

CODE LEARNER : APLIKASI BELAJAR PRAKTEK PEMROGRAMAN ONLINE SECARA MANDIRI BERBASIS MOBILE

Hanif Fatullah¹, Jumail^{2*}, dan Rita Dewi Risanty³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jakarta

*jumail@umj.ac.id

Abstrak

Berdasarkan Buku Panduan Penyelenggaraan Pembelajaran Semester Gasal 2020/2021 Perguruan tinggi EDISI I 22 JUNI 2020, belajar mandiri adalah sebuah proses yang diperlukan setiap mahasiswa dalam periode tertentu khususnya pada masa pandemi saat ini. Mayoritas mahasiswa lebih suka menggunakan aplikasi WA Grup ketika melakukan pembelajaran secara daring dan tingkat pemahaman materi mereka terbilang masih kurang. Bahkan, kuliah online sering diidentikkan dengan banyaknya tugas yang diberikan kepada mahasiswa sehingga mengakibatkan proses perkuliahan tidak efektif. Kurangnya pemahaman tersebut dapat disebabkan oleh mahasiswa yang tidak serius mengerjakan tugas mereka dan selalu mengandalkan search engine saat mengerjakan tugas yang ada, serta penyelenggaraan praktikum kuliah yang sulit dilakukan sehingga dibutuhkan sebuah solusi dalam permasalahan tersebut. Karna itu penulis membuat sebuah sarana aplikasi latihan pemrograman dengan compiler sederhana yang bernama Code Learner sebagai sarana aplikasi pembelajaran mandiri yang lebih interaktif dan mudah di mengerti khususnya di bidang pemrograman.

Kata Kunci: *Compiler, Error Handling, Aplikasi Android, Latihan pemrograman*

Abstract

Based on the “Buku Panduan Penyelenggaraan Pembelajaran Semester Gasal 2020/2021 Perguruan Tinggi EDISI I 22 JUNI 2020”, individual learning is a process that is needed by every student in a certain period, especially during the current pandemic. The majority of students prefer to use the WhatsApp Group application when doing online learning and their level of understanding of the material is still lacking. In fact, online lecture are often identify with the many assignments given to students, resulting in an ineffective lecture process. This lack of understanding could be cause by students who are not serious about doing their assignments and always rely on search engines when doing existing assignments, as well as conducting lectures that are difficult to do so that a solution is need for these problems. Because of that, the author make a programming practice application facility with a simple compiler called Code Learner as a means of independent learning applications that are more interactive and easy to understand, especially in the programming field.

Keywords: *Compiler, Error Handling, Android Application, Programming practice*

1. Pendahuluan

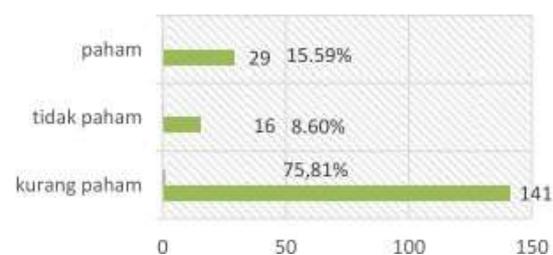
Teknologi Informasi berkembang sangat pesat, khususnya teknologi yang mendukung penyelenggaraan pembelajaran berbasis elektronik (e-learning). Dikutip berdasarkan Roy Sembel dan Sarah Sembel mengemukakan bahwa “Pembelajaran elektronik atau e-learning adalah pembelajaran jarak jauh yang memanfaatkan teknologi komputer dan juga jaringan internet” (Sadiman, 2008). E-learning dapat mengembangkan kemandirian mahasiswa dalam pengalaman saat menggunakannya. Selain itu, e-learning dapat membantu dosen dalam memantau aktivitas mahasiswa pada berbagai tugas.

Compiler sebagai alat untuk melakukan proses terhadap suatu bahasa sumber menjadi suatu bahasa target yang diinginkan. Selain melakukan transformasi, compiler mempunyai error handling untuk menemukan pesan error yang dapat mempermudah dalam menangani masalah pada bahasa pemrograman.

