

## **PENGARUH PENGGUNAAN *VERSION CONTROL SYSTEM* TERHADAP PROSES BELAJAR PEMROGRAMAN MAHASISWA**

**Indra Azimi**

Program Studi Rekayasa Perangkat Lunak Aplikasi, Universitas Telkom  
Bandung, Indonesia  
e-mail: indraazimi@tass.telkomuniversity.ac.id

### **Abstrak**

Kondisi sumber daya manusia Indonesia di bidang Teknologi Informasi dan Komunikasi, khususnya pengembangan perangkat lunak dan pemrograman cukup mengkhawatirkan. Salah satu penyebabnya adalah tidak seimbangnya permintaan dari industri dengan lulusan yang disediakan oleh perguruan tinggi. Oleh sebab itu, perguruan tinggi perlu meningkatkan proses belajar dengan menggunakan teknologi yang telah umum digunakan oleh industri, yaitu *version control system* (VCS). Berdasarkan penelitian yang dilakukan terhadap 89 mahasiswa Rekayasa Perangkat Lunak Aplikasi, Universitas Telkom, didapat hasil bahwa penggunaan VCS terbukti memberikan pengaruh yang positif terhadap proses belajar pemrograman mahasiswa dalam pengembangan sebuah perangkat lunak, diantaranya kemudahan berkolaborasi dalam tim, meningkatnya motivasi *coding* mahasiswa dan proses penilaian yang menjadi lebih transparan karena kontribusi yang terukur.

**Kata kunci:** *version control system*, belajar pemrograman

### **Abstract**

*The condition of Indonesia's human resources in the field of Information and Communication Technology, especially software development and programming is quite alarming. One reason is the imbalance in demand from industry with graduates provided by universities. Therefore, universities need to improve the learning process by using technology that has been commonly used by industry, namely the version control system (VCS). Based on research conducted on 89 students of Application Software Engineering, Telkom University, the results showed that the use of VCS proved to have a positive influence on the learning process of student programming in software development, including ease of collaboration in teams, increased student coding motivation and assessment processes that become more transparent because of measurable contributions.*

**Keywords:** *version control system*, learn programming

## PENDAHULUAN

Berdasarkan data dari peta okupasi nasional, kondisi sumber daya manusia Indonesia di bidang Teknologi Informasi dan Komunikasi (TIK) cukup mengkhawatirkan (Kemkominfo, 2017). Dari 16 fungsi kunci bidang TIK dalam Kerangka Kualifikasi Nasional Indonesia (Kemkominfo, 2018), fungsi Pengembangan Perangkat Lunak dan Pemrograman adalah yang paling minim pemenuhannya.

Minimnya pemenuhan sumber daya manusia ini salah satunya disebabkan oleh tidak seimbangnya permintaan dari industri dengan lulusan yang disediakan oleh perguruan tinggi, baik dalam hal kuantitas maupun kualitasnya.

Industri dalam negeri yang terkait TIK berkembang sangat pesat di tahun-tahun terakhir. Hal ini ditandai dengan banyaknya bermunculan *startup-startup* baru. Bahkan sudah ada 4 *startup* Indonesia yang menjadi *unicorn* (Suroso, 2018), yaitu startup dengan valuasi lebih dari satu miliar dolar (Kanbach, 2016). Kondisi ini otomatis memerlukan sumber daya manusia yang tidak hanya banyak dalam jumlah, tapi juga dapat diandalkan.

Di sisi lain, menjamurnya program studi terkait TIK dalam 10-15 tahun terakhir, baik yang baru dibuat karena melihat peluang ini maupun yang sudah ada terlebih dahulu, ternyata masih belum mampu memenuhi kebutuhan industri TIK.

Berdasarkan data dari Kemenristek Dikti, saat ini terdapat lebih dari 1.250 program studi terkait TIK di Indonesia (Kemenristek Dikti, 2019). Program studi ini berjenjang, mulai dari tingkat diploma, sarjana, master hingga doktor. Fokus dari program studi ini pun beragam, mulai dari teknik informatika, manajemen informatika hingga rekayasa perangkat lunak.

