



SERI MANAJEMEN PROYEK SOFTWARE

---

## Markdown dan Pandoc

---

MEMBUAT DOKUMENTASI DENGAN MARKDOWN DAN PANDOC

*Endy Muhardin*  
5 September 2012  
Versi: 1.1



© 2013 ArtiVisi Intermedia



# Daftar Isi

---

<b>1</b>	<b>Pendahuluan</b>	<b>1</b>
1.1	Tentang Markdown . . . . .	1
1.1.1	Apa itu Markdown . . . . .	1
1.1.2	Mengapa memilih Markdown . . . . .	1
1.2	Tentang Pandoc . . . . .	3
1.3	Tentang Buku Ini . . . . .	4
1.3.1	Mengapa buku ini ditulis . . . . .	4
1.3.2	Siapa yang sebaiknya membaca . . . . .	4
1.3.3	Lisensi . . . . .	4
1.3.4	Tools . . . . .	5
1.3.5	Kontribusi . . . . .	5
1.3.6	Reviewer . . . . .	5
1.3.7	Penulis . . . . .	5
<b>2</b>	<b>Sintaks Markdown</b>	<b>7</b>
2.1	Paragraf . . . . .	8
2.2	Headers . . . . .	8
2.2.1	Pemberian ID untuk Header . . . . .	8
2.3	Kutipan . . . . .	9
2.4	Kode Program . . . . .	10
2.5	List dan Nomor . . . . .	10
2.5.1	List . . . . .	10
2.5.2	Nomor . . . . .	11
2.6	Garis Mendatar . . . . .	12
2.7	Tabel . . . . .	13
2.8	Cover . . . . .	14
2.9	Backslash . . . . .	14
2.10	Inline Formatting . . . . .	15

2.10.1	Huruf Miring . . . . .	15
2.10.2	Huruf Tebal . . . . .	15
2.10.3	Strikeout . . . . .	15
2.10.4	Verbatim . . . . .	16
2.11	Link . . . . .	16
2.11.1	Link Otomatis . . . . .	16
2.11.2	Inline Link . . . . .	16
2.11.3	Reference Link . . . . .	17
2.11.4	Inline Link . . . . .	17
2.12	Image . . . . .	17
2.13	Catatan kaki . . . . .	19
2.14	Referensi dan Bibliografi . . . . .	20
<b>3</b>	<b>Output</b>	<b>21</b>
3.1	Beberapa kesepakatan . . . . .	21
3.1.1	Lokasi eksekusi perintah . . . . .	21
3.1.2	Ekstensi File . . . . .	21
3.2	PDF . . . . .	22
3.2.1	Perintah Dasar . . . . .	22
3.2.2	Menggunakan L <sup>A</sup> T <sub>E</sub> X Template . . . . .	22
3.2.3	Manual Page Break . . . . .	22
3.2.4	Link dan Catatan Kaki . . . . .	23
3.2.5	Tanpa Warna . . . . .	23
3.3	Slide Presentasi . . . . .	24
3.4	Open Office . . . . .	24
3.5	HTML . . . . .	25
<b>4</b>	<b>Penutup</b>	<b>27</b>

# BAB 1 | Pendahuluan

---

## 1.1 Tentang Markdown

### 1.1.1 Apa itu Markdown

Markdown<sup>1</sup> adalah format penulisan dokumen yang ditemukan oleh John Gruber<sup>2</sup>. Markdown dibuat dengan filosofi sebagai berikut:

The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. - John Gruber<sup>3</sup>

Markdown memiliki beberapa karakteristik, diantaranya:

- berbasis teks biasa (plain text)
- mudah dibaca apa adanya (tanpa aplikasi khusus)
- bisa dikonversi menjadi format lain (misalnya PDF, HTML, dsb)

### 1.1.2 Mengapa memilih Markdown

Di ArtiVisi, kita ingin melakukan standarisasi dalam penulisan dokumen. Kriteria utama dari format dokumentasi yang kita inginkan adalah **berbasis text**. Format berbasis text memiliki beberapa keunggulan, diantaranya:

- bisa memanfaatkan fitur version control secara maksimal seperti diff, blame, branch, merge, dan sebagainya.
- mudah diedit, pakai notepad pun jadi

---

<sup>1</sup><http://daringfireball.net/projects/markdown/>

<sup>2</sup><http://daringfireball.net/>

<sup>3</sup><http://daringfireball.net/projects/markdown/>

- bisa diedit dimana saja, misalnya di tablet atau handphone (contohnya bab ini saya ketik di angkutan umum menggunakan aplikasi DroidEdit di handphone)
- ukurannya kecil
- mudah disimpan di layanan berbasis cloud seperti Dropbox

Ada beberapa format yang memenuhi kriteria tersebut, diantaranya:

- html
- markdown
- docbook
- $\text{\LaTeX}$

Aplikasi Office baik Microsoft maupun Open langsung gugur pada babak pertama ini karena formatnya bukan text.

