

# *DISTRIBUTED VERSION CONTROL DAN OPEN FORMAT SEBAGAI SOLUSI PERMASALAHAN PADA TUGAS KOLABORATIF DI UNIVERSITAS.*

Azzam Syawqi Aziz

February 22, 2018

Kerja kelompok sudah menjadi napas para mahasiswa di kampus. Terlebih di fakultas yang memiliki mata kuliah dengan segudang proyek akhir. Saya sendiri terbiasa mengerjakan porsi lebih banyak daripada teman sekelompok. Keinginan untuk memahami materi lebih dalam dan kecintaan pada *computer science* membuat saya tidak terganggu, meskipun saya harus mengerjakan tugas atau proyek akhir seorang diri.

Seiring berjalannya waktu, beberapa hal ganjal menghampiri. Kali ini saya tak lagi tenang meskipun sudah terbiasa mengerjakan tugas akhir seorang diri. Semester lalu, saya melihat Andi\* selalu marah dan mengeluh di kelas. Ia bercerita bahwa teman sekelompoknya tidak pernah mau ikut andil dalam tugas kelompoknya. Bahkan mereka membalas ajakan untuk kerja kelompok secara acuh. Maklum saja karena memang tidak sedikit mahasiswa kampus yang lebih fokus pada hal-hal di luar kampus.

Berbeda dengan Andi yang berani menghapus nama teman yang tak aktif bekerja, Budi\* lebih memilih untuk diam. Saya selalu diliputi rasa iba tatkala bertemu dengannya di kelas. Kali ini teman-temannya tidak membantu bukan karena acuh, melainkan karena mereka tidak paham bagaimana cara mengerjakan tugas tersebut.

Begitu pun Ani\* yang secara lugas menjawab "sudah biasa", tatkala ditanya mengapa ia bekerja sendiri. Pada awalnya saya pikir hal-hal ini mungkin hanya terjadi pada beberapa teman saya saja. Tetapi munculnya banyak *post* di sosial media yang mengeluhkan bahwa teman mereka tidak pernah ikut bekerja atau hanya muncul ketika ditanya nomor induk mahasiswa, membuat saya merasa permasalahan ini harus memiliki solusi.

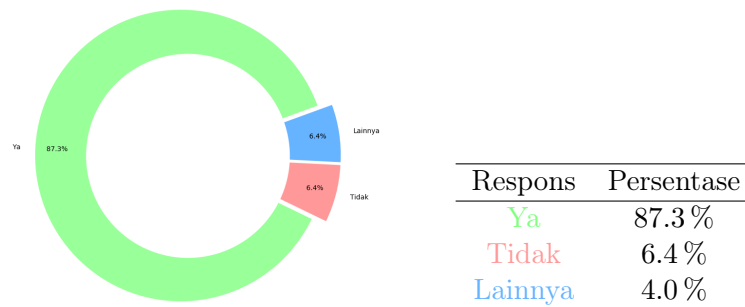
---

\*Nama Fiktif

## 1 Jumlah yang tidak sedikit

Hasil mengejutkan terlihat pada data yang dihimpun dari  $\pm 100$  responden dari total  $\pm 800$  populasi menghasilkan statistik berikut.

**Berapa kali anda mengerjakan tugas kelompok seorang diri ?**



**Figure 1 & Table 1:** berapa kali bekerja sendiri

Sebanyak 87 % dari responden pernah bekerja sendiri dalam proyek akhir mata kuliah tertentu yang seharusnya dikerjakan secara berkelompok. Artinya **jika diambil 10 orang secara acak dari 800 orang, maka setiap 10 orang ada 8 orang yang mengerjakan tugas kelompoknya seorang diri**. Diantaranya rata-rata pernah bekerja sendirian mulai 1 hingga lebih dari 5 kali.

Dengan kata lain, jika 800 dari populasi dikelompokkan dengan maksimal 5 orang, maka terdapat total 160 kelompok, **sehingga terdapat  $\pm 139$  kelompok dari 160, dimana di dalamnya ada seorang anak yang mengerjakan tanpa bantuan yang lain**. Pada data ini terdapat kemungkinan seseorang bekerja pada tim A sendirian, namun tidak aktif di tim B.