Untuk mengatasi masalah minimnya sumber daya manusia Indonesia di bidang TIK di atas, diperlukan sebuah upaya dari perguruan tinggi untuk dapat meningkatkan lagi efektivitas proses belajar mahasiswa, khususnya dalam bidang pemrograman dan pengembangan perangkat lunak. Proses belajar pemrograman yang efektif ini pada akhirnya diharapkan dapat meningkatkan kemampuan mahasiswa yang bersangkutan, sehingga mereka berpeluang besar lulus tepat pada waktunya dan lebih cepat diserap oleh industri.

Salah satu cara untuk meningkatkan efektivitas proses belajar mahasiswa adalah dengan mengintegrasikan penggunaan teknologi dalam proses belajar mengajar. Sebagai contoh, penggunaan teknologi AR (*augmented reality*) untuk proses belajar geometri solid di Taiwan terbukti dapat meningkatkan persepsi spasial siswa-siswanya (Lin, 2015).

Teknologi yang diintegrasikan ke dalam proses belajar mengajar sedapatnya harus merupakan teknologi yang umum digunakan di industri. Hal ini selain akan sangat membantu mahasiswa di tempat kerjanya nanti setelah mereka lulus kuliah, juga akan memberikan motivasi lebih bagi mahasiswa tersebut untuk mempelajari teknologi itu (Forsyth, 2016).

Di bidang pemrograman sendiri, salah satu teknologi yang sering dipakai oleh *developer* di perusahaan dalam melakukan pekerjaan mereka adalah *version control system* atau disingkat VCS (Hogblom, 2015). CVS digunakan oleh para *developer* sebagai alat untuk manajemen *source code* mereka dan memungkinkan untuk berkolaborasi dengan *developer* lain dalam tim dengan lebih mudah (Majumdar, 2017). Dengan VCS, *developer* juga dapat melihat riwayat perubahan yang dilakukan pada *source code* aplikasi.

Oleh sebab itu, menjadi menarik untuk mengetahui pengaruh penggunaan VCS ini terhadap proses belajar pemrograman mahasiswa di perguruan tinggi, khususnya dalam pengembangan perangkat lunak. Pembahasan penelitian ini akan dimulai dari metode dari penelitian yang dilakukan, analisis hasil dan pembahasan serta diakhiri dengan suatu kesimpulan dan saran.

## METODE PENELITIAN

Penelitian ini dilakukan terhadap mahasiswa yang mengambil mata kuliah Pemrograman untuk Perangkat Bergerak Lanjut (PPBL) di Program Studi Rekayasa Perangkat Lunak Aplikasi, Fakultas Ilmu Terapan, Universitas Telkom. Mata kuliah PPBL ini berada di semester lima dan merupakan lanjutan dari mata kuliah Pemrograman untuk Perangkat Bergerak Dasar.

Di mata kuliah PPBL ini mahasiswa diajarkan bagaimana cara mengembangkan aplikasi berbasis Android. Mulai dari *responsive design* di *handphone* dan *tablet*, penggunaan

*service* dan *broadcast receiver*, penggunaan *library* untuk mempermudah pekerjaan mereka, penggunaan *automated testing* untuk pengujian otomatis di berbagai *device* hingga cara merilis aplikasi ke Google Play Store.

Terdapat sebanyak 89 mahasiswa yang diikutsertakan dalam penelitian ini. Mahasiswa-mahasiswa tersebut berasal dari 3 kelas yang berbeda di angkatan yang sama. Berdasarkan survei yang dilakukan, dari 89 orang mahasiswa tadi, terdapat sebanyak 80,9% yang belum pernah menggunakan *version control system* (VCS) sebelumnya. Oleh karena jumlah mereka mayoritas, agar penelitian ini dapat dilanjutkan, maka diadakanlah pembekalan penggunaan VCS terlebih dahulu.

Dari banyak pilihan VCS yang tersedia, VCS yang dikenalkan kepada mahasiswa adalah Git. Git dipilih karena merupakan salah satu VCS yang paling populer (Davis, 2015). Selain itu, Git juga merupakan VCS yang bisa digunakan secara gratis, dan berbasis *open source* (Leonard, 2017). Adapun *hosting client* yang digunakan adalah Github, repositori *source code* terbesar di dunia (Borges, 2016).

Pembekalan penggunaan VCS ini diadakan selama 4 jam yang terbagi menjadi 2 sesi. Sesi pertama berisi pemaparan teori mengenai VCS, mulai dari *version control overview*, *version control workflow*, contoh *commit* dan pesan yang baik, serta contoh *push* dan *pull*. Adapun sesi kedua berupa praktek dari teori yang disampaikan di sesi pertama. Contoh kasus yang dipraktikkan yaitu berupa penggunaan VCS di Android Studio untuk pengerjaan sebuah aplikasi dalam tim.