Selanjutnya, kriteria kedua adalah mudah dipahami. Pekerjaan membuat dokumentasi di ArtiVisi biasanya diserahkan ke anak magang. Untuk itu, kita harus bisa mengajarkan sistem dokumentasi ini dengan cepat, karena tingkat turnover anak magang relatif singkat. Jangan sampai terjadi, masa magangnya cuma 3 bulan, 2 minggu dihabiskan untuk belajar sistem dokumentasi.

Dari empat kandidat di atas,  $\text{\LaTeX}$  gugur karena dia relatif sulit dipelajari. Tinggal tersisa tiga kontestan. Kita lanjutkan ke babak selanjutnya.

Kriteria ketiga adalah kemampuan konversi ke berbagai format lain. Kita ingin konversi ke PDF agar bisa dicetak. Kita juga ingin bisa konversi ke HTML agar bisa dipublish di internet. Di jaman modern ini, perlu juga kemampuan konversi menjadi format buku digital (epub). Selain itu, kadangkala perlu juga konversi ke format OpenOffice atau MS Office.

Di babak ketiga ini, format HTML gugur. Untuk mengkonversi format HTML ke format lain, dibutuhkan prosedur yang rumit dan panjang. Dengan demikian, kita sudah mencapai babak final, mempertandingkan Markdown vs Docbook.

Dari sini, sebetulnya kedua format sudah layak untuk digunakan. Walaupun demikian, kita akan melihat beberapa kriteria lain yang bersifat *nice to have*, ada bagus, tidak ada juga tidak apa-apa.

Kriteria tambahan pertama adalah ketersediaan tools. Docbook dapat dikonversi menjadi beberapa format dengan banyak tools, diantaranya:

- Maven (Java)
- Bookshop (Ruby)

- Pandoc (Haskell)
- Tools lain (misalnya xsltproc, makefile, dsb)

Sedangkan untuk Markdown, sebagian besar toolsnya adalah untuk konversi menjadi HTML, seperti misalnya:

- Markdown.pl
- Pandoc
- Jekyll

Untuk urusan tools juga nilainya seri. Kedua format dapat dikonversi dengan mudah ke format lainnya. Mari kita tinjau dari sisi user.

Nantinya, urusan dokumentasi akan diserahkan ke kasta terbawah dari rantai makanan programmer, yaitu para magang dan karyawan baru. Karakteristik dari user ini adalah tingkat ketelitiannya yang minim, tidak seperti para senior programmer yang dapat menemukan kelebihan satu spasi (*ingat, spasi tidak terlihat oleh mata telanjang*) di dalam ribuan baris kode program.

Oleh karena itu, format dokumentasi harus *foolproof*. Dia harus bisa ditulis dengan tingkat ketelitian yang rendah. Nah, di sini kita menemukan pemenangnya. Docbook berbasis XML yang terkenal ketat (strict). Kurang satu tanda kutip saja, dokumen tidak dapat diproses. Ini menyebabkan format ini rentan error. Jangan sampai nanti para senior programmer harus direpotkan karena dokumennya tidak dapat dikonversi dengan baik.

Berbeda halnya dengan Markdown. Dia sangat permisif. Walaupun ada kesalahan, paling hanya berdampak merusak format tampilan, tidak sampai error pemrosesan.

Dengan demikian, format yang akan kita gunakan adalah Markdown.

## 1.2 Tentang Pandoc

Dokumen dalam format text saja tidak cukup. Seringkali kita ingin mengirimkan dokumen tersebut ke orang lain, yang kemudian akan dicetak, ataupun ditampilkan di website. Untuk itu, kita butuh format lain seperti misalnya PDF dan HTML.

Kita bisa mengkonversi Markdown menjadi PDF dan HTML menggunakan aplikasi yang bernama Pandoc<sup>4</sup>. Pandoc ini adalah aplikasi yang dibuat oleh John MacFarlane<sup>5</sup>, seorang profesor filosofi di University of California, Berkeley. Pandoc dibuat dengan bahasa pemrograman Haskell<sup>6</sup>.

<sup>4</sup><http://johnmacfarlane.net/pandoc/>

<sup>5</sup><http://johnmacfarlane.net>

<sup>6</sup>[http://en.wikipedia.org/wiki/Haskell\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Haskell_(programming_language))

## 1.3 Tentang Buku Ini

### 1.3.1 Mengapa buku ini ditulis

Di ArtiVisi, kami banyak menulis buku, modul pelatihan, slide presentasi, user manual aplikasi, dan berbagai dokumen lainnya. Daripada menjelaskan berkali-kali setiap ada karyawan atau magang yang baru bergabung, lebih baik saya investasikan waktu dan tenaga satu kali saja untuk menulis panduan ini.

