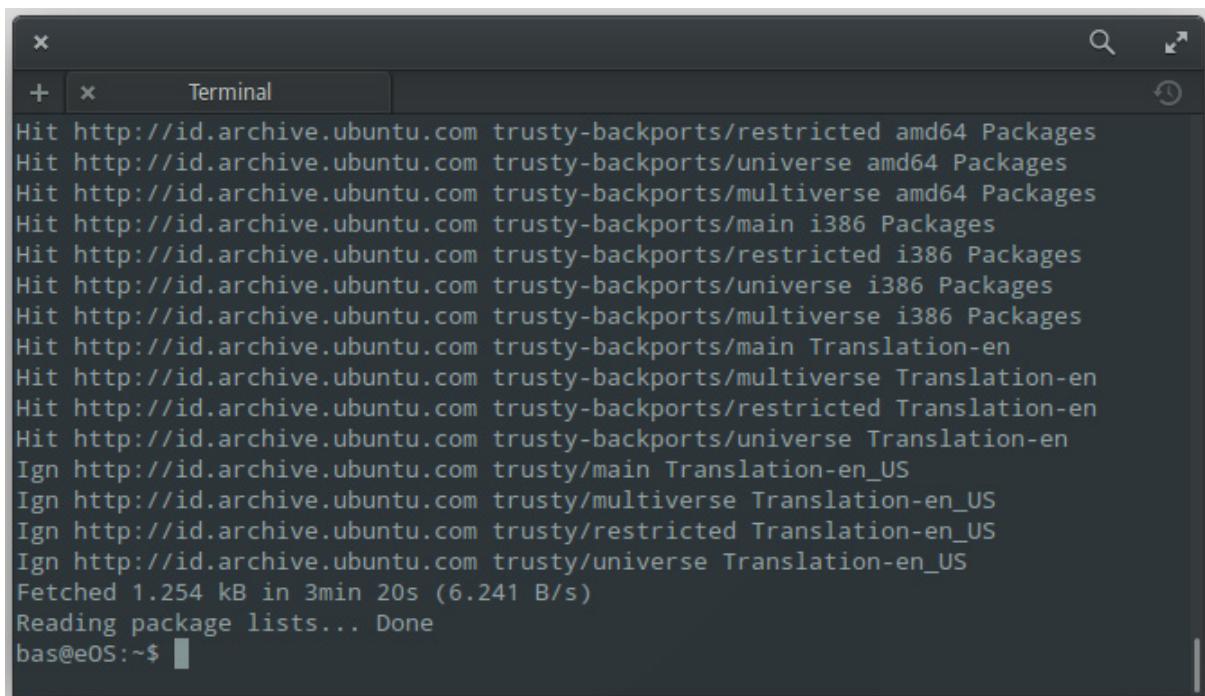
```
bas@eOS:~$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
This ppa proposes package backport of Ubuntu make for supported releases.
More info: https://launchpad.net/~ubuntu-desktop/+archive/ubuntu/ubuntu-make
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring '/tmp/tmpsfhfu5ah/secring.gpg' created
gpg: keyring '/tmp/tmpsfhfu5ah/pubring.gpg' created
gpg: requesting key A1231595 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpsfhfu5ah/trustdb.gpg: trustdb created
gpg: key A1231595: public key "Launchpad PPA for Ubuntu Desktop" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:          imported: 1  (RSA: 1)
OK
bas@eOS:~$
```

Setelah *repository* Ubuntu ditambahkan, sekarang saatnya untuk memperbarui daftar paket aplikasi Ubuntu dengan perintah:

```
$ sudo apt-get update
```

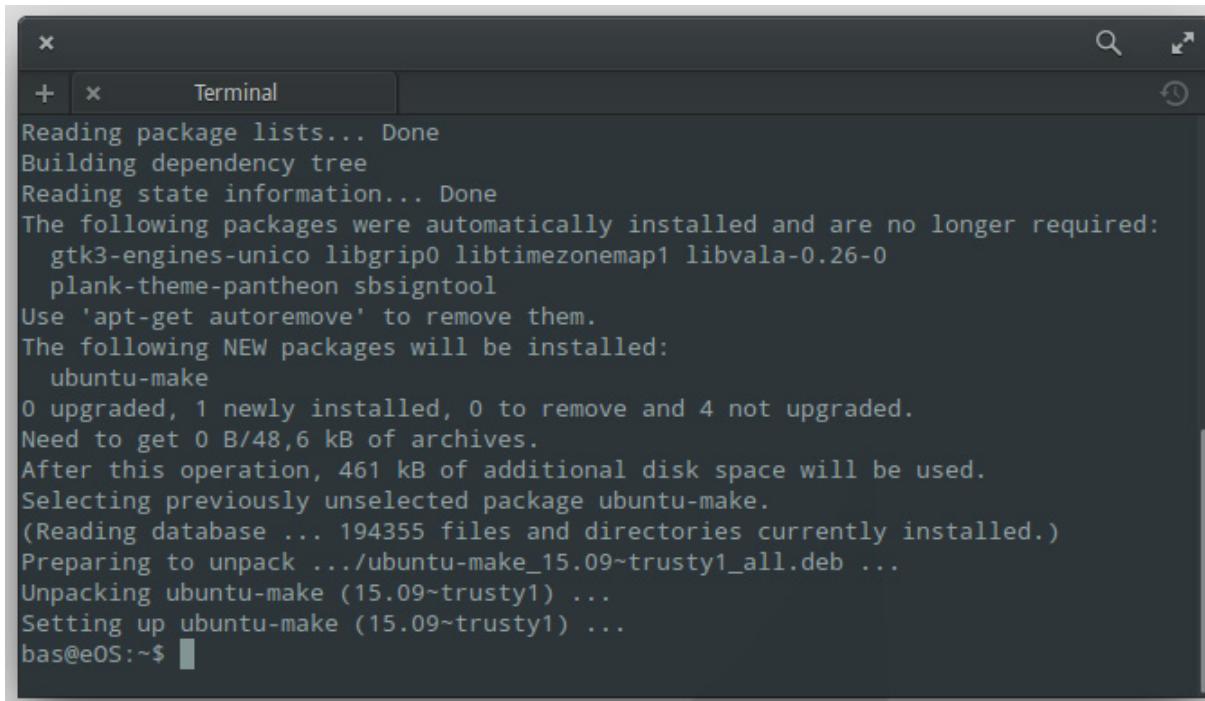
Illustration



```
Hit http://id.archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://id.archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://id.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://id.archive.ubuntu.com trusty-backports/main i386 Packages
Hit http://id.archive.ubuntu.com trusty-backports/restricted i386 Packages
Hit http://id.archive.ubuntu.com trusty-backports/universe i386 Packages
Hit http://id.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Hit http://id.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://id.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://id.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://id.archive.ubuntu.com trusty-backports/universe Translation-en
Ign http://id.archive.ubuntu.com trusty/main Translation-en_US
Ign http://id.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://id.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://id.archive.ubuntu.com trusty/universe Translation-en_US
Fetched 1.254 kB in 3min 20s (6.241 B/s)
Reading package lists... Done
bas@eOS:~$
```

Setelah *repository* diperbarui, sekarang saatnya untuk memasang Ubuntu Make.