Setelah mahasiswa memahami dan dapat menggunakan VCS, mahasiswa diberi sebuah tugas besar. Tugas mereka adalah mengembangkan sebuah aplikasi Android yang dalam pengerjaannya mereka wajib menggunakan VCS. Agar mahasiswa dapat belajar berkolaborasi dengan orang lain dalam tim sebagaimana di dunia kerja nanti, tugas ini dikerjakan secara berkelompok dengan anggota 2-3 orang.

Batas pengumpulan tugas besar ini adalah ketika perkuliahan berakhir, sekitar 13 minggu setelah pembekalan penggunaan VCS. Untuk menjaga kualitas dari aplikasi yang dikembangkan, dilakukan dua kali presentasi kemajuan (*progress*) aplikasi. Presentasi pertama

diadakan 3 minggu setelah pemberian tugas besar. Adapun presentasi kemajuan yang kedua diadakan 5 minggu setelah presentasi kemajuan yang pertama.

Setiap presentasi kemajuan dinilai. Nilai tim bergantung kepada *progress* dari aplikasi yang dikembangkan, dan nilai individu setiap mahasiswa dalam tim bergantung kepada kontribusi yang mereka lakukan di repositori Github. Kontribusi ini dilihat dari beberapa indikator antara lain seperti berapa jumlah *commit* yang di-*push* ke repositori Github, kapan *commit* dilakukan, berapa banyak kode yang ditambahkan dan dihapus.

Di akhir masa perkuliahan, dilakukan presentasi akhir aplikasi yang dikembangkan oleh mahasiswa. Proses penilaian yang dilakukan kurang lebih sama seperti saat presentasi kemajuan pertama dan kedua. Setelah semua mahasiswa menyelesaikan presentasi akhirnya, mereka diminta untuk mengisi kuesioner mengenai VCS yang mereka gunakan, yaitu Git dan Github.

Terdapat 7 pertanyaan yang diajukan kepada mahasiswa dalam kuesioner yang diberikan. Empat pertanyaan menggunakan skala Likert, sebuah alat psikometrik yang sering digunakan dalam penelitian ilmu sosial dan pendidikan (Joshi, 2015). Dua pertanyaan berupa pertanyaan ya/tidak, dan satu pertanyaan terakhir berupa pertanyaan terbuka.

Pengisian kuesioner ini dilakukan secara anonim, sehingga diharapkan mahasiswa akan merasa nyaman dalam menyampaikan pendapat murni mereka. Isi kuesioner mahasiswa ini selanjutnya akan dipakai untuk menilai pengaruh penggunaan VCS terhadap proses belajar pemrograman mereka dalam pengembangan perangkat lunak.