Berdasarkan Buku Panduan Penyelenggaraan Pembelajaran Semester Gasal 2020/2021 Perguruan tinggi EDISI I 22 JUNI 2020, belajar mandiri adalah sebuah proses yang diperlukan setiap mahasiswa dalam periode tertentu khususnya pada masa pandemi saat ini. Untuk membantu lancarnya pembelajaran mandiri dosen diharapkan menyiapkan beragam tugas yang mampu meningkatkan kemampuan belajar mahasiswa dengan memanfaatkan Teknologi Informasi dan Komunikasi (TIK) yang ada (Dirjen- Dikti, 2020).

Hasil survei yang dilakukan oleh UNICEF mengenai tingkat kenyamanan belajar dirumah saat melakukan proses pembelajaran daring adalah 66 persen dari 4000 responden (Ayunda Pininta Kasih, 2020). Menurut Aan Widiyono saat

melakukan penelitian pada suatu universitas, Mayoritas mahasiswa lebih suka menggunakan aplikasi WA Grup ketika melakukan pembelajaran secara daring dan tingkat pemahaman materi mereka terbilang masih kurang. Bahkan, kuliah online sering diidentikkan dengan banyaknya tugas yang diberikan kepada mahasiswa sehingga mengakibatkan proses perkuliahan tidak efektif. (Widiyono, 2020).



Gambar 1 Pemahaman pembelajaran daring selama Pandemi Covid 19 (Widiyono, 2020)

Pada gambar 1.1 terdapat tingkat pemahaman pada saat dilakukannya pembelajaran daring oleh Aan Widiyono, 29 orang paham pada saat pembelajaran daring,

16 orang tidak paham, dan 141 orang mengaku kurang paham pada masa pembelajaran daring dilakukan. Kurangnya pemahaman tersebut dapat disebabkan oleh mahasiswa yang tidak serius mengerjakan tugas mereka dan selalu mengandalkan search engine saat mengerjakan tugas yang ada, serta penyelenggaraan praktikum kuliah yang sulit dilakukan sehingga dibutuhkan sebuah solusi dalam permasalahan tersebut.

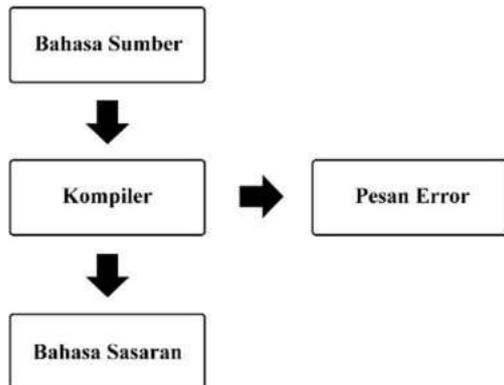
Untuk itu dalam membantu meningkatkan kemampuan pemahaman dan meningkatkan kemauan belajar mahasiswa diperlukan sebuah sarana aplikasi pembelajaran mandiri yang lebih interaktif dan mudah dimengerti khususnya di bidang pemrograman. Karna itu penulis membuat sebuah sarana aplikasi latihan pemrograman dengan

compiler sederhana yang bernama Code Learner. Code Learner adalah sebuah aplikasi latihan pemrograman mobile yang dilengkapi dengan compiler sederhana dengan tujuan agar setiap tugas pemrograman belajar mandiri dapat lebih interaktif dan mudah dipahami dalam proses pembuatan logika program dengan adanya aplikasi ini khususnya pembelajaran di bidang pemrograman untuk mahasiswa.

2. Literature Review

Compiler

Compiler merupakan alat yang membantu melakukan sebuah proses terhadap suatu Bahasa sumber menjadi Bahasa yang diinginkan. Bahasa sumber biasanya menggunakan Bahasa tingkat tinggi, sedangkan Bahasa yang diinginkan adalah Bahasa yang akan diterjemahkan menjadi Bahasa mesin itu sendiri. Compiler bisa juga kita sebut sebagai penterjemah. Compiler harus dapat mengetahui sebuah kesalahan dalam program sumber (Fauziah et al., 2019).



Gambar 2 Konsep Dasar Teknik Kompilasi

Compiler adalah perangkat lunak yang dapat merubah instruksi dari Bahasa basic ke dalam Bahasa pemrograman tingkat tinggi yang bertujuan dapat dimengerti oleh system (Koli et al., 2015).