### 1.3.2 Siapa yang sebaiknya membaca

Buku ini bisa bermanfaat buat:

- Karyawan / magang di ArtiVisi yang ditugaskan membuat buku, modul pelatihan, slide presentasi, user manual, dan sebagainya.
- Guru, dosen, atau mahasiswa yang ingin membuat skripsi, thesis, atau tulisan ilmiah lainnya. Dengan Pandoc, dokumen Markdown bisa dikonversi menjadi  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , yaitu format dokumen yang lazim digunakan untuk membuat tulisan ilmiah. Format  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ini jauh lebih superior daripada word processor seperti MS Word atau OpenOffice Writer<sup>7</sup>.

### 1.3.3 Lisensi

Buku ini memiliki lisensi Creative Commons Attribution Share Alike (CC-BY-SA). Artinya, semua orang:

- bebas menggunakan buku ini tanpa harus membayar, baik untuk keperluan non-profit maupun komersil. Anda boleh membuka pelatihan berbayar menggunakan buku ini.
- bebas membagikan buku ini kepada siapa saja.
- bebas membuat perubahan terhadap isi buku ini.

Semua kebebasan di atas hanya memiliki syarat yaitu tetap harus menyebutkan nama pengarang yang aslinya. Ini disebut dengan istilah attribution. Singkatnya, boleh dipakai dan dibagikan asal jangan diakui sebagai karya sendiri. Selain itu, segala perubahan yang dibuat juga harus dilisensikan sama dengan buku ini. Ini disebut dengan istilah Share-Alike. Lebih lanjut tentang lisensi ini bisa dilihat di website Creative Commons<sup>8</sup>

<sup>7</sup><http://ricardo.ecn.wfu.edu/~cottrell/wp.html>

<sup>8</sup><http://creativecommons.org/licenses/>



### 1.3.4 Tools

Buku ini dibuat menggunakan perangkat pembantu :

- Markdown : format text untuk menulis buku
- Pandoc : aplikasi untuk mengkonversi markdown menjadi PDF atau HTML
- $\LaTeX$ : format perantara dari Markdown menjadi PDF

### 1.3.5 Kontribusi

Semua orang boleh dan dianjurkan untuk ikut membantu penulisan buku ini. Bagaimana caranya? Gampang. Ada beberapa pekerjaan yang dapat dilakukan.

### 1.3.6 Reviewer

Tugasnya adalah memeriksa isi buku dan memberikan koreksi. Apa saja boleh dikoreksi, mulai dari tanda baca, salah ketik, contoh latihan tidak bisa dijalankan, apa saja. Kalau ada penjelasan yang kurang jelas juga boleh dikomentari. Apapun yang bisa membuat buku ini lebih baik. Hasil review dapat dientri di halaman Issue di Github<sup>9</sup>.

### 1.3.7 Penulis

Bagus sekali kalau Anda ingin menyumbangkan tulisan. Lebih banyak yang mencerdaskan bangsa lebih baik. Begini caranya. Sebagai penulis buku Git, tentu Anda sudah paham cara menggunakan Git, dan juga kemungkinan besar sudah punya account di Github. Langsung saja fork repository buku-git ini dan segeralah berkarya. Begitu dirasa sudah memadai, kirimkan pull request ke saya. Nanti akan saya merge ke repository saya.

---

<sup>9</sup><https://github.com/endymuhardin/buku-pandoc/issues>



## BAB 2 | Sintaks Markdown

---

Untuk menulis buku, kita menggunakan sintaks markdown. Markdown pertama kali diciptakan oleh John Gruber. Berikut filosofi Markdown menurut John Gruber.

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber<sup>1</sup>

Diterjemahkan secara bebas sebagai berikut.

Dokumen berformat Markdown harus bisa diterbitkan apa adanya dalam bentuk *plain text*, tanpa terlihat bahwa dokumen tersebut sudah dihiasi perintah formatting tertentu.

Versi Markdown asli dari John Gruber memiliki beberapa kekurangan, diantaranya dia tidak menjelaskan bagaimana cara membuat tabel. Selain versi John Gruber, ada juga versi lain seperti MultiMarkdown dan Pandoc.

Kita akan menggunakan versi Pandoc yang lebih lengkap, karena sudah mendukung:

- tabel
- mewarnai source code
- sampul
- catatan kaki

Selanjutnya, kita akan membahas tentang cara-cara memformat tulisan dengan sintaks markdown aliran Pandoc.

---

<sup>1</sup><http://daringfireball.net/projects/markdown/syntax#philosophy>

## 2.1 Paragraf

Paragraf adalah satu atau lebih baris tulisan. Paragraf satu dan lainnya dibatasi oleh satu baris kosong. Pergantian baris yang kita ketik tidak menentukan pergantian baris di hasil akhir, sehingga kita bisa mengganti baris sesuka kita. Kalau kita menginginkan hasil akhirnya juga ganti baris, kita dapat menggunakan spasi ganda di akhir baris, atau backslash (\) diikuti oleh baris baru.

## 2.2 Headers

Ada dua jenis penulisan header, setext dan atx. Kita hanya akan membahas atx saja.