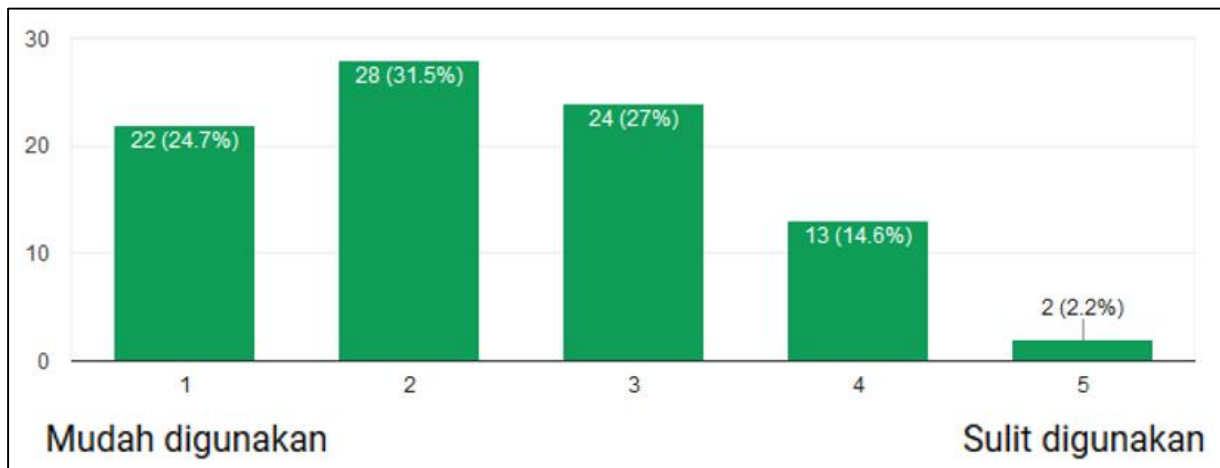
## HASIL PENELITIAN DAN PEMBAHASAN

Ketika sebuah teknologi dianggap mudah untuk digunakan, penggunaannya cenderung meyakini bahwa teknologi itu berguna dan akan mengadopsi teknologi tersebut (Kim, 2015). Oleh sebab itu, hal pertama yang diukur dalam penelitian ini adalah seberapa besar tingkat kemudahan penggunaan VCS (Git dan Github) bagi mahasiswa.

Berdasarkan survei yang dilakukan, mayoritas mahasiswa (56,18%) menyatakan

bahwa Git dan Github mudah digunakan. 26,97% menyatakan netral, tidak mudah dan juga tidak sulit. Sisanya sebesar 16,85% berpendapat bahwa menggunakan Git dan Github sulit. Distribusi jawaban dari mahasiswa yang lebih detail dapat dilihat pada Gambar 1.

Perlu diingat bahwa hasil ini didapat setelah mahasiswa tersebut mendapatkan pembekalan penggunaan VCS selama hanya 4 jam saja, dan mereka langsung dapat mengadopsi teknologi VCS ini dalam pengerjaan tugas besar mereka selama 13 minggu.



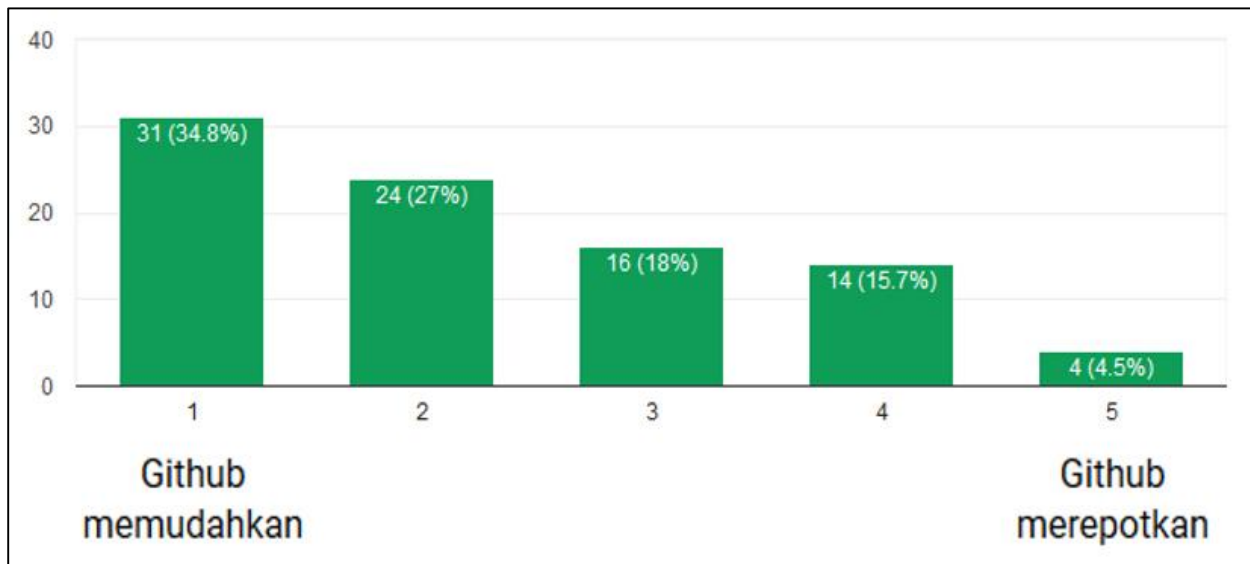
Gambar 1. Kemudahan penggunaan VCS menurut mahasiswa

Jika pertanyaan pertama mengukur kemudahan penggunaan VCS secara umum, pertanyaan kedua ditujukan khusus untuk mengukur apakah penggunaan VCS ini dapat memudahkan seorang mahasiswa berkolaborasi dengan mahasiswa lainnya dalam tim.