```
$ sudo apt-get install ubuntu-make
```



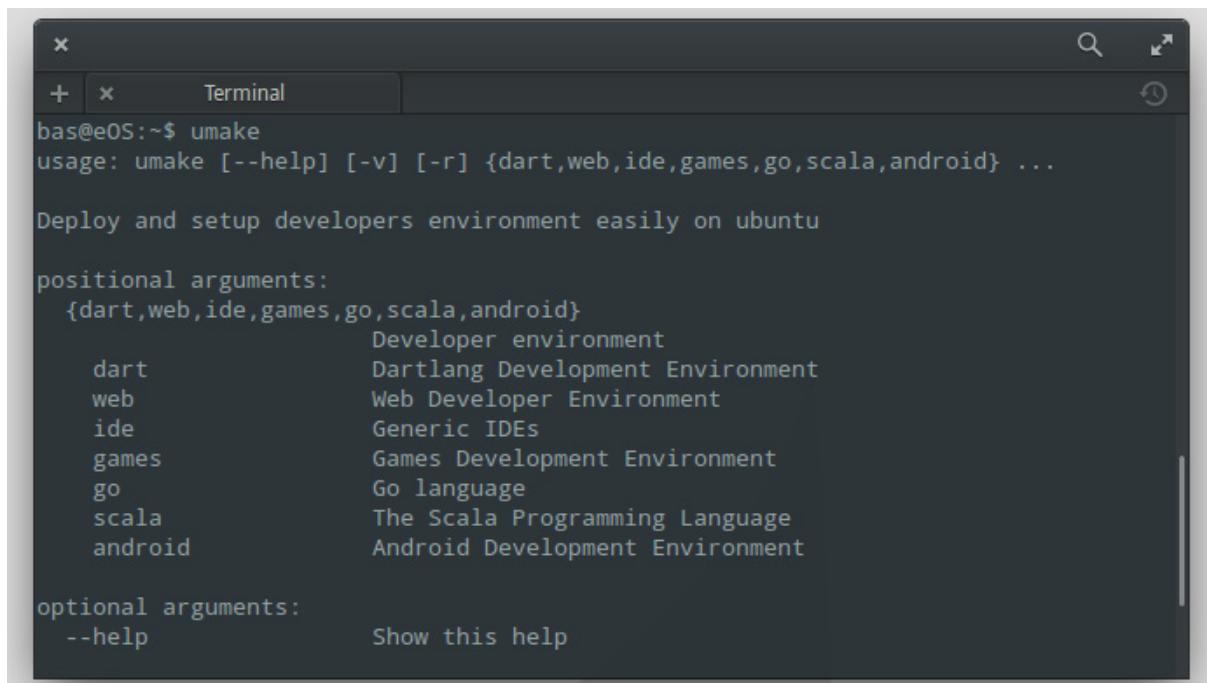
The screenshot shows a terminal window with the title 'Terminal'. The window contains the following text output from the apt-get command:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gtk3-engines-unico libgrip0 libtimezonemap1 libvala-0.26-0
  plank-theme-pantheon sbsigntool
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  ubuntu-make
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 0 B/48,6 kB of archives.
After this operation, 461 kB of additional disk space will be used.
Selecting previously unselected package ubuntu-make.
(Reading database ... 194355 files and directories currently installed.)
Preparing to unpack .../ubuntu-make_15.09~trusty1_all.deb ...
Unpacking ubuntu-make (15.09~trusty1) ...
Setting up ubuntu-make (15.09~trusty1) ...
bas@eOS:~$
```

Apabila telah terpasang, Ubuntu Make dapat dijalankan dengan perintah ‘umake’ melalui Terminal. Saat tulisan ini dibuat, Ubuntu Make telah memiliki tujuh kategori *developer environment* yang dapat dipasang. Ketujuh kategori tersebut adalah:

1. Dart
2. Web
3. IDE
4. Games
5. Go
6. Scala
7. Android

Diantara ketujuh kategori di atas, beberapa diantaranya memiliki pilihan tambahan. Sebagai contoh kategori *games* memiliki dua pilihan yakni stencyl dan unity3d. Sedangkan, kategori IDE memiliki pilihan yang lebih banyak seperti eclipse, pycharm, arduino, webstorm, rubymine, dsb.



```
bas@eOS:~$ umake
usage: umake [--help] [-v] [-r] {dart,web,ide,games,go,scala,android} ...