Analisis dan Sintesis

Proses compiler dibagi dua :

- 1) Analisis: di tahap ini, program yang ditulis oleh user pecah menjadi bagian bagian yang akan diolah Kembali ke

dalam program sumber. Operasi yang dilakukan oleh program sumber ditulis dalam satu struktur yang disebut dengan nama tree.

- 2) Sintesis: Program tujuan dibuat oleh representasi yang di hasilkan pada analisis sebelumnya. Compiler juga perlu editor yang melakukan penghubungan dua hal dengan loading dan juga link-editing. Loading dilakukan pada saat pengambilan kode mesin yang mana perintah dan data yang didapat diletakkan pada memori (Fauziah et al., 2019).

Analisis Leksikal

Analisis leksikal merupakan bentuk tahapan pertama pada pemrograman *compiler*. Analisis leksikal menerima masukan berbentuk *source code* dalam bentuk *character*. *Character* akan diproses untuk membuat token, token yang dibuat mempunyai *pattern* yang menghasilkan *keyword, identifier, number, float, string, dll* (Febrianza & Purnamasari, 2017)

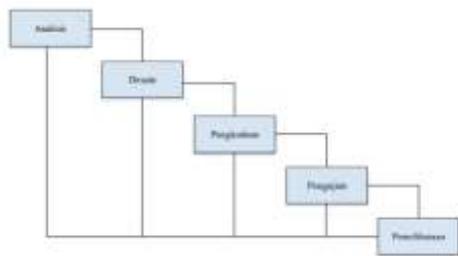
Analisis Leksikal merupakan salah satu pemrosesan Bahasa alami yang merupakan bagian dari sistem cerdas. Analisis leksikal dapat mengidentifikasi kode sumber (Muktiadi & Wibowo, 2014).

E-Learning

Perkembangan teknologi yang pesat sangat mendukung adanya pembelajaran yang diselenggarakan secara daring. E-Learning mampu mengembangkan kemandirian siswa dengan tingkat interaktifitas yang tinggi. Interaktifitas tersebut yang selain menyajikan materi belajar berbentuk sebuah file, e-learning juga mempunyai menu yang lebih bervariasi seperti fasilitas konsultasi online (Wibowo et al., 2015)

SDLC (Software Development Life Cycles)

SDLC atau Software Development Life Cycles adalah sebuah tahapan pekerjaan yang dilakukan oleh analis sistem atau programmer dalam membangun sebuah aplikasi (Dwanoko, 2016) .

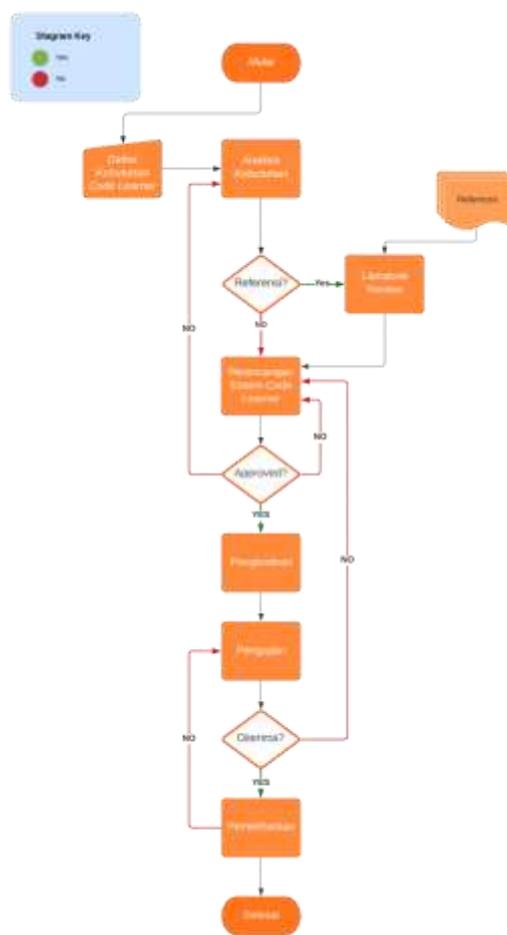


Gambar 1 Model Waterfall (Kusniyati & Putro, 2018)

SDLC dengan model Waterfall berfungsi untuk menggambarkan langkah langkah dari setiap tahapan secara disiplin, terbagi dalam empat kegiatan utama, yaitu requirement, analisis, design, programming, testing dan implementasi serta pemeliharaan (Andry, 2017).