Atx-header diawali oleh satu sampai enam tanda #. Berikut contohnya:

```
# Bab 1

## Sub bab 1.1

### Subnya sub bab 1.1.1
```

Kita boleh menambahkan akhiran # ataupun menghilangkannya. Contoh berikut:

```
# Bab 1

    sama dengan ini:

# Bab 1 #
```

### 2.2.1 Pemberian ID untuk Header

Pada waktu dikonversi menjadi HTML, tiap header akan diberikan id supaya bisa dijadikan tautan. Berikut aturan pemberian id:

- Semua formatting, link, dsb, dihilangkan.
- Semua tanda baca dihapus kecuali underscore, hyphen, dan titik.
- Mengganti spasi dan ganti baris dengan hyphen.
- Semua huruf dijadikan huruf kecil.
- Semua karakter aneh dihapus sampai huruf pertama (id tidak boleh diawali angka atau tanda baca).

- Bila habis semua, gunakan id section.

Contohnya:

Header	Identifier
Header identifiers in HTML	header-identifiers-in-html
<i>Dogs?</i> –in <i>my</i> house?	dogs--in-my-house
<b>HTML</b> , [S5], or [RTF]?	html-s5-or-rtf
3. Applications	applications
33	section

Bila dalam prosesnya ditemukan identifier yang sama, maka akan ditambahkan angka urutan, misalnya `section-1`, `section-2`, dan seterusnya.

Identifier ini bisa digunakan sebagai link di dalam dokumen, seperti ini:

Silahkan lihat di

[penjelasan tentang sintaks markdown](#sintaks-markdown).

Selain itu, identifier ini juga digunakan dalam pembuatan daftar isi.

## 2.3 Kutipan

Untuk menampilkan kutipan, kita menggunakan sintaks seperti pada waktu membalas email, yaitu menggunakan penanda `>`. Penanda ini bisa ditulis hanya di awal kutipan seperti ini:

```
> The difference between stupidity and genius
is that genius has its limits.
```

Dan bisa juga di tiap barisnya seperti ini:

```
> Have you ever noticed that
> anybody driving slower than you is an idiot,
> and anyone going faster than you is a maniac?
```

Kecuali di awal file, kutipan harus didahului satu baris kosong. Berikut adalah contoh tampilan kutipan.

When you borrow five thousand from a bank and you can't pay back, you've got a problem. When you borrow five million from a bank and you can't pay back, they've got a problem.

## 2.4 Kode Program

Untuk menampilkan kode program, kita menggunakan tiga backtick (`), seperti ini:

```
```\nSystem.out.println("Halo dunia");\n```
```

Ini akan menghasilkan tampilan seperti ini:

```
System.out.println("Halo dunia");
```

Kita bisa mengaktifkan *syntax highlighting* dengan mencantumkan jenis bahasa pemrograman yang digunakan.

```
```java\nSystem.out.println("Halo dunia");\n```
```

Ini akan menghasilkan tampilan seperti ini:

```
System.out.println("Halo dunia");
```

Kalau kita output menjadi PDF, contoh kedua ini akan diwarnai sesuai keyword bahasa pemrograman yang dipilih.

## 2.5 List dan Nomor

### 2.5.1 List

List diawali oleh tanda \*, +, atau -. Berikut contohnya:

```
* apel\n* mangga\n* durian
```

Bila terlalu panjang, kita bebas untuk ganti baris, seperti ini:

- \* markdown. Ini adalah  
format yang ditemukan oleh John Gruber
- \* docbook
- \* latex

Kita juga bisa membuat list di dalam list, seperti ini:

- \* buah
  - + apel
  - + mangga
    - indramayu
    - harum manis
  - + durian
- \* sayur
  - + kangkung
  - + bayam

Syaratnya, list level kedua harus diindentasi 4 spasi. Contoh di atas akan menghasilkan tampilan seperti ini:

- buah
  - apel
  - mangga
    - \* indramayu
    - \* harum manis
  - durian
- sayur
  - kangkung
  - bayam

### 2.5.2 Nomor

Secara garis besar, nomor sama dengan list. Bedanya cuma di karakter penandanya. Nomor ditandai dengan angka seperti ini:

1. Adi
2. Awan
3. Ifnu

Nomornya tidak harusurut, begini juga boleh:

1. Adi
3. Awan
1. Ifnu

Nomor awal akan mengikuti nomor pertama yang kita gunakan. Bila kita tulis seperti ini:

4. Adi
2. Awan
3. Ifnu

maka hasilnya adalah

4. Adi
5. Awan
6. Ifnu

Selain menggunakan angka arab, kita juga bisa menggunakan huruf dan angka romawi. Selain itu, masih ada fitur lain seperti penomoran contoh, cara memutus list, dan sebagainya. Detailnya bisa dilihat di dokumentasi Pandoc<sup>2</sup>.

## 2.6 Garis Mendatar

Baris yang memuat tiga atau lebih karakter \*, -, atau \_ secara berturut-turut (boleh diselingi spasi) akan dikonversi menjadi garis mendatar.