Deploy and setup developers environment easily on ubuntu

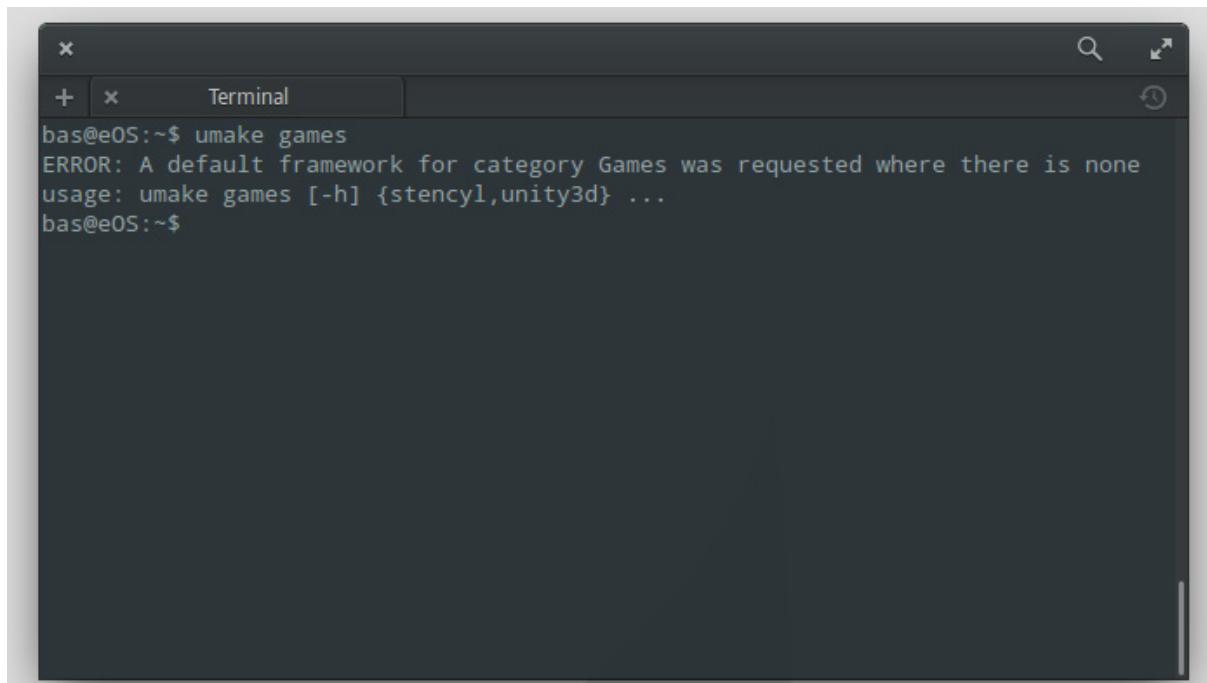
positional arguments:
  {dart,web,ide,games,go,scala,android}
                        Developer environment
  dart                  Dartlang Development Environment
  web                   Web Developer Environment
  ide                   Generic IDEs
  games                 Games Development Environment
  go                    Go language
  scala                 The Scala Programming Language
  android               Android Development Environment

optional arguments:
  --help                Show this help
```

Sekarang, mari kita coba memasang unity3d dengan Ubuntu Make. Langkah pertama, pastikan aplikasi Terminal telah terbuka dan jalankan perintah berikut:

```
$ umake games
```

Aplikasi terminal pembaca seharusnya akan menampilkan keluaran seperti gambar di bawah.



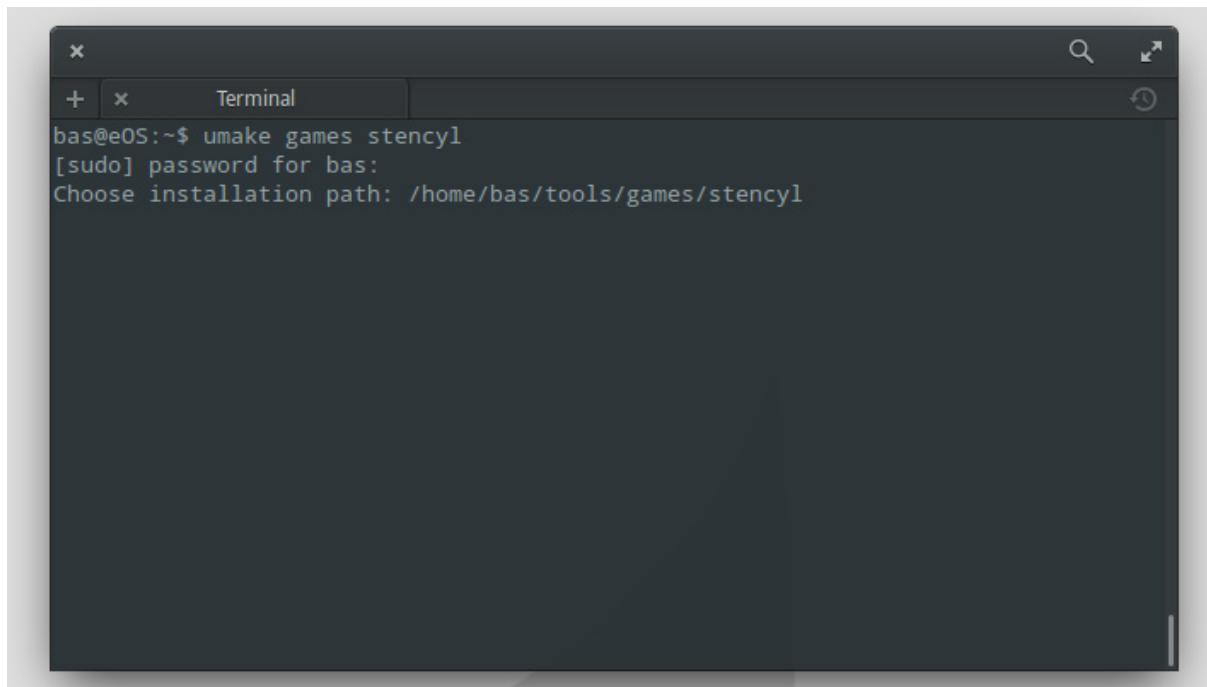
```
bas@eOS:~$ umake games
ERROR: A default framework for category Games was requested where there is none
usage: umake games [-h] {stencyl,unity3d} ...
bas@eOS:~$
```

Pesan kesalahan di atas adalah normal apabila kategori yang dipilih merupakan kategori yang memiliki pilihan tambahan. Apabila kategori tersebut tidak memiliki pilihan tambahan seperti Dart, maka pesan kesalahan di atas tidak akan muncul.