Ada hal tak biasa yang saya temukan pada bagian "Apakah ada yang ingin anda sampaikan ?". Tidak sedikit diantara mereka yang meminta tolong dan berharap ada solusi untuk masalah ini. Tidak jarang pula ada yang mendukung dan memberikan semangat dengan harapan riset kecil ini dapat membuahkan hasil.

**Apakah dosen anda mengetahui bahwa anda bekerja sendiri ?**

Tentu para mahasiswa tidak mungkin menyampaikan hal ini kepada dosen, dengan berbagai macam alasan. Harapan kedepannya, data yang dihasilkan dari alur kerja essay ini dapat membantu dosen memberikan hak kepada mereka yang berhak, serta membangun suasana yang sehat dan sportif di lingkungan Universitas.

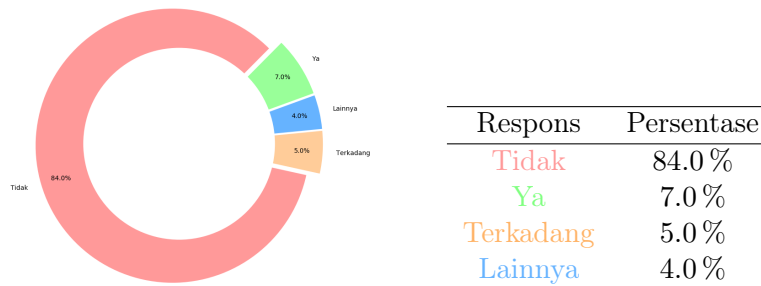


Figure 2 & Table 2: apakah dosen tau

## 2 *Version control*[1] dan *Open format*[2].

Kedua alat ini akan memberikan banyak jawaban terhadap permasalahan kerja kelompok yang kerap kali terjadi, ditinjau dari segi popularitas *version control* yang kerap kali digunakan adalah git, dan  $\text{\LaTeX}$  atau Markdown untuk *open format*.

Penggunaan *open format* membuat git lebih mudah untuk melacak perubahan, melihat statistik kontribusi, sejarah perubahan, maupun waktu perubahan berkas. Singkatnya git membantu kita dalam menjawab **oleh siapa dan kapan**.

Awalnya mahasiswa dituntut untuk mempelajari metodologi dari kedua alat tersebut, mempelajari alat baru tentu membutuhkan investasi waktu, senada dengan apa yang disampaikan Miyamoto Musashi[3] bahwa semua hal pada awalnya memang sulit. Namun, penguasaan teknis *version control* sudah seharusnya dikuasai oleh mahasiswa ilmu komputer[4].

Tanpa kita sadari, kini git tak hanya terbatas penggunaannya pada civitas akademika ilmu komputer dan para pengembang perangkat lunak untuk melakukan kolaborasi, begitu pun  $\text{\LaTeX}$  yang sudah dikuasai oleh banyak elemen masyarakat untuk menulis karya ilmiah dan berbagai hal lain, penggunaan bahasa markah seperti Markdown sudah marak digunakan untuk berkolaborasi di wiki, berkomentar di sosial media, maupun untuk menulis artikel *blog*.

## 3 Penggunaan Alat

Git memiliki banyak fitur, tetapi hanya sebagian fitur yang akan digunakan pada penyelesaian permasalahan ini.

Untuk melihat banyaknya statistik kontribusi seseorang pada sebuah berkas, kita dapat melakukannya dengan cara:

---

```
1 git shortlog -sn --no-merges
```

---

```
$ git shortlog -sn --no-merges
113 vindarel
24 Cody Reichert
10 Koz Ross
8 azzamsa
7 ReadmeCritic
6 Sebastian Christ
3 Claire Medeiros
2 Jean-Philippe Paradis
2 Michael Malis
1 Anwesh Nayak
1 John Mercouris
1 Kay Z
1 Mario Rodas
1 Michał Herda
1 Ric da Silva
1 Svante
1 ryn1x
```