Contohnya sebagai berikut:

```
***
halo
- - -
```

akan menghasilkan output seperti ini:

---

<sup>2</sup><http://johnmacfarlane.net/pandoc/README.html>



## 2.7 Tabel

Berikut contoh cara penulisan tabel:

Kanan	Kiri	Tengah	Default
12	12	12	12
123	123	123	123
1	1	1	1

Table: Contoh tabel sederhana.

Yang akan menghasilkan tabel seperti ini:

Kanan	Kiri	Tengah	Default
12	12	12	12
123	123	123	123
1	1	1	1

Tabel 2.2: Contoh tabel sederhana.

Dari contoh di atas, kita juga dapat memahami cara membuat rata kiri, rata kanan, dan tengah. Selain itu, kita juga bisa memberikan judul tabel.

Bila satu row terdiri dari banyak baris, contohnya seperti ini:

Rata Tengah	Default Aligned	Rata Rata Kanan Kiri
Pertama	halo	12.0 Contoh row yang lebih dari satu baris
Kedua	coba	5.0 Ini row kedua. Antar row dipisahkan oleh baris kosong.

Table: Ini labelnya.

Label juga boleh multi baris

Hasilnya seperti ini:

	Default		
Rata Tengah	Aligned	Rata Kanan	Rata Kiri
Pertama	halo	12.0	Contoh row yang lebih dari satu baris
Kedua	coba	5.0	Ini row kedua. Antar row dipisahkan oleh baris kosong.

Tabel 2.3: Ini labelnya. Label juga boleh multi baris

## 2.8 Cover

Untuk membuat cover buku, kita harus membuat satu file berisi title block seperti ini:

```
% Menggunakan Pandoc dan Markdown
% Endy Muhardin
% 5 September 2012
```

Baris pertama adalah judul, baris kedua adalah pengarang, dan baris ketiga adalah tanggal penulisan. Bila pengarang lebih dari satu, ditulis seperti ini:

```
% Menggunakan Pandoc dan Markdown
% Endy Muhardin
  Ifnu Bima
% 5 September 2012
```

## 2.9 Backslash

Kecuali dalam code block atau inline code, semua tanda baca dan spasi yang didahului backspace akan ditulis apa adanya, termasuk karakter formatting. Contoh:

v1.1

```
*\*halo\**
```

akan menghasilkan

```
<em>*halo*</em>
```

bukannya

```
<strong>halo</strong>
```

## 2.10 Inline Formatting

### 2.10.1 Huruf Miring

Untuk membuat *huruf miring*, gunakan karakter `*` atau `_`. Contohnya:

Penulisan kode program di Java adalah `_case sensitive_`.  
Berbeda dengan SQL yang `*case insensitive*`.

Contoh di atas akan menghasilkan tampilan seperti ini:

Penulisan kode program di Java adalah *case sensitive*. Berbeda dengan SQL yang *case insensitive*.

### 2.10.2 Huruf Tebal

Huruf **tebal** sama dengan huruf miring, tapi menggunakan dua karakter. Contohnya:

Penulisan kode program di Java adalah `__case sensitive__`.  
Berbeda dengan SQL yang `**case insensitive**`.

Contoh di atas akan menghasilkan tampilan seperti ini:

Penulisan kode program di Java adalah **case sensitive**. Berbeda dengan SQL yang **case insensitive**.

### 2.10.3 Strikeout

Untuk menulis ~~correction~~ koreksi, gunakan `~~` seperti ini:

Kelemahan Spring Framework adalah  
~~keharusan untuk menulis file xml yang banyak~~.

### 2.10.4 Verbatim

Verbatim artinya apa adanya, tidak dikonversi menjadi apapun. Untuk membuat verbatim, kita gunakan backtick seperti ini:

Untuk membandingkan angka, gunakan tanda lebih besar ``>``

## 2.11 Link

### 2.11.1 Link Otomatis

Markdown bisa mengkonversi tulisan menjadi link secara otomatis, asal diapit oleh kurung siku. Berikut contohnya:

```
<http://software.endy.muhammadin.com>  
<endy.muhammadin@gmail.com>
```

### 2.11.2 Inline Link

Link dan URL yang dituju bisa langsung digabungkan dalam satu bagian.

Untuk menampilkan link ke website tertentu, formatnya seperti ini:

```
Silahkan lihat ke  
[website saya](http://software.endy.muhammadin.com/about "Websitenya Endy")  
untuk informasi lebih lengkap.
```

Ada tiga bagian untuk membuat link, yaitu:

1. Tulisan yang akan diberi link. Ditulis di dalam kurung kotak []
2. URL untuk link tersebut, ditulis dalam tanda kurung biasa ()
3. Judul link, ditulis dalam tanda kurung, setelah URL, dilengkapi dengan tanda kutip ganda. Judul link ini optional, boleh ada ataupun tidak.

Contoh di atas akan menghasilkan output seperti ini:

Silahkan lihat ke website saya<sup>3</sup> untuk informasi lebih lengkap.