Kembali ke terminal, kita lengkapi perintah di atas dengan:

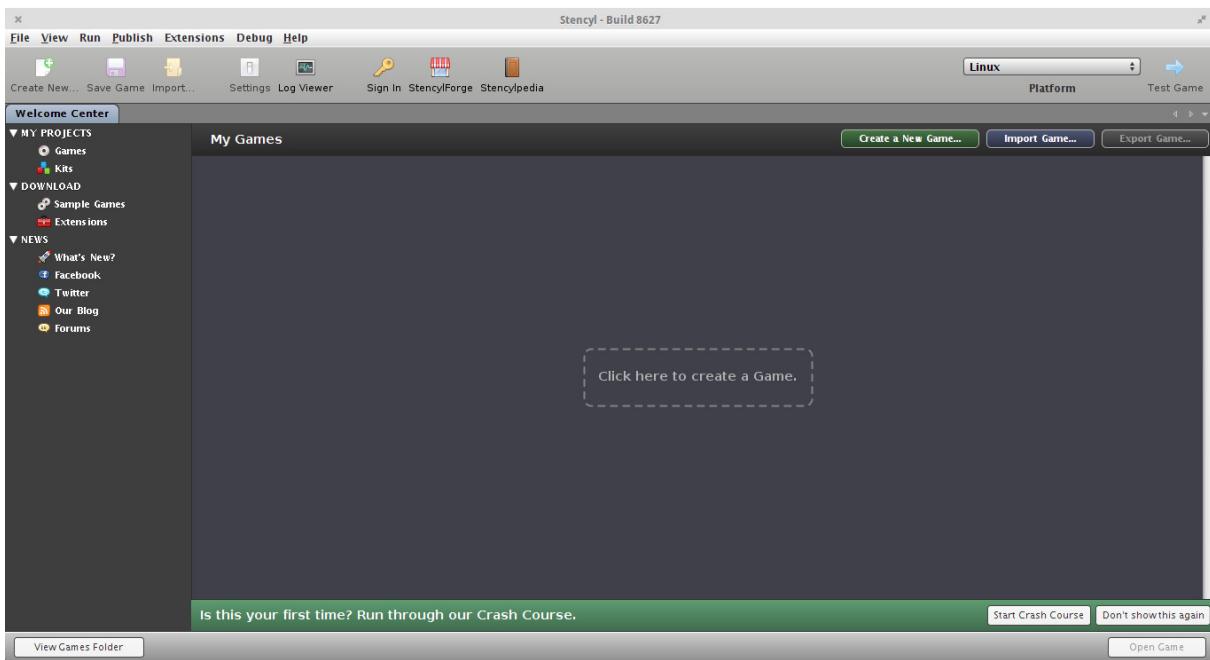
```
$ umake games stencyl
```

Jika perintah benar, maka Ubuntu Make akan meminta kita memilih sendiri direktori (folder) tempat menyimpan aplikasi yang ingin dipasang, pada contoh di atas memasang Stencyl. Jika tidak ingin mengubah pengaturannya, tekan langsung tombol Enter.



Ubuntu Make akan mulai mengunduh aplikasi yang dipilih dan memasangkannya begitu proses unduhan telah selesai. Pesan *installation done* akan menyapa apabila proses pengunduhan dan pemasangan telah berhasil.

```
bas@eOS:~$ umake games stencyl
[sudo] password for bas:
Choose installation path: /home/bas/tools/games/stencyl
Downloading and installing requirements
100% |#####
Installing Stencyl
#####
Installation done
bas@eOS:~$
```



Mudah sekali bukan? Dengan proses yang sama kita bisa memasang berbagai *developer environment* lewat satu aplikasi, Ubuntu Make.



Gambar 1 (belakang kiri) Dave Mellis, Tom Igoe; (depan kiri) Gianluca Martino, David Cuartielles, Massimo Banzi

Memprogram Dunia Fisik Bersama Arduino

- Toni Haryanto -

Hai CoderTalker, siapa yang belum kenal sama Arduino? Mungkinkah hanya kalangan tertentu saja yang sudah akrab dengan nama ini? Atau justru para programmer Indonesia udah ga asing lagi bahkan sudah pernah berjabat tangan? Terutama buat Kamu yang baru denger atau sudah denger tapi belum pernah colek-colek, mestinya tidak melewatkannya mainan yang satu ini. Ya, bagi programmer, spesialis bahasa pemrograman apapun, Arduino mestilah jadi mainan, kalopun tidak bisa jadi partner profesi. So, Arduino itu, orang yang mana, kok jadi bahan permainan segala?

Arduino itu bukan nama orang Bro. Yang dipajang di cover itu adalah para pengembang Arduino. Arduino merupakan single-board microcontroller atau pengendali mikro papan tunggal yang bersifat open-source. Atau dalam bahasa sederhana, Arduino merupakan board atau PCB yang sudah terpasang berbagai komponen mikroelektronika termasuk mikrokontroller yang dapat diprogram dengan mudah, baik software maupun hardwarenya bersifat open-source. di bawah lisensi Creative Commons Attribution Share-Alike, siapapun dapat memodifikasi, memproduksi dan mendistribusikannya dengan bebas.

Bayangan, terutama kalo selama ini Kamu sebagai programmer software dan berkecimpung di dunia digital, semua produk yang pernah Kamu buat adalah produk perangkat lunak yang ‘ga keliatan’ buat sebagian orang. Sekarang, dengan Arduino, kode-kode program Kamu bisa mengemuka dan keliatan efeknya secara fisik. Kode-kode Kamu bisa berinteraksi dengan dunia nyata, menangkap gejala alam seperti cahaya, gerakan, suhu udara, tekanan, serta memberi respon yang nyata pula dalam bentuk gerakan, cahaya, dan lain sebagainya. Bukankah itu hal yang menakjubkan? Lebih jauh lagi, Kamu bisa bikin robot menggunakan Arduino, mulai dari yang sederhana hingga yang ekspert. Kalo selama ini Kamu berpikir remote TV, kalkulator, speedometer motormu, mainan elektronik, joystick PS dan handphone hanya bisa dibuat oleh perusahaan besar, Kamu mesti benerbener nyoba nih teknologi yang sebenarnya sudah ada sejak satu dekade kebelakang ini.