Berdasarkan survei yang dilakukan, mayoritas mahasiswa (61,80%) menyatakan bahwa Git dan Github memudahkan mereka berkolaborasi. 17,98% menyatakan netral, tidak memudahkan dan tidak merepotkan. Sisanya yaitu sebesar 20,22% berpendapat bahwa menggunakan VCS (Git dan Github) malah lebih merepotkan mereka dalam berkolaborasi. Distribusi dari jawaban mahasiswa yang lebih detail dapat dilihat pada Gambar 2.

Hasil ini sejalan dengan pertanyaan selanjutnya yang mengukur sejauh mana pengaruh penggunaan VCS terhadap cara kerja mahasiswa dalam tim. Hampir semua mahasiswa (97,3%) menyatakan mereka meninggalkan cara lama dalam melakukan kolaborasi.

Sebelum menggunakan VCS, setiap mahasiswa dalam tim melakukan *coding* bagian mereka di komputer/laptop mereka masing-masing, lalu memakai flashdisk atau email untuk mengirimkan *coding* tersebut ke satu orang yang bertugas menggabungkan pekerjaan mereka. Setelah menggunakan VCS, mereka cukup melakukan *commit* dan *push* ke repositori Github. Rekan satu tim dapat langsung menerima perubahan yang dilakukan dengan *pull*.



Gambar 2. Kemudahan berkolaborasi dalam tim menggunakan VCS

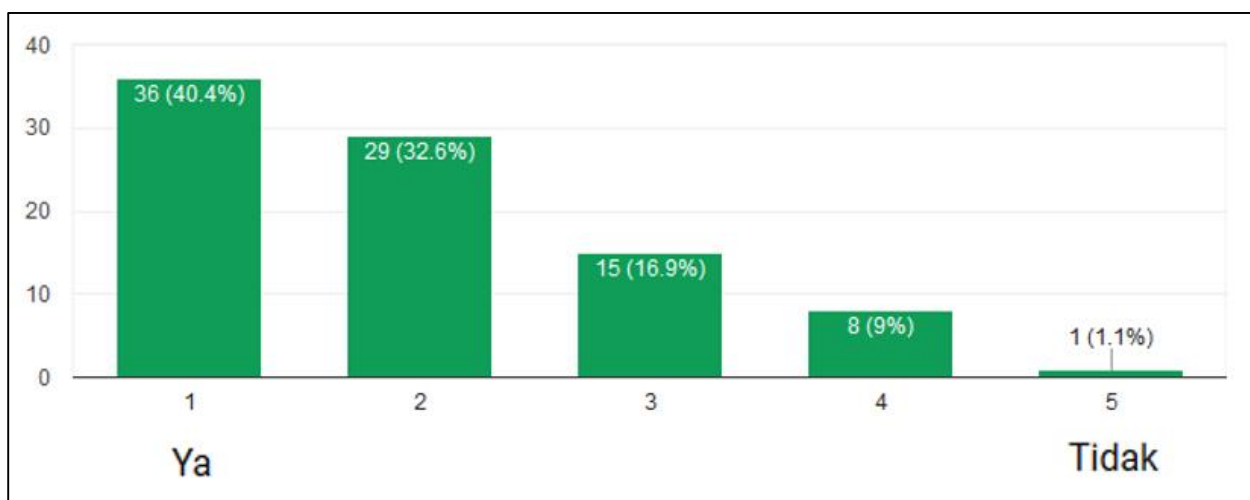
Di sisi lain, penggunaan VCS ternyata dapat meningkatkan motivasi mahasiswa untuk melakukan lebih banyak *coding* pada pengerjaan tugas besar mereka. Hal ini terungkap pada hasil survei pertanyaan keempat dimana terdapat sebanyak 73,03% mahasiswa yang setuju bahwa penggunaan Git dan Github membuat mereka termotivasi untuk melakukan lebih banyak *coding*.

Sebanyak 16,85% menyatakan netral dan sisanya sebesar 10,12% berpendapat bahwa penggunaan VCS (Git dan Github) tidak memotivasi mereka untuk melakukan lebih banyak *coding*. Distribusi dari jawaban

mahasiswa yang lebih detail dapat dilihat pada Gambar 3.