3. Metodologi Penelitian

Dalam mencapai tujuan penelitian dirancang satu methodology penelitian untuk menjelaskan tahapan-tahapan penelitian yang dilalui untuk menyelesaikan penelitian ini. Berikut adalah matodologi yang digunakan dalam penelitian ini, dimana metode kualitatif diterapkan untuk mengukur usability atau daya guna Code Learner sebagai satu perangkat lunak, dan juga mengukur tingkat penerimaan pengguna terhadap aplikasi Code Learner yang dibangun. Gambar 4 adalah metodologi yang diterapkan dalam penelitian ini.



Gambar 4. Metodologi Penelitian

4. Analisis Kebutuhan

Code learner memiliki beberapa kebutuhan fungsional yang dibagi keatas dua target pengguna utama atau *actors* yaitu Dosen dan Mahasiswa. Kebutuhan kedua actors penting di dalam perancangan sistem Code Learner, dimana interaksi pembelajaran akan melibatkan keduanya,yaitu Dosen dan Mahasiswa.

Adapun beberapa kebutuhan fungsional sistem Code Learner dijelaskan dalam Tabel 1.

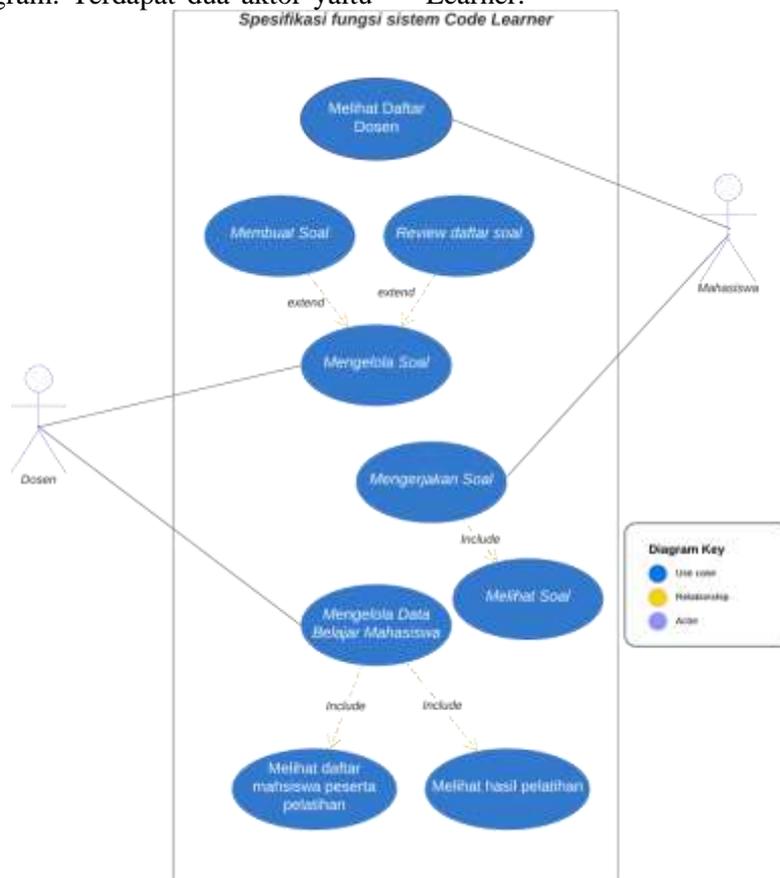
Tabel 1. Kebutuhan Fungsional Code Learner

| No | Aktor | Kebutuhan Sistem | Deskripsi |
|----|-----------|-------------------------------|--|
| 1 | Dosen | Melihat Soal | Dosen dapat melihat soal yang telah dibuat. |
| | | Menginput Soal | Dosen dapat menginput Soal logika sederhana dengan output yang pasti. |
| | | List Mahasiswa | Dosen dapat melihat mahasiswa yang mengikutinya. |
| | | Hasil Latihan | Dosen dapat melihat hasil latihan mahasiswa pada tiap soal yang dibuatnya. |
| 2 | Mahasiswa | List Soal | Mahasiswa dapat melihat list soal dosen yang diikuti jika masih belum di selesaikan oleh mahasiswa tersebut. |
| | | Kerjakan Soal | Mahasiswa dapat mengerjakan soal yang telah dibuat oleh dosen yang diikuti. |
| | | Mengikuti Dosen | Mahasiswa dapat mengikuti dosen yang diinginkan. |
| | | List Dosen yang telah diikuti | Mahasiswa dapat melihat dosen yang telah diikutinya. |