Figure 3: git shortlog tanpa merge

```
$ g l
ae9dcb2 (HEAD -> master, origin/master, origin/HEAD) 6 weeks ago add simplified CLHS (azzamsa)
18874c4 (upstream/master) 6 weeks ago Merge pull request #132 from vindarel/compilers (vindarel)
9c519a2 6 weeks ago Merge pull request #137 from vindarel/cryptocurrencies (vindarel)
e9d814b 6 weeks ago Merge pull request #136 from vindarel/datastructures (vindarel)
2f55feb 6 weeks ago Added Casting SPELs in LISP online tutorial (#139) (ryn1x)
3b587c9 6 weeks ago add sycamore, purely functional data structure (vindarel)
2ab5099 6 weeks ago more data structures (BST, binary heap,...) (vindarel)
52e4715 6 weeks ago add C, C++ and JS compilers/code generators (vindarel)
fd94ae3 6 weeks ago lisp-machine has disappeared, give cl-devel2 docker container instead (vindarel)
799449f 6 weeks ago Merge pull request #131 from vindarel/defclass/std (vindarel)
9734e3f 6 weeks ago Merge pull request #127 from jmercouris/master (vindarel)
cdeb935 6 weeks ago link to awesome *software* lists. (vindarel)
5090ab3 6 weeks ago add defclass-std, shortcut macro (vindarel)
83a1202 6 weeks ago add 4 cryptocurrencies libs (by Ironclad's maintainer and Monero contributor) (vindarel)
57e6d02 6 weeks ago Ironclad is now maintained at sharplispers (vindarel)
9b90705 6 weeks ago Merge pull request #128 from vindarel/screamer (vindarel)
434fe1b 6 weeks ago add Screamer, Screamer+, blog post. Logic, constraint programming. (vindarel)
f752333 6 weeks ago (minor) move Language Extensions in own section (vindarel)
939db2c 6 weeks ago add nEXT Browser (John Mercouris)
eca8894 6 weeks ago Fix typo. ("Language" -> "Language") (Jean-Philippe Paradis)
d984c60 6 weeks ago Merge pull request #124 from Hexstream/master (vindarel)
2beb188 6 weeks ago Add modern public domain online CLOS MOP specification. (Jean-Philippe Paradis)
697d8a5 6 weeks ago (minor) fix badly sorted subsection (vindarel)
e04d531 6 weeks ago (minor) fix hanchor (vindarel)
b746ff0 6 weeks ago Merge pull request #123 from vindarel/sblint (vindarel)
005138a 6 weeks ago add sblint, source code linter suited for Reviewdog (vindarel)
28a41bd 6 weeks ago (minor) utilities - reorder sub-sections (vindarel)
```

Figure 4: git log

Melihat log perubahan berkas berdasarkan waktu. (Figure 4)

---

```
1 git log --pretty=custom
```

---

Melihat siapa penulis pada baris tertentu. (Figure 5)

---

```
1 git blame foo.txt
```

---

Git sangat populer sehingga banyak fitur dari git di dukung oleh alat-alat lain, seperti menggunakan magit di GNU Emacs. (Figure 6)