---

<sup>3</sup><http://software.endy.muhammadin.com/about>

### 2.11.3 Reference Link

Kita juga bisa memisahkan antara link dan URL yang dituju dengan format berikut:

```
[link label][id link]
```

Dengan cara di atas, URL yang dituju tidak langsung ditulis di sebelah labelnya. Kita harus sediakan referensi URL yang dituju di bagian lain dokumen (biasanya di akhir) seperti ini:

```
[id link]: http://url-yang-dituju.com "Judul Linknya"
```

Id link bisa dihilangkan, sehingga id link sama dengan labelnya, seperti ini:

```
Silahkan kunjungi [website saya]
```

Di akhir dokumen, sertakan referensinya:

```
[website saya]: http://software.endy.muhammad.com "Websitenya Endy"
```

### 2.11.4 Inline Link

Kita bisa membuat link ke bagian lain dokumen, misalnya heading. Seperti sudah kita bahas di bagian **Pemberian ID untuk Header**, pandoc akan membuat id untuk setiap heading yang kita definisikan.

Berikut contohnya:

```
Silahkan lihat bagian [Inline Formatting](#inline-formatting).
```

## 2.12 Image

Menampilkan image mirip dengan link. Bedanya, di depan kurung kotak kita berikan tanda seru. Contohnya menggunakan relative path seperti ini:

```
![Logo ArtiVisi](resources/logo-artivisi.png)
```

Contoh di atas akan menghasilkan tampilan seperti ini:



Gambar 2.1: Logo ArtiVisi

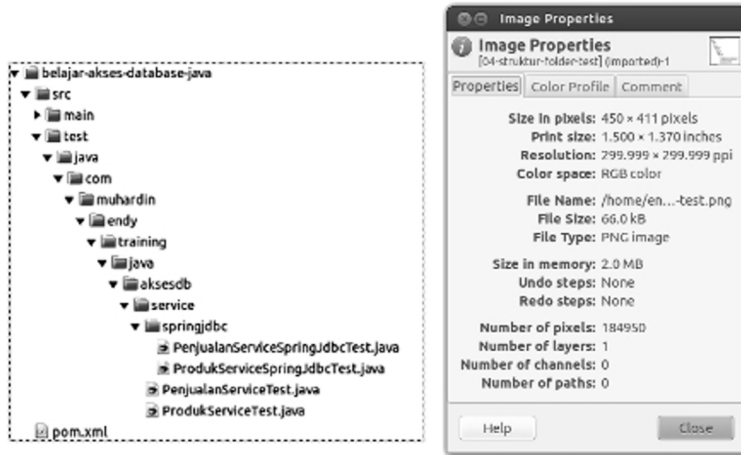
Bila kita perhatikan, gambar logo ArtiVisi di atas agak buram. Berbeda dengan logo ArtiVisi di halaman paling depan. Penyebabnya adalah karena resolusi gambar di atas hanya 90 dpi. Agar gambarnya tajam, resolusi idealnya adalah 300 dpi.

Hal ini perlu diperhatikan pada saat memasang screenshot. Umumnya aplikasi pengambilan screenshot akan menghasilkan gambar dengan resolusi 72 dpi, sehingga pasti akan terlihat lebih buram daripada logo di atas. Supaya gambar tampil dengan baik, kita perlu mengubah resolusinya menjadi 300 dpi.

Ukuran gambar juga perlu disesuaikan agar tidak terlalu lebar ataupun terlalu kecil pada saat dikonversi menjadi dokumen PDF. Untuk PDF berukuran A4 dengan layout buku bawaan  $\text{\LaTeX}$ , yaitu `\documentclass[a4]{book}`, ukuran gambar yang memenuhi satu halaman kira-kira 1200 pixel lebarnya dan 1500 pixel tingginya.

Resolusi dan ukuran gambar bisa diedit menggunakan aplikasi pengolah gambar seperti Gimp atau Photoshop. Tapi perlu diperhatikan bahwa memperbesar resolusi dan ukuran akan membuat gambar menjadi buram. Untuk itu kita perlu melakukan tindakan lain seperti *sharpening*, pengaturan kontras, dan lainnya.

Berikut tampilan screenshot yang diedit dalam Gimp.



Gambar 2.2: Setting Image dalam Gimp

Gambar di atas berukuran lebar 1200 pixel dan tinggi 741 pixel dengan resolusi 300dpi.

Uraian di atas menjelaskan pada kita bahwa screenshot yang kita ambil tidak bisa langsung dipasang begitu saja dalam dokumen. Kita harus melakukan pengeditan dulu dengan aplikasi pengolah gambar seperti Gimp atau Photoshop.

### 2.13 Catatan kaki

Catatan kaki ditulis seperti ini:

Tulisan ini ada catatan kakinya, `[^1]` dan ini juga. `[^longnote]`

`[^1]`: Catatan kaki nomer satu.

`[^longnote]`: Catatan kaki panjang multi baris.