5. Perancangan Fungsional Sistem

Berikut adalah perancangan fungsional sistem Code Learner yang di visualisasikan dalam Use Case diagram. Terdapat dua aktor yaitu

Dosen dan Mahasiswa yang memiliki akses langsung kepada sistem dan menggunakan fungsi-fungsi yang ada dalam sistem. Gambar 5 adalah use case diagram aplikasi Code Learner.



Gambar 5 Use case diagram Code Learner

6. Pengujian Code Learner

Pengujian dilakukan untuk menilai ketepenuhan spesifikasi fungsi sistem aplikasi Code Learner, sesuai dengan analisis dan perancangan yang telah dilakukan. Tabel-tabel berikut menggambarkan beberapa hasil uji yang telah dilakukan terhadap target

perngguna yaitu Dosen dan Mahasiswa dari Fakultas Teknik, Universitas Muhammadiyah Jakarta. Tabel 2 menunjukkan hasil uji akses data dari aplikasi Code Learner ke basisdata oleh pengguna – Dosen, dan Tabel 3 target tugas yang sama namun dilakukan oleh Mahasiswa.

Tabel 2. Pengujian akses data via Code Learner ke basisdata oleh Dosen.

| Target User - Dosen | | | | |
|--|-------|---|--|----------|
| Yang Diuji | Input | Yang Diharapkan | Keterangan | Status |
| Melihat hasil latihan mahasiswa terbaru | - | <ul style="list-style-type: none"> Jika mahasiswa berhasil mengerjakan latihan tampilan akan berwarna cokelat. Jika mahasiswa gagal mengerjakan latihan tampilan akan berwarna merah. | Data hasil latihan berhasil ditampilkan dengan highlight warna cokelat. Data yang ditampilkan sesuai data yang tersimpan dalam basisdata. | Berhasil |
| Mendapatkan riwayat hasil latihan "Data Not Found" | | <ul style="list-style-type: none"> Jika hasil latihan Mahasiswa tidak ada di dalam database atau gagal didapatkan maka tampilan akan menampilkan tulisan "—Data Not Found—" | Data dalam basisdata tidak tersedia saat diminta. | Berhasil |

Tabel 3. Pengujian akses data via Code Learner ke basisdata oleh Mahasiswa.

| Target User – Mahasiswa | | | | |
|--|-------|--|--|----------|
| Yang Diuji | Input | Yang Diharapkan | Keterangan | Status |
| Melihat hasil latihan pribadi mahasiswa terbaru | - | Jika mahasiswa berhasil mengerjakan latihan tampilan akan berwarna cokelat. Jika mahasiswa gagal mengerjakan latihan tampilan akan berwarna merah. | Data hasil latihan berhasil ditampilkan dengan highlight warna cokelat. Data yang ditampilkan sesuai data yang tersimpan dalam basisdata. | Berhasil |
| Mendapatkan riwayat hasil latihan "Data Not Found" | | <ul style="list-style-type: none"> Jika hasil latihan Mahasiswa tidak ada di dalam database atau gagal didapatkan maka tampilan akan menampilkan tulisan "—Data Not Found—" | Data dalam basisdata tidak tersedia saat diminta. | Berhasil |

Salah satu ujian substansi dalam pengembangan aplikasi Code Learner ini

adalah memasukan kode sumber sebagai bahan latihan kedalam code learner. Pengujian ini untuk mengevaluasi apakah Code Learner dapat menjalankan fungsinya untuk

memberikan data respond berdasarkan input teknis kode sumber seperti penggunaan *conditional statement IF*.