Kilas Balik Arduino

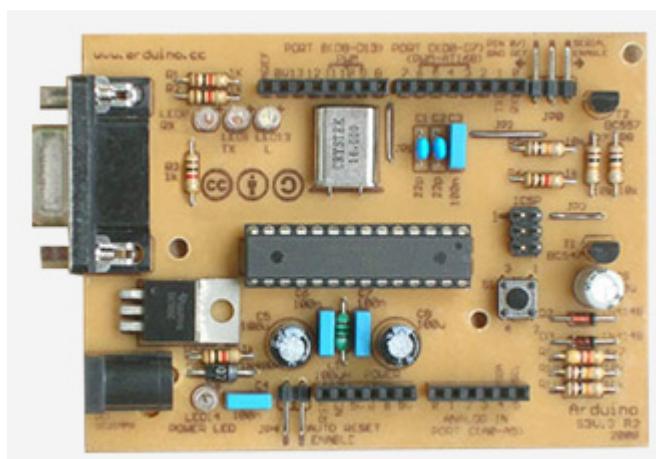
Proyek Arduino pertama kali dibuat di Interaction Design Institute Ivrea di Italia sebagai media proyek bagi mahasiswa. Awalnya mereka pada saat itu menggunakan Basic Stamp, salahsatu produk prototype board berbasis bahasa pemrograman BASIC yang sudah populer sejak tahun 1990an yang dibuat oleh Parallax, sebuah perusahaan manufaktur elektronik

yang berbasis di California. Namun, Massimo Banzi, salah satu founder Arduino, menemukan setidaknya dua kekurangan pada Basic Stamp: spesifikasi papan yang terlalu rendah dan seringkali tidak dapat memenuhi kebutuhan proyek mahasiswanya, dan harganya yang cukup mahal, \$100 untuk setiap board.

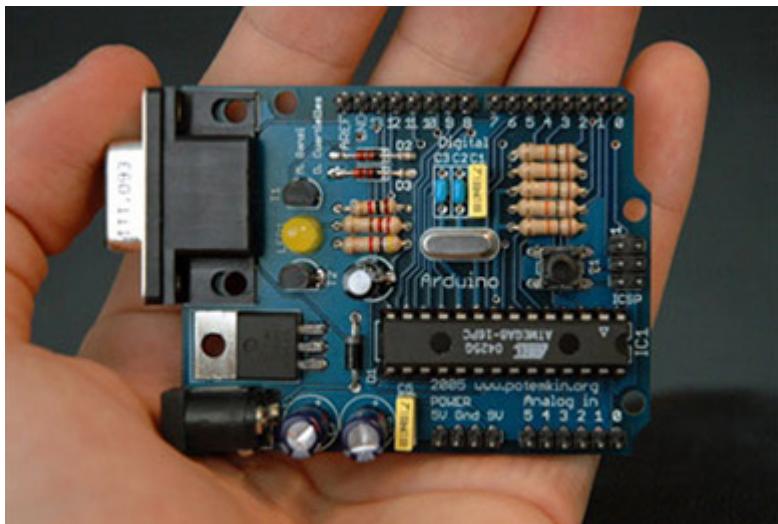
Selain itu Banzi juga tertarik dengan aplikasi buatan koleganya Casey Reas dan Benjamin Fry dari MIT, yang mereka beri nama Processing. Processing adalah bahasa pemrograman sekaligus IDE atau Integrated Development Environment yang ditujukan untuk memudahkan para desainer dalam memvisualisasikan data. Processing berkembang dengan cepat dan menarik perhatian banyak kalangan karena programmer pemula pun bahkan dengan mudah dapat memprogram visualisasi data yang kompleks dan cantik dengan Processing. Banzi membayangkan jika dia juga bisa membuat hal yang serupa untuk memprogram mikrokontroller.

Salah satu mahasiswanya Hernando Barragán, mengawali langkah tersebut. Dia membuat prototype board yang ia beri nama Wiring, berikut dengan IDEnya. Proyek tersebut sangat menjanjikan dan nyatanya masih terus aktif hingga saat ini. Namun Massimo membayangkan sesuatu yang jauh lebih besar dari itu. Dia ingin membuat platform yang jauh lebih sederhana, jauh lebih mudah digunakan dan jauh lebih ekonomis. Singkat kisah lahirnya Arduino dengan inisiasi para founder yakni Massimo Banzi, David

Cuartielles, Tom Igoe, Gianluca Martino, dan David Mellis di sebuah bar bernama Arduin yang juga nama salah satu raja Italia tahun 1002-1014 dan menjadi inspirasi nama Arduino. Dan belakangan ini pada bulan Mei 2015 nama Genuino muncul sebagai merek dagang untuk penggunaan di luar Amerika Serikat.



Gambar 2 Versi prototype Arduino Serial dengan jalur satu sisi
(arduino.cc)



Gambar 3 Versi pertama Arduino, menggunakan koneksi serial (arduino.cc)

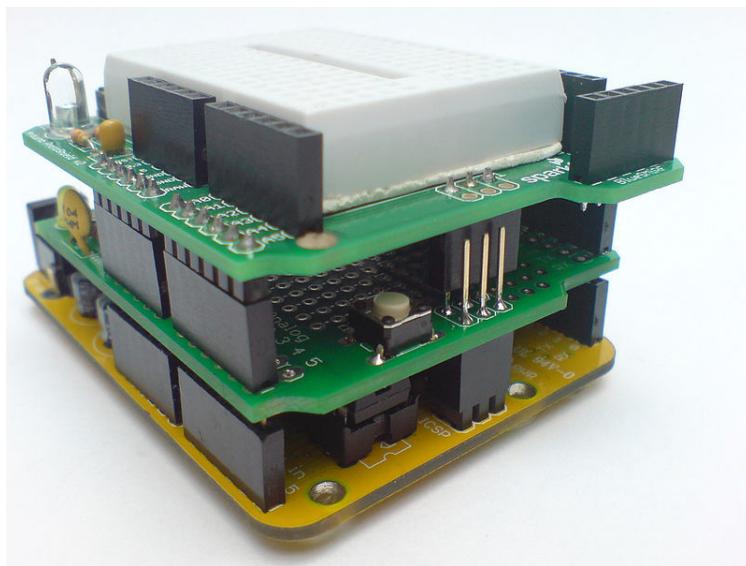
Boleh dibilang Arduino adalah cikal bakal dari trend open source hardware. Umumnya open source dikenakan pada perangkat lunak. Open source membuat sebuah perangkat lunak berkembang dengan

pesat karena dukungan dan kontribusi dari banyak orang dari berbagai tempat. Namun para founder ingin tidak hanya perangkat lunaknya saja yang open source, tapi juga hardwarenya. Maka akhirnya skema papan Arduino pun dipublikasikan di bawah lisensi Creative Commons, sehingga siapapun dapat membuat papan Arduino versi mereka sendiri, membuat varian dan meningkatkan performanya sesuai keperluan. Dan boleh jadi faktor open source ini juga yang menjadi salah satu penyebab Arduino mendunia meski tanpa strategi marketing yang gede-gedean.