Salah satu hal yang menyebabkan mahasiswa termotivasi untuk lebih banyak melakukan *coding* adalah kenyataan bahwa di repositori Github, dosen dapat melihat siapa mengerjakan apa. Dengan demikian, kontribusi dari setiap mahasiswa dalam pengerjaan tugas besar dapat dilihat secara transparan.

Oleh karena nilai individu tugas besar memperhitungkan kontribusi ini, mahasiswa cenderung lebih aktif dalam melakukan *coding*. Pembagian tugas dalam kelompok juga menjadi lebih adil, karena masing-masing mahasiswa ingin memiliki kontribusi yang seimbang.



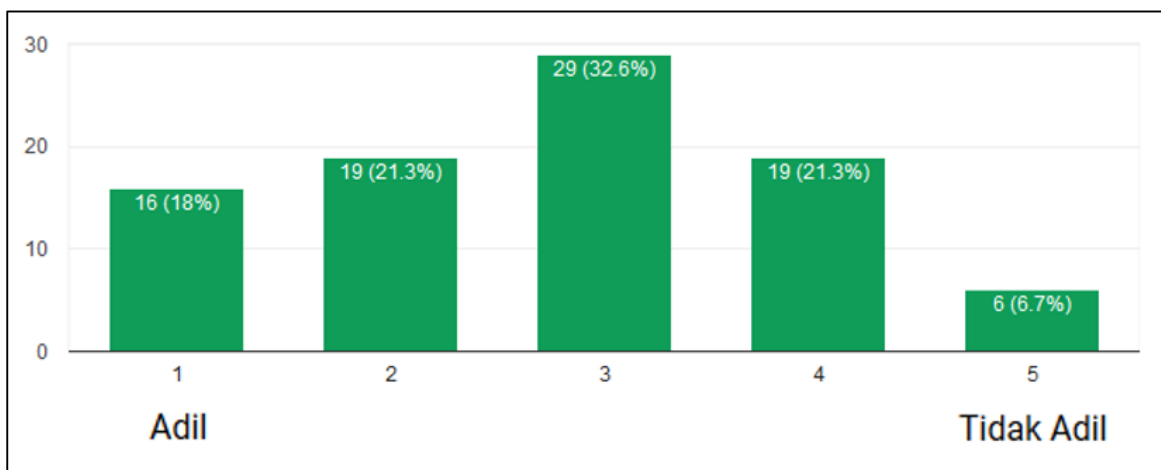
Gambar 3. Pengaruh VCS terhadap motivasi coding mahasiswa

Hal di atas berbeda jauh sebelum pengerjaan proyek tugas besar diwajibkan menggunakan VCS. Di masa itu, proses penilaian kontribusi ini lebih sulit karena tidak ada data yang memadai. Tugas besar yang dikumpulkan hanya memuat kondisi akhir proyek. Dengan demikian, cara untuk menilai kontribusi masing-masing individu hanya melalui tanya jawab pada saat mahasiswa melakukan presentasi tugas besarnya.

Namun ternyata, ketika mahasiswa dihadapkan kepada pertanyaan terkait rasa keadilan dalam proses penilaian, respon mereka berbeda jauh dengan sebelumnya. Persentase mahasiswa yang merasakan bahwa penilaian telah dilakukan secara adil dengan mahasiswa yang netral dan merasa tidak adil berturut-turut

sebesar 39,33%, 32,58% dan 28,09%. Dengan demikian dapat dikatakan tidak ada pendapat yang menjadi mayoritas. Distribusi dari jawaban mahasiswa yang lebih detail dapat dilihat pada Gambar 4.

Berdasarkan jawaban yang diberikan mahasiswa ketika menjawab pertanyaan terbuka (pertanyaan terakhir), hal di atas disebabkan karena proses penilaian kontribusi yang memberi nilai lebih tinggi bagi yang melakukan *coding* lebih banyak. Padahal, menurut mahasiswa, bagian yang mereka kerjakan adalah fitur penting atau fitur utama aplikasi. Hanya saja, karena fitur tersebut memerlukan jumlah kode yang lebih sedikit, mereka jadi mendapat nilai yang lebih kecil.



Gambar 4. Persepsi rasa keadilan dalam proses penilaian

Meski proses penilaian berbasis kontribusi belum dapat dikatakan telah memenuhi rasa keadilan bagi mayoritas mahasiswa sebagaimana yang dibahas di atas, namun jika dilihat secara umum, mahasiswa dapat merasakan bahwa penggunaan VCS dalam perkuliahan dapat membantu mereka.