---

```
1 M-x magit-blame
```

---

```
$ git blame README.md
20f49f32 (Cody Reichert      2015-10-30 13:17:24 -0500   1) <div align="center">
20f49f32 (Cody Reichert      2015-10-30 13:17:24 -0500   2)   
20f49f32 (Cody Reichert      2015-10-30 13:17:24 -0500   3) </div>
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500   4)
f5d62d2c (Cody Reichert      2017-04-21 12:51:37 -0500   5) # Awesome Common Lisp [[Awesome]](https://cdn.rawgit.com/sindresorhus/a
d8e8829/media/badge.svg)](https://github.com/sindresorhus/awesome) [[Assertible status]](https://assertible.com/apis/102e134d-f9a8-45b
5a28683832efffb)](https://assertible.com/docs/guide/deployments)
*bb7fe97 (Koz Ross          2014-12-29 12:04:07 +1300   6)
cdeb9359 (vindarel          2018-01-08 11:25:09 +0100   7) A curated list of _awesome_ Common Lisp libraries.
cdeb9359 (vindarel          2018-01-08 11:25:09 +0100   8)
cdeb9359 (vindarel          2018-01-08 11:25:09 +0100   9) For awesome *software*, see [lisp-lang.org's success stories](http://li
l-software)](https://github.com/azzamsa/awesome-cl-software) list.
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  10)
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  11) Add something new! See the [contributing](#contributing) section for ad
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  12) list.
5dc2a9b4 (Cody Reichert      2015-08-12 18:30:01 -0500  13)
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  14) All libraries listed here are available from [Quicklisp][16] unless
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  15) stated otherwise.
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  16)
433860a9 (Cody Reichert      2015-08-15 23:11:20 -0500  17) This is released under the GNU Free Documentation License - its text
433860a9 (Cody Reichert      2015-08-15 23:11:20 -0500  18) is provided in the LICENSE file. This repository is also mirrored on
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  19) [NotABug](https://notabug.org/CodyReichert/awesome-cl) - a **fully-free
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  20) alternative to Github. Preference is given to [free software][13] and
757a9763 (Cody Reichert      2015-10-30 13:13:00 -0500  21) sellers who aren't evil for physical resources.
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  22)
419476b8 (vindarel          2016-12-21 18:23:08 +0100  23) <!-- markdown-toc start - Don't edit this section. Run M-x markdown-toc
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  24) **Table of Contents**
419476b8 (vindarel          2016-12-21 18:23:08 +0100  25)
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  26) - [Build Systems](#build-systems)
52e47150 (vindarel          2018-01-08 11:50:32 +0100  27) - [Compilers, code generators](#compilers-code-generators)
52e47150 (vindarel          2018-01-08 11:50:32 +0100  28)   - [C, C++(C-c)]
52e47150 (vindarel          2018-01-08 11:50:32 +0100  29)   - [JavaScript](#javascript)
05af5b68 (Cody Reichert      2015-08-12 20:44:01 -0500  30)   - [Crypto](#crypto)
83a12025 (vindarel          2018-01-06 21:11:25 +0100  31) - [Cryptocurrencies](#cryptocurrencies)
```

Figure 5: git blame

```
Cody Reichert      2015-10-30 20:17 Spacing fix in README
<div align="center">
  
</div>
Cody Reichert      2015-10-30 20:13 Update README.md

Cody Reichert      2017-04-21 19:51 Update Assertible status badge
# Awesome Common Lisp [[Awesome]](https://cdn.rawgit.com/sindresorhus/awesome/d730f38d29fed78fa85652e3a63e154dd8e8829/media/badge.svg)](https://github
Koz Ross          2014-12-29 19:04 Initial commit

vindarel          2018-01-08 18:25 link to awesome *software* lists.
A curated list of _awesome_ Common Lisp libraries.

For awesome *software*, see [lisp-lang.org's success stories](http://lisp-lang.org/success/) and the [awesome-cl-software](https://github.com/azzamsa/
Cody Reichert      2015-10-30 20:13 Update README.md

Add something new! See the [contributing](#contributing) section for adding something to the
list.
Cody Reichert      2015-08-13 01:30 Mirror list from NotABug to GH

Cody Reichert      2015-08-13 03:44 Add table of contents
All libraries listed here are available from [Quicklisp][16] unless
stated otherwise.

Cody Reichert      2015-08-16 06:11 Add contributing section to readme
This is released under the GNU Free Documentation License - its text
is provided in the LICENSE file. This repository is also mirrored on
Cody Reichert      2015-10-30 20:13 Update README.md
[NotABug](https://notabug.org/CodyReichert/awesome-cl) - a **fully-free** (as in libre)
alternative to Github. Preference is given to [free software][13] and
sellers who aren't evil for physical resources.
Cody Reichert      2015-08-13 03:44 Add table of contents
```

Figure 6: magit-blame

```
$ git log --shortstat --author="azzamsa" | grep -E "fil(e|es) changed" \
→ | awk '{files+=$1; inserted+=$4; deleted+=$6} END {print "files changed : ", \
→ files, "\n" "lines inserted: ", inserted, "\n" "lines deleted : ", deleted }'
```

```
files changed : 8
lines inserted: 14
lines deleted : 5
```