Paragraf berikutnya diindentasi untuk menunjukkan bahwa dia bagian dari catatan kaki di atasnya.

```
{ contoh kode program }
```

Indentasi boleh di baris pertama saja atau di seluruh paragraf

Paragraf ini tidak termasuk catatan kaki, karena tidak diindentasi.

Contoh di atas akan menghasilkan tampilan sebagai berikut:

---

Tulisan ini ada catatan kakinya,<sup>4</sup> dan ini juga.<sup>5</sup>

Paragraf ini tidak termasuk catatan kaki, karena tidak diindentasi.

---

Catatan kaki perlu diberikan identifier, yaitu label berawalan `^` di dalam kurung kotak. Identifier ini tidak boleh mengandung spasi, tab, atau ganti baris. Identifier hanya digunakan untuk menghubungkan titik referensi dengan catatan kakinya. Sedangkan penomoran di outputnya akan dihitung otomatis.

## 2.14 Referensi dan Bibliografi

Pandoc juga mendukung referensi (citation) ke tulisan orang lain, seperti yang biasa dicantumkan sebagai daftar pustaka. Ini biasanya digunakan kalau kita membuat tulisan ilmiah seperti jurnal, skripsi, atau thesis.

Untuk lebih jelasnya, bisa dilihat di dokumentasi pandoc.

---

<sup>4</sup>Catatan kaki nomer satu.

<sup>5</sup>Catatan kaki panjang multi baris.

Paragraf berikutnya diindentasi untuk menunjukkan bahwa dia bagian dari catatan kaki di atasnya.

```
{ contoh kode program }
```

Indentasi boleh di baris pertama saja atau di seluruh paragraf



## BAB 3 | Output

---

Dokumen markdown yang sudah kita tulis bisa dikonversi menjadi berbagai format sesuai kebutuhan. Di antara format yang sering digunakan adalah:

- PDF : untuk dicetak atau dikirim melalui email
- HTML : untuk ditampilkan di website
- Slide Presentasi : sebetulnya format slide presentasi yang dihasilkan adalah HTML. Tapi karena formatnya spesifik dan berbeda, maka baiklah kita pisahkan.

### 3.1 Beberapa kesepakatan

#### 3.1.1 Lokasi eksekusi perintah

Proses konversi dilakukan dengan cara mengetik perintah pandoc di command prompt. Perintah ini dilakukan di lokasi folder tempat file yang akan diproses berada. Jika file yang akan diproses berada di folder lain, saya asumsikan pembaca sudah mengetahui cara penulisan path baik absolute ataupun relative<sup>1</sup>.

#### 3.1.2 Ekstensi File

Walaupun file markdown bisa diberikan ekstensi `.txt`, tapi kita akan menggunakan ekstensi `.md` supaya bisa ditampilkan dengan baik di text editor seperti Gedit atau Notepad++.

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Path\\_\(computing\)](http://en.wikipedia.org/wiki/Path_(computing))

## 3.2 PDF

### 3.2.1 Perintah Dasar

Untuk menghasilkan file PDF, perintahnya adalah sebagai berikut:

```
pandoc -o hasil.pdf *.md
```

Kita bisa menambahkan daftar isi (Table of Contents) dengan opsi `--toc`. Supaya bab dan sub-bab diberi nomer, sertakan opsi `-N`.

```
pandoc --toc -N -o hasil.pdf *.md
```

### 3.2.2 Menggunakan L<sup>A</sup>T<sub>E</sub>X Template

Khusus untuk dokumen ArtiVisi, biasanya saya sudah menyertakan template L<sup>A</sup>T<sub>E</sub>X tersendiri. Template ini memiliki opsi untuk mengganti font dan menaruh nomer versi. Kita membutuhkan engine Xetex agar template ini bisa digunakan. Berikut perintahnya (jalankan dalam satu baris):

```
pandoc \  
--template artivisi-template.tex \  
--variable mainfont="Droid Serif" \  
--variable sansfont="Droid Sans" \  
--variable fontsize=12pt \  
--variable version=1.0 \  
--latex-engine=xelatex --toc -N -o hasil.pdf *.md
```

### 3.2.3 Manual Page Break

Khusus PDF, biasanya kita menginginkan tampilan cetak yang rapi dan bagus. Layout otomatis yang disediakan L<sup>A</sup>T<sub>E</sub>X kadangkala kurang baik dalam mengatur penempatan gambar, sehingga seringkali gambar tampil meleset tidak sesuai dengan urutan tulisan yang disebabkan karena pengaturan *page break*. Untuk itu, kita membutuhkan pengaturan manual. Caranya adalah dengan menambahkan perintah L<sup>A</sup>T<sub>E</sub>X seperti ini:

```
Paragraf pertama  
\newpage  
Paragraf ini akan ditulis di halaman baru
```

Ini dijelaskan oleh John MacFarlane melalui email

There's no general way to force page breaks, by the way. If you just want page breaks in PDF (via latex), you can insert a raw latex command,

```
\newpage
```

This should be ignored in HTML and ODT output, so it will only affect latex and PDF via latex.