Versi pertama dari Arduino dijual dengan nama Arduino itu sendiri. Versi ini menggunakan mikroprosesor ATMega8 dan masih menggunakan port serial. Versi ini dijual di halaman web dengan URL potemkin.org. Selain itu pada versi pertama ini belum ada LED default yang terhubung dengan pin 13. Pengembangan selanjutnya dari versi ini sudah menggunakan port USB dan dipublikasikan di halaman web dengan URL arduino.berlios.de. Versi selanjutnya diberi nama Arduino Diecimila dan sudah menggunakan mikroprosesor ATMega168. Berbeda dengan versi sebelumnya yang mengharuskan menekan tombol reset pada saat mengunggah program ke mikrokontroler, proses reset pada Diecimila sudah dilakukan otomatis oleh aplikasi pengunggah program yang ada di komputer. Selain itu pada Diecimila sudah terpasang LED yang terhubung dengan pin 13.

Kini sudah tersedia berbagai varian Arduino dengan variasi spesifikasi mikrokontroller yang digunakan dan fitur-fitur lainnya yang mendukung penggunaan yang lebih luas. Varian standar yang paling sering digunakan terutama untuk belajar Arduino adalah board Arduino Uno. Arduino Uno yang sekarang sudah menggunakan mikrokontroler ATMega328. Board ini memiliki 14 pin input/output digital yang mana 6 diantaranya dapat digunakan untuk output PWM (Pulse Width Modulation), 6 pin input analog, kristal quartz 16MHz. Selain itu board ini juga dilengkapi power jack untuk port catudaya, pin header untuk koneksi ICSP dan tombol reset.

Arduino Uno dan beberapa varian lain yang memiliki posisi pin header female dengan pola yang tertentu dibuat standar agar dapat dipasangi board tambahan lain untuk meningkatkan kemampuan board dengan komponen lain seperti sensor, chip, ekstra pin dan port. Board ekstensi ini disebut dengan istilah shield. Ada banyak beragam shield yang tersedia di pasaran. Kita tinggal pasang dengan ditumpuk di atas board utama dan shield pun langsung dapat diprogram.

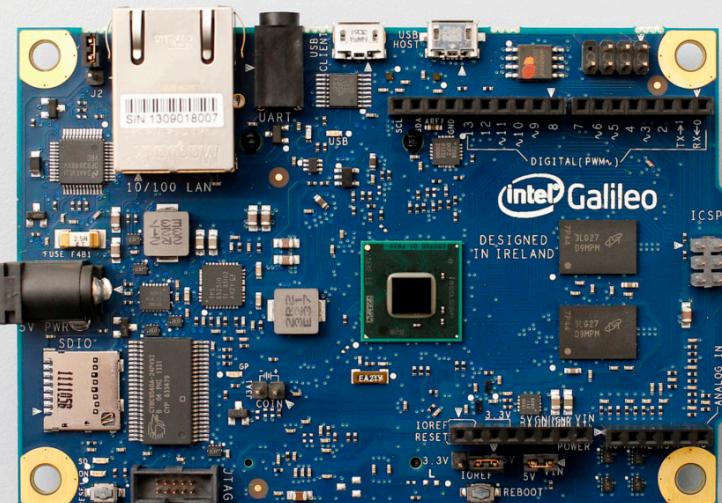


Gambar 4 Arduino shield yang terpasang pada Arduino Uno (wikipedia.org)

Varian Arduino dan Proyek Turunannya

Ada banyak varian Arduino, yang masing-masing dibedakan menurut kekuatan komputasi mikrokontroler, jumlah pin dan variasi input/output, serta ukuran papan. Beberapa varian terkenal diantaranya Arduino/Genuino Uno, Arduino Pro, Arduino Micro, Arduino Pro Mini, Arduino Nano. Varian-varian tersebut dikelompokkan ke dalam kategori entry level yang berarti cocok untuk digunakan dalam pengenalan dan pembelajaran dasar. Adapun varian di atasnya yang memiliki spesifikasi yang jauh lebih tinggi, performa lebih cepat dan cocok untuk proyek yang lebih besar dan kompleks dikategorikan dalam enhanced features, yang mana diantaranya adalah Arduino Mega, Arduino Zero, dan Arduino Due. Ada pula kelompok varian yang dikembangkan khusus untuk keperluan IoT atau Internet of Things, seperti Arduino Yun, serta beberapa papan shield seperti Arduino Ethernet Shield, Arduino GSM Shield, dan Arduino Wifi Shield. Selain itu ada juga kelompok varian yang dikembangkan khusus untuk penerapan di bidang fashion dan segala perlengkapan aksesoris, diantaranya Arduino Gemma, Lilypad Arduino, Lilypad Simple Arduino, dan Lilypad USB

Arduino. Kesemua varian tersebut masih diproduksi hingga saat tulisan ini diterbitkan dan dapat dipesan secara online melalui distributor-distributor yang tersebar di seluruh dunia.



Gambar 5 Intel Galileo, prototype board dengan mikrokontroller Intel yang support Arduino (arduino.cc)

Selain dari varian resmi yang dikembangkan oleh tim utama Arduino, disebabkan oleh karena sifatnya yang open source, bermunculanlah produk-produk papan prototype lain yang memanfaatkan bahasa pemrograman dan aplikasi Arduino IDE sebagai basis pemrogramannya, dengan pertimbangan sudah umum dan memasyarakatnya Arduino, maka pembuatan board lain dengan basis pemrograman Arduino meningkatkan potensi penggunaan dan penjualan yang tinggi karena pengguna tidak perlu lagi mempelajari bahasa pemrograman baru dan menginstal aplikasi IDE baru. Beberapa proyek prototype board yang dibuat yang dapat diprogram menggunakan Arduino diantaranya Intel Galileo, Intel Edison, LittleBits, Spark Core, Digispark, Microduino, BLEduino, Teensy, Paperduino, dan masih banyak lagi.

Produk-produk baru bermunculan, terutama atas peran situs-situs crowdfunding seperti Kickstarter dan Indiegogo yang membolehkan siapapun untuk mengkampanyekan penggalangan dana untuk membantu proyeknya. Selain itu ada banyak event maker seperti Maker Faire yang turut mendukung para pengembang dan hobbyist untuk memamerkan kreasi mereka.