**Figure 7:** git dengan alat Unix

Banyak hal yang bisa kita lihat dari data yang dimiliki git hanya menggunakan git ataupun dengan kombinasi alat-alat di lingkungan GNU/Linux, seperti untuk melihat jumlah penghapusan dan penambahan baris. (Figure 7)

---

```
1 git log --shortstat --author="azzamsa" | grep -E "fil(e|es)
↪ changed" | awk -f lines.awk
```

---



---

```
1 % lines.awk
2 {
3     files+=$1;
4     inserted+=$4;
5     deleted+=$6;
6 }
7 END {
8     print "files changed : ", files
9     print "lines inserted: ", inserted
10    print "lines deleted : ", deleted
11 }
```

---

Alat-alat di atas akan lebih *powerfull* jika dikombinasikan dengan alat lainnya, seperti `git_stats`, `gitstats`, atau `gitinspector`. Selain memiliki fitur-fitur yang baik untuk membuat laporan statistik, semua alat yang disebutkan di atas berlisensi bebas[5].

Kali ini saya hanya akan memperagakan penggunaan `gitinspector`, dengan berkas dari *repository* `awesome-cl`.

Menggunakan `gitinspector` dengan parameter `-r` untuk menampilkan seberapa besar tanggung jawab (*resposible*) seorang pengarang terhadap suatu berkas. (Figure 8)

```
gitinspector --file-types=md --format=html -r > hasil.html
```

Menampilkan pekerjaan menurut waktu. (Figure 9)

```
gitinspector --file-types=md --format=html -r -T
↪ --since=2017-01-01 > hasil.html
```

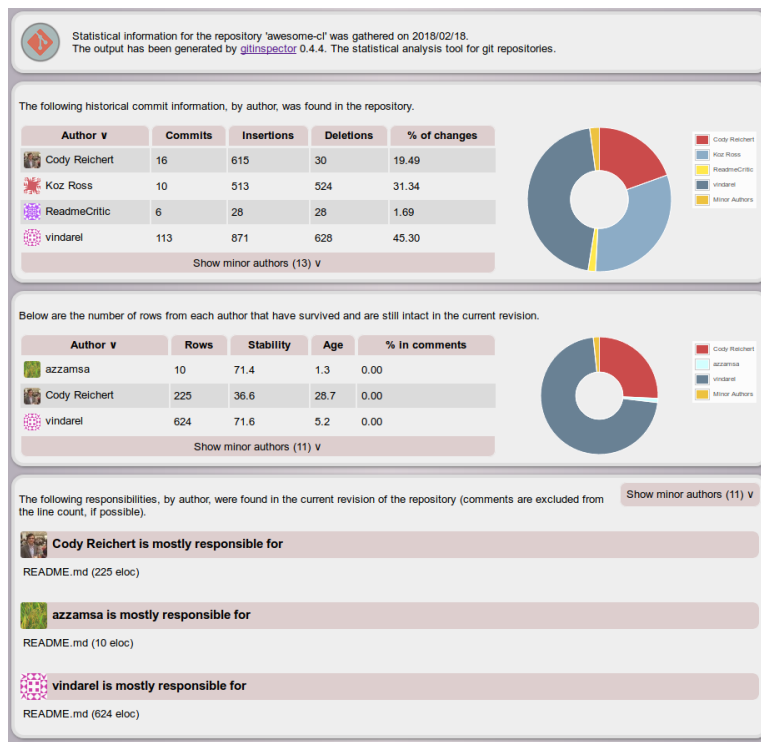


Figure 8: gitinspector responsible



Figure 9: gitinspector dengan *history*

Berkas yang dihasilkan gitinspector berbentuk HTML, JSON, atau pun plain text. HTML yang dihasilkan mengandung javascript, sehingga kontennya dinamis. Kita dapat melakukan perubahan urutan dengan melakukan klik pada *header* setiap kolom. Masih banyak fitur lain yang dimiliki gitinspector[6], tetapi tentunya tidak cukup untuk saya tampilkan semua di sini.