Perintah `\newpage` ini akan diabaikan bila kita menghasilkan output HTML dan ODT.

### 3.2.4 Link dan Catatan Kaki

Secara default, semua *hyperlink* yang kita tulis akan dibuat menjadi link yang bisa diklik di dokumen PDF yang dihasilkan. Hal ini bagus bila PDF tersebut hanya kita baca di komputer, kalau kita ingin membuka linknya, kita tinggal klik saja dan browser akan terbuka.

Tapi lain halnya bila kita ingin mencetak dokumen PDF tersebut. Semua link hanya akan menjadi tulisan berwarna biru. Oleh karena itu, kita ingin mengkonversi *hyperlink* menjadi catatan kaki. Ini bisa dilakukan dengan variabel `links-as-notes` seperti ini:

```
pandoc -V links-as-notes -o hasil.pdf *.md
```

### 3.2.5 Tanpa Warna

Adakalanya kita ingin hasil PDF yang tidak berwarna agar bisa dicetak tanpa tinta warna. Penggunaan warna di printer yang hitam-putih akan menyebabkan warna huruf menjadi tidak hitam sempurna sehingga dokumen sulit difotokopi. Untuk itu, kita bisa menambahkan opsi `--no-highlight` agar kode program tidak diwarnai. Selain kode program, hyperlink juga biasanya diwarnai biru. Kita bisa mengganti warna biru tersebut menjadi hitam melalui variabel `linkcolor` dan `urlcolor`. Berikut perintah lengkapnya

```
pandoc \  
-V urlcolor=black \  
-V linkcolor=black \  
--no-highlight \  
-o markdown-dan-pandoc-bw.pdf *.md
```

### 3.3 Slide Presentasi

Untuk menghasilkan slide presentasi, perintahnya adalah sebagai berikut:

```
pandoc -s --self-contained -t s5 -o slide.html menggunakan-pandoc.md
```

Perintah di atas akan menghasilkan slide presentasi yang sudah siap pakai, lengkap dengan warna dan setting font standar dari S5<sup>2</sup>.

Kadangkala kita ingin menggunakan theme lain seperti yang bisa didapatkan di S5 Reloaded<sup>3</sup>. Untuk itu, kita tidak memproduksi slide presentasi yang lengkap, tapi cukup HTMLnya saja supaya nanti bisa diganti theme nya sesuai keinginan.

Gunakan perintah berikut:

```
pandoc -s -t s5 -o slide.html menggunakan-pandoc.md
```

Setelah itu, file `slide.html` yang dihasilkan bisa diedit, isi tag `<head></head>` diganti dengan yang ada di dalam custom theme dari S5 Reloaded.

Supaya *bullet point* tampil satu persatu, kita perlu memberikan opsi `-i` yaitu incremental.

```
pandoc -s --self-contained -i -t s5 -o slide.html menggunakan-pandoc.md
```

### 3.4 Open Office

Untuk menghasilkan file OpenOffice Writer, perintahnya adalah sebagai berikut:

```
pandoc -o hasil.odt *md
```

Bila kita ingin menyesuaikan style, yaitu setting jenis huruf, ukuran huruf, jarak antar paragraf, dsb, kita bisa mengedit file `hasil.odt` tersebut dan menjadikannya template.

Setelah kita memiliki template, kita berikan kepada pandoc dengan opsi `--reference-odt` sebagai berikut:

```
pandoc --reference-odt=template.odt -o hasil.odt *md
```

<sup>2</sup>[http://en.wikipedia.org/wiki/S5\\_\(file\\_format\)](http://en.wikipedia.org/wiki/S5_(file_format))

<sup>3</sup><http://www.netzgesta.de/S5/>

## 3.5 HTML

Untuk menghasilkan output HTML, perintahnya sebagai berikut:

```
pandoc -o hasil.html *md
```

Seperti halnya PDF, kita juga bisa membuatkan daftar isi:

```
pandoc --toc -N -o hasil.html *md
```

Agar daftar isinya tampil, kita perlu memberikan opsi `-s` yaitu standalone. Artinya menghasilkan output yang komplit dengan halaman cover.

```
pandoc -s --toc -N -o hasil.html *md
```



## BAB 4 | Penutup

---

Demikianlah tutorial tentang format Markdown dan aplikasi Pandoc. Kritik, komentar, dan saran silahkan disampaikan melalui:

- Email : [endy.muhammadin@gmail.com](mailto:endy.muhammadin@gmail.com)<sup>1</sup>
- Twitter : [@endymuhammadin](https://twitter.com/endymuhammadin)<sup>2</sup>
- Yahoo Messenger : endymuhammadin

Untuk mengetahui update terbaru terhadap buku ini, silahkan watch repository Github<sup>3</sup>.

---

<sup>1</sup><mailto:endy.muhammadin@gmail.com>

<sup>2</sup><https://twitter.com/endymuhammadin>

<sup>3</sup><https://github.com/endymuhammadin/buku-pandoc>