Gambar 6 Keyboardio, salahsatu produk kreasi menggunakan Arduino (shop.keyboard.io)

Seberapa Mudah Penulisan Program pada Arduino

Arduino dapat diprogram menggunakan bahasa pemrograman C dan C++. Tapi secara default Arduino sudah menyiapkan bahasa pemrogramannya sendiri untuk lebih menyederhanakan penulisan sintaks.

Kalo kita pertama kali belajar bahasa pemrograman, kode pertama yang kita buat biasanya adalah ‘Hello World’, ya kan? Nah, di Arduino, script pertama yang disodorkan untuk mengetes board Arduino adalah program Blink. Program ini berisi kode untuk menyalakan dan mematikan LED pada board terus menerus setiap jeda 1 detik. Kodenya dapat dilihat pada tabel berikut:

```
1 void setup() {  
2     pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(13, HIGH);  
7     delay(1000);  
8     digitalWrite(13, LOW);  
9     delay(1000);  
10 }
```

Untuk dapat mengontrol *hardware*, paling tidak ada dua fungsi yang harus digunakan, yaitu fungsi `setup()` dan `loop()`. Fungsi `setup()` adalah fungsi yang dijalankan pertama kali saat *board* dinyalakan. Kemudian fungsi `loop()` akan dipanggil terus menerus selama *board* menyala.

Pada baris dua terdapat kode untuk menginisiasi pin mode 13 untuk digunakan sebagai keluaran. Pin 13 ini terhubung langsung dengan LED

yang diteruskan ke *ground*. Dengan kata lain ketika pin 13 mengeluarkan sinyal listrik, maka LED akan menyala.

Baris 6 adalah kode untuk memberikan output HIGH (tegangan 5V) supaya LED menyala. Setelah LED menyala, buat sistem untuk menunggu selama 1000 milidetik atau 1 detik, seperti pada baris 7. Baris 8 dan 9 sama seperti baris di atasnya, hanya saja set output ke LOW membuat tengangan output menjadi 0V. Lihat *kan*. Dengan beberapa baris sederhana kita sudah bisa berinteraksi dengan lingkungan fisik. Masih banyak lagi komponen elektronik lain yang bisa kita gunakan untuk melakukan fungsi-fungsi lainnya.

* * *

So, secara umum Arduino adalah pilihan yang bagus untuk memulai belajar pemrograman *hardware* dan membuat proyek-proyek keren yang menyenangkan. Ada banyak prototype board lain di luar Arduino yang bisa dipilih sebenarnya, tapi kalo Kamu orang baru yang ingin mencicipi mainan baru di ranah pemrograman *hardware*, Arduino adalah pilihan yang tepat. Selain itu, *IoT* adalah teknologi masa depan yang akan terus berkembang, dan pemrograman hardware adalah salah satu komponen wajib untuk dipelajari untuk masuk ke ranah sana. Mau masuk ke bidang tersebut ataupun tidak, setidaknya Arduino bisa jadi mainan asyik buat Kamu para programmer :D

Menginstall dan Menggunakan Arduino IDE

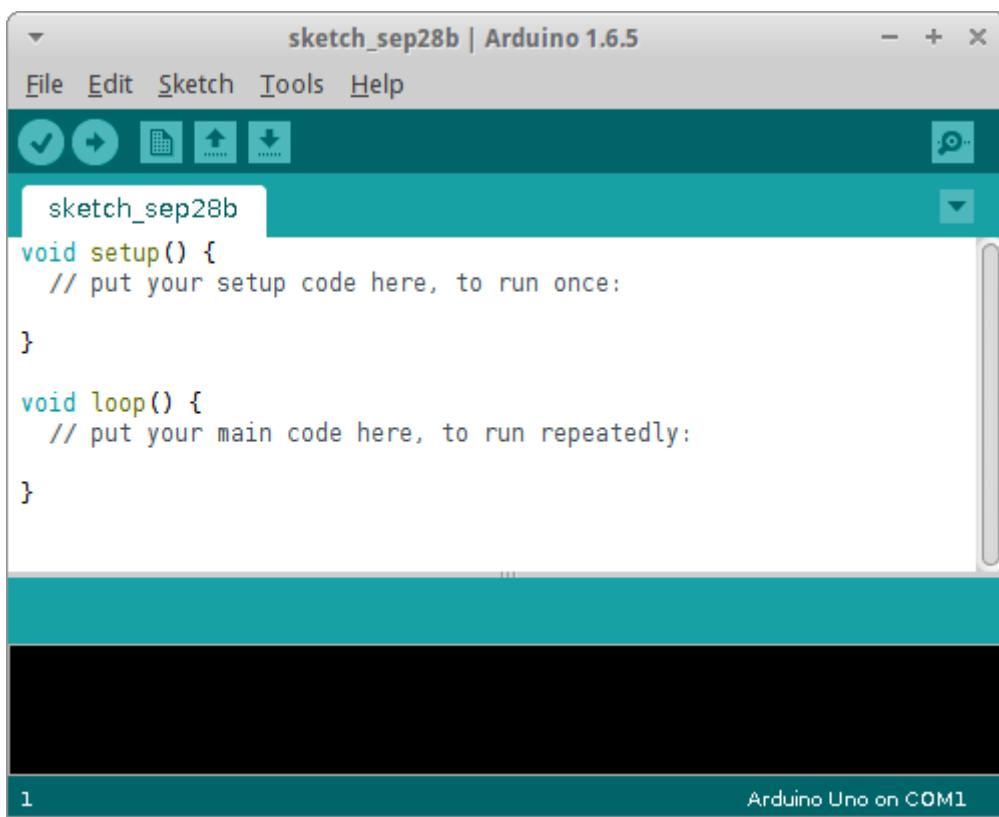
Arduino sudah menyediakan Arduino IDE (Integrated Development Environment) atau aplikasi yang sudah menyediakan fungsi lengkap untuk membangun program, mulai dari editor kode untuk menulis kode program, fungsi *debugging* dan *compile* program sampai mengunggah program ke mikrokontroller. Untuk dapat memprogram *board* Arduino, kita hanya

perlu menginstall dan menggunakan satu aplikasi ini saja. Arduino IDE juga merupakan aplikasi *cross-platform* yang dibangun menggunakan Java, yang artinya dapat diinstal di sistem operasi manapun yang mendukung Java Runtime Environment (JRE).