## 4 Alat yang lebih mudah

L<sup>A</sup>T<sub>E</sub>X memiliki beragam pengolah berkas berlisensi bebas yang mudah digunakan, seperti texmaker dan texpad. Begitu pun dengan git yang memiliki banyak dukungan pihak ketiga yang memudahkan penggunaannya, beberapa diantaranya disebutkan langsung oleh pengembang git di daftar git GUI clients[7].

Markdown dapat digunakan sebagai alternatif, jika L<sup>A</sup>T<sub>E</sub>X dirasa menelan waktu lebih lama untuk diadopsi, proses ekspor Markdown kedalam bentuk PDF dapat dilakukan dengan pandoc.

## 5 Timbulnya banyak manfaat lain

Jika penggunaan *open/libre format* serta *version control* sudah menjadi budaya di sebuah lingkungan Universitas, hal ini dapat memberikan banyak manfaat lain, seperti:

- Penulisan tugas-tugas besar seperti skripsi maupun proyek akhir sudah tidak lagi menggunakan *versioning manual* seperti “tugas 1 fix, tugas 1 fix sekali, tugas 1 final”.
- Dosen dapat dengan mudah memastikan orisinalitas pekerjaan mahasiswa dengan melihat *log*, *commit sign*[8], dan bertanya melalui *history* perubahan-perubahan yang dilakukan. Sehingga orisinalitasnya dapat dengan mudah dilacak dengan membaca pola pikir mahasiswa dalam melakukan perubahan-perubahan tersebut.
- Penggunaan *Open/libre format* dapat mengangkat wibawa dan nama Universitas. Karena sudah seharusnya Universitas mengajarkan kepada mahasiswa tentang berkarya dan berkolaborasi. Dengan mengajarkan mahasiswa tentang perangkat lunak bebas, Universitas dapat mencetak lulusan yang siap terjun dalam masyarakat digital yang bebas. Ini akan membantu masyarakat secara keseluruhan untuk keluar dari dominasi oleh perusahaan-perusahaan besar[9]. Stallman juga menambahkan bahwa Universitas seharusnya mendukung penggunaan perangkat lunak bebas. Karena alat-alat tersebut berkontribusi untuk ilmu pengetahuan, layaknya universitas mendorong para mahasiswa dan ilmuwan untuk mempublikasikan karya mereka.



- Universitas terbebas dari *vendor lock-in*, dan memberikan kebebasan kepada mahasiswa untuk menggunakan alat apa pun, tanpa harus menggunakan alat *proprietary*[10] tertentu. Mahasiswa tidak lagi dibatasi dengan proses pengumpulan dengan ekstensi tertentu dan versi tertentu.
- Mahasiswa ilmu komputer yang memiliki lingkungan kampus dengan penggunaan alat-alat bebas, membuat mahasiswa termotivasi untuk mempelajari alat-alat bebas lainnya, membaca kode sumbernya, dan mempelajari integrasi antara suatu alat dengan alat yang lain.

Beberapa hal di atas tentunya hanyalah sebagian dari banyak manfaat lainnya, jika suatu Universitas memiliki lingkungan belajar yang baik dan kolaborasi antar mahasiswa yang tinggi, tentu para mahasiswa akan termotivasi untuk selalu mempelajari bagaimana sesuatu bekerja. Perangkat lunak bebas mewadahi mereka untuk mempelajari cara kerja alat-alat tersebut. Sehingga motivasi mereka untuk belajar, berkarya dan berkontribusi kepada masyarakat semakin tinggi.

## 6 Harapan

Dengan menggunakan alat-alat di atas, saya berharap teman-teman yang sebelumnya banyak keteteran dalam mengerjakan tugas karena tidak mendapat bantuan sedikit pun dari kelompoknya, teman-teman yang seringkali meminta izin untuk telat mengumpulkan karena mengerjakan sendiri, teman-teman yang selalu sayu wajahnya karena temannya aktif berteriak lantang di luar kelas tetapi acuh dengan tanggung jawabnya di kelas, bisa mendapatkan haknya. Karena kita harus memberikan hak kepada orang-orang yang berhak, ucap Salman Al-farisi[11].