Oleh karena itu, ketika mahasiswa diberikan pertanyaan apakah penggunaan VCS ini sebaiknya dilanjutkan ke angkatan setelah mereka atau tidak, mayoritas mahasiswa menjawab lanjutan dengan persentase mencapai sebesar 93,3%. Hanya 6,7% saja mahasiswa yang tidak merekomendasikan penggunaan VCS bagi angkatan setelah mereka.

#### KESIMPULAN DAN SARAN

Penggunaan *Version Control System* (VCS) terbukti dapat memberikan pengaruh positif terhadap proses belajar pemrograman mahasiswa dalam pengembangan sebuah perangkat lunak. VCS mudah digunakan, terbukti dengan waktu pembekalan yang relatif singkat, mahasiswa langsung dapat mengadopsi teknologi ini.

Manfaat yang dirasakan sangat besar, diantaranya kemudahan untuk berkolaborasi sesama anggota tim, meningkatkan motivasi mahasiswa untuk melakukan lebih banyak *coding*, pembagian kerja dalam kelompok yang lebih seimbang dan proses penilaian yang menjadi lebih transparan karena kontribusi yang terukur.

Penelitian selanjutnya akan mencoba untuk memformulasikan suatu cara menilai kontribusi dalam kelompok yang dapat memberikan rasa keadilan bagi mayoritas mahasiswa. Dengan demikian, selain mahasiswa akan merasa puas dengan proses belajar yang mereka lalui, mereka juga akan merasa puas dengan nilai yang mereka dapatkan di akhir semester.

#### **DAFTAR PUSTAKA**

- Borges, H., Hora, A., & Valente, M. T. (2016, September). Predicting the popularity of GitHub repositories. In *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering* (p. 9). ACM.
- Davis, R. C. (2015). Git and GitHub for librarians. *Behavioral & Social Sciences Librarian*, 34(3), 158-164.
- Forsyth, B., Kimble, C., Birch, J., Deel, G., & Brauer, T. (2016). Maximizing the Adaptive Learning Technology Experience. *Journal of Higher Education Theory & Practice*, 16(4).
- Höglblom, M., & Green, V. (2015). Version Control Systems in Corporations: Centralized and Distributed-An explorative case study into the corporate use of version control systems.
- Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, 7(4), 396.
- Kemkominfo. (2017). Indonesia Darurat Tenaga Programmer. Retrieved 20 Maret 2019 from [https://kominfo.go.id/content/detail/10247/indonesia-darurat-tenaga-programmer/0/sorotan\\_media](https://kominfo.go.id/content/detail/10247/indonesia-darurat-tenaga-programmer/0/sorotan_media)
- Kanbach, D. K., & Stubner, S. (2016). Corporate accelerators as recent form of startup engagement: The what, the why, and the how. *Journal of Applied Business Research*, 32(6), 1761.
- Kemenristek Dikti. (2019). Pangkalan Data Pendidikan Tinggi. Retrieved 21 Maret 2019 from <https://forlap.ristekdikti.go.id/prodi>
- Kemkominfo. (2018). Daftar Unit Kom-petensi Okupasi dalam Kerangka Kualifikasi Nasional Indonesia Bidang Teknologi Informasi dan Komunikasi.
- Kim, K. J., & Shin, D. H. (2015). An acceptance model for smart watches: implications for the adoption of future wearable technology. *Internet Research*, 25(4), 527-541.
- Leonard, A., Currie, S., Alley, J., Andersson, M., Avenant, P., Fellows, B., ... & Wilhelmsen, C. (2017). Source Control. In *The Biml Book* (pp. 391-412). Apress, Berkeley, CA.
- Lin, H. C. K., Chen, M. C., & Chang, C. K. (2015). Assessing the effectiveness of learning solid geometry by using an augmented reality-assisted learning system. *Interactive Learning Environments*, 23(6), 799-810.
- Majumdar, R., Jain, R., Barthwal, S., & Choudhary, C. (2017, September). Source code management using version control system. In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 278-281). IEEE.
- Suroso JS, Sriratnasari SR. A Literature Review on The Challenges of Adopting Cloud Computing for Startup in Indonesia. In *2018 Indonesian Association for Pattern Recognition International Conference (INAPR) 2018 Sep 7* (pp. 315-321). IEEE.