Instalasi Arduino IDE

Untuk menginstall Arduino IDE, Kamu bisa langsung menuju ke halaman unduh software di website Arduino ini <https://www.arduino.cc/en/Main/Software>. Ada tiga paket Arduino IDE yang disediakan, yakni untuk sistem operasi Windows, Mac OS X dan Linux. Unduh aplikasi Arduino IDE yang sesuai dengan sistem operasi Kamu. Kemudian ikuti langkah-langkah instalasinya.

Tampilan antarmuka Arduino IDE dapat dilihat seperti pada gambar di bawah ini:



Seperti aplikasi pada umumnya, Arduino IDE memiliki titlebar (bagian paling atas aplikasi, berisi tulisan nama file dan nama aplikasi, serta beberapa tombol untuk *minimize*, *maximize* dan *close*), menubar (bagian di bawah titlebar, berisi menu-menu aplikasi seperti File, Edit, dan sebagainya), toolbar (berisi ikon-ikon tombol akses cepat), dan statusbar (bagian paling bawah pada aplikasi). Selain itu terdapat pula *code editor* tempat kita menuliskan kode program, dan kolom *debugging* untuk menampilkan pesan status *debugging* dan *compile* (bagian kotak berlatar hitam di bawah *code editor*).

Cara pakenya mudah. Tinggal tulis kodemu di bagian *code editor*. Pada toolbar terdapat dua ikon paling kiri, yakni Verify (tombol dengan ikon ceklis) dan Upload (tombol dengan ikon panah ke kanan). Tombol Verify berfungsi untuk meng-*compile* kode program yang kamu buat dan mengecek apakah sudah valid atau masih terdapat kesalahan. Tombol Upload berfungsi untuk meng-*compile* dan mengunggah program yang sudah di-*compile* ke dalam memori penyimpanan mikrokontroller.

Contoh Kode Program Arduino

Untuk memulai belajar pemrograman Arduino, Arduino IDE sudah menyediakan banyak sekali contoh kode program yang dapat diakses melalui menu File > Examples. Pada banyak tutorial dasar yang tersebar di internet, hampir setiap contoh program pertama menggunakan program Blink, yang mana dapat diakses pada bagian 01.Basics pada menu Examples. Program Blink tersebut setelah dicompile dan diunggah akan membuat mikrokontroler menyalaikan dan mematikan LED yang ada pada board Arduino.

Namun perlu diingat bahwa *board* Arduino itu sendiri hanyalah papan kontrol yang tidak banyak berguna bila tidak dikombinasikan dengan komponen mikroelektronik lain seperti sensor dan aktuator. Misalkan

Kamu ingin menangkap nilai intensitas cahaya di ruangan, maka kamu dapat menggunakan komponen LDR (Light Emitting Diode) untuk menangkap sinyal cahaya. LDR memiliki dua kaki, yang satu dipasangkan pada pin analog untuk menerima sinyal analog, dan pin satunya lagi dipasangkan pada pin 5V. Sinyal analog tersebut kemudian dikonversikan ke dalam nilai digital menggunakan fungsi `analogRead()` dan kita dapat menampilkan hasilnya dengan mencetak nilai tersebut pada jendela Serial Monitor pada Arduino IDE (tombol paling kanan pada toolbar dengan ikon bergambar kaca pembesar). Dan kalo Kamu masih bingung bagaimana menuliskan program untuk kasus tersebut, Kamu bisa lihat contoh programnya pada menu Examples > 01.Basics, program bernama `AnalogReadSerial`. Ada banyak contoh program yang bisa kita eksplorasi. Adapun untuk ketersediaan komponen sensor dan aktuator, kita sudah mudah diperoleh, baik dari toko online ataupun toko elektronik terdekat.

Memilih Board yang Digunakan

Arduino memiliki banyak varian *board* mikrokontroler, seperti Arduino Uno, Arduino Mega, Arduino Mikro, dan lain sebagainya. Setiap *board* memiliki konfigurasinya masing-masing. Tapi dengan menggunakan Arduino IDE, kita cukup memilih *board* apa yang digunakan pada menu Tools > Board. Kalo Kamu akan memprogram Arduino Uno, maka pilih Board Arduino Uno.

Menginstall Digispark Library pada Arduino IDE

Kamu sudah tahu teknik dasar dalam menggunakan Arduino IDE. Sekarang, karena rubrik Weekend Project menggunakan *board* Digispark, sedangkan Digispark adalah *board* yang librarynya tidak tersedia secara

default pada Arduino IDE. Buktinya adalah tidak ada pilihan Digispark pada menu Tools > Board. Maka kita mesti mengunduh librarynya terlebih dahulu. Langkah-langkah untuk memasang library Digispark adalah sebagai berikut:

Jalankan aplikasi Arduino IDE

Buka menu File > Preferences. Akan muncul jendela Preferences. Pada jendela Preferences, pada kolom Additional Boards Manager URLs, isikan URL “http://digistump.com/package_digistump_index.json”. Tekan tombol OK.

Buka menu Tools > Board > Boards Manager. Akan muncul jendela Boards Manager. Pada dropdown Type, pilih Contributed. Klik paket “Digistump AVR Boards” dan tekan tombol Install di sebelah kanan paket.

Tunggu aplikasi selesai mengunduh library Digispark. Untuk menginstal library ini Kamu memerlukan koneksi internet.

Pada sistem operasi windows, ketika instalasi selesai akan muncul jendela *pop up* instalasi driver. Klik tombol Next untuk menginstall driver Digistump boards.

Setelah library Digispark berhasil terinstall, Kamu dapat langsung menggunakan *board* Digispark dengan terlebih dahulu memilih board “Digispark (Default – 16,5 mhz)” pada menu Tools > Board.

Yosh! Persiapan Arduino IDE sudah beres. Selanjutnya terserah Anda! Hehe.. Selamat berprojek akhir pekan! :D

DON'T READ This POSTER Or You Will CONNECT With US

 CodePolitan

 @CodePolitan

 +CodePolitan

 @CodePolitan

{• CodePolitan

Media Edukasi Online Seputar Pemrograman Berbasis Komunitas

Visit Us at www.codepolitan.com