Penilaian para dosen berbeda-beda. Jika alur kerja menggunakan alat di atas dijadikan standar, maka dosen dengan proses penilaian spesifik dan berdasar pada jumlah kontribusi, bisa menjadikannya sebagai acuan. Sebaliknya dosen yang memberikan nilai yang sama pada setiap anggota, terlepas ikut andil atau tidaknya seseorang pada suatu kelompok, dapat mengabaikan data yang ada, meskipun hal ini tentunya tidak kita harapkan. Kita tidak ingin lagi ada teman-teman yang merasa tidak mendapat keadilan.

Saya berharap data yang dihasilkan tidak untuk menjerumuskan teman-teman yang jarang berkontribusi dan menjadikan raja bagi para mahasiswa yang mencintai bidang keilmuannya, tetapi bisa bersama membangun, belajar, berbagi, berkarya, saling menolong, saling mengajak dan mengajari.

Saya yakin dengan adanya hal ini, para mahasiswa yang belum memiliki *inner motivation* akan terpaksa pada awalnya untuk mengikuti standar yang ada, dan bersyukur di kemudian hari atas ilmu yang didapat.

Semoga alur kerja ini bisa digunakan dan bermanfaat untuk banyak kalangan, dan semakin meningkat seiring berjalannya waktu.

## 7 *Credit*

- Donald Knuth dan kotributor T<sub>E</sub>X.
- Leslie Lamport dan kontributor L<sup>A</sup>T<sub>E</sub>X.
- Linus Torvalds, Junio Hamano dan kontributor git.
- John Gruber dan Aaron Swartz sebagai pengembang Markdown.
- John MacFarlane dan kontributor sebagai pengembang pandoc.
- Adam Waldenberg dan kontributor sebagai pengembang gitinspector.
- Marius Vollmer, Jonas Bernoulli dan kontributor sebagai pengembang Magit.

## References

- [1] Distributed version control. (n.d.). Didapat Februari 20, 2018, dari [https://en.wikipedia.org/wiki/Distributed\\_version\\_control](https://en.wikipedia.org/wiki/Distributed_version_control)
- [2] List of open formats. (n.d.). Didapat Februari 20, 2018, dari [https://en.wikipedia.org/wiki/List\\_of\\_open\\_formats](https://en.wikipedia.org/wiki/List_of_open_formats)
- [3] Miyamoto Musashi. (n.d.). Didapat Februari 20, 2018 dari [https://en.wikipedia.org/wiki/Miyamoto\\_Musashi](https://en.wikipedia.org/wiki/Miyamoto_Musashi)
- [4] Might, M. (n.d.). What every computer science major should know. Didapat Februari 20, 2018 dari <http://matt.might.net/articles/what-cs-majors-should-know/>
- [5] Free software license. (n.d.). Didapat Februari 20, 2018, dari [https://en.wikipedia.org/wiki/Free\\_software\\_license](https://en.wikipedia.org/wiki/Free_software_license)
- [6] Gitinspector Docs. (n.d.). Didapat Februari 20, 2018. dari <https://github.com/ejwa/gitinspector/blob/master/docs/gitinspector.pdf>
- [7] git GUI clients. (n.d.). Didapat Februari 20, 2018. dari <https://git-scm.com/download/gui/linux>
- [8] Straub, B dan Chacon S. (2014). *Pro Git* (pp. 272-276). New York City, NY USA: Apress.

- [9] Stallman, R. (2003). *Free Software, Free Society: Selected Essays of Richard M. Stallman* (pp. 57-58). Boston, MA USA: GNU Press.
- [10] Proprietary software. (n.d.). Didapat Februari 20, 2018 dari [https://en.wikipedia.org/wiki/Proprietary\\_software](https://en.wikipedia.org/wiki/Proprietary_software)
- [11] Salman Al-farisi. (n.d.). Didapat Februari 20, 2018. dari [https://en.wikipedia.org/wiki/Salman\\_the\\_Persian](https://en.wikipedia.org/wiki/Salman_the_Persian)