Tabel 4. Pengujian Mengerjakan Soal Teknis Pemrograman dengan Code Learner

| Mengerjakan Soal - Mahasiswa | | | | |
|--------------------------------|---|--|--|----------|
| Yang Diuji | Data Masukan | Yang Diharapkan | Pengamatan | Status |
| Menjalankan Compiler program | LET nums = 11290 IF nums > 2000 THEN PRINT "benar" ENDIF | <ul style="list-style-type: none"> • Output sesuai dengan target cetak. • Jika terjadi error maka akan muncul highlight berwarna merah. • Jika terjadi ketidaksamaan dengan jawaban akan berwarna kuning, • Jika output yang dihasilkan berhasil sama maka akan berwarna biru. | Data berhasil di uji dan menghasilkan output "benar" warnabiru | Berhasil |
| | nums = 11290 IF nums > 2000 THEN PRINT "benar" ENDIF | | Data mengeluarkan output error " Error. Referencing variable before assignment: num" dan berwarna merah | Berhasil |
| | LET nums = 11290 IF nums > 2000 THEN PRINT "bentar" ENDIF | | Data berhasil di uji dan menghasilkan output "bentar" beserta notifikasi output tidak sesuai berwarna kuning | Berhasil |
| Submit Hasil Latihan Mahasiswa | Menekan tombol submit | <ul style="list-style-type: none"> • Program yang telah dibuat dan telah di compile dapat di submit akan memunculkan notifikasi sukses dan menuju ketampilan awal mahasiswa | Data berhasil disimpan dan muncul notifikasi ditampilkan awal mahasiswa | Berhasil |

7. Pembahasan

Berdasarkan penelitian yang telah dilakukan maka terdapat beberapa faktor penting baik itu faktor bisnis maupun teknis diantaranya sebagai berikut:

1. Pembuatan Aplikasi Code Learner ini menggunakan Metode Software Development Life Cycle (SDLC) melalui tahapan analisis, desain menggunakan UML, ERD dan juga Desain UI, Pengkodean pada aplikasi ini

menggunakan bahasa Java sebagai Back End Service, Python sebagai Compiler Service, dan React Native sebagai Front End Mobile. Pengujian pada aplikasi ini menggunakan teknik Black Box.

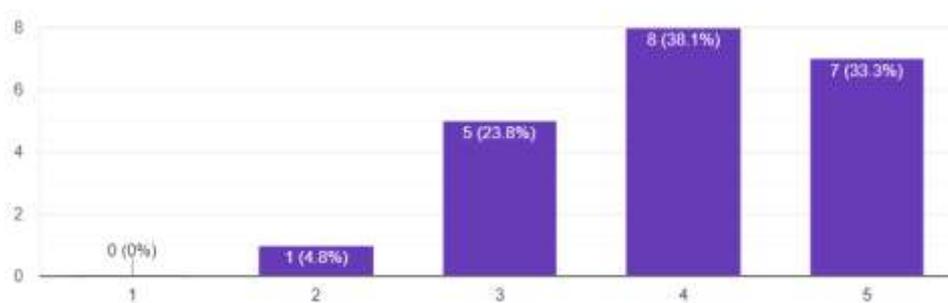
2. Analisis kebutuhan sistem berdasarkan studi literatur, didapat point permasalahan pada mahasiswa dan dosen seperti, kurangnya tingkat minat dan pemahaman pembelajaran yang disebabkan oleh mahasiswa yang tidak serius dalam mengerjakan tugas dan selalu mengandalkan search engine saat

mengerjakan tugas pembelajaran mandiri khususnya pada bidang pemrograman, serta tidak adanya praktik langsung yang dapat meningkatkan pemahaman mereka dikarenakan pembelajaran masih daring. Pembelajaran mandiri juga harus diawasi dosen untuk memantau perkembangan setiap mahasiswanya khususnya dibidang pemrograman.

3. Aplikasi Code Learner ini menyediakan kebutuhan mahasiswa dan dosen terkait pembelajaran mandiri tentang pemrograman. Mahasiswa dapat mengasah logika pemrograman dasar mereka dengan cara mengerjakan soal soal yang telah di buat oleh dosen dengan compiler sederhana pada Code Learner. Tersedia fitur seperti tambah soal, cari dosen, list dosen, list mahasiswa, hasil latihan mahasiswa, list soal, error handling pada program yang membuat aplikasi menjadi lebih interaktif.
4. 89% Mahasiswa yang mengikuti pengujian aplikasi dan mengisi kuesioner menyetujui bahwa aplikasi code learner dapat mempermudah mereka dalam membuat program sederhana. Error handling yang ada pada aplikasi code learner juga membantu mereka dalam mengetahui kesalahan pada program yang mereka buat.

User Interface dari code learner sangat mudah di pelajari

21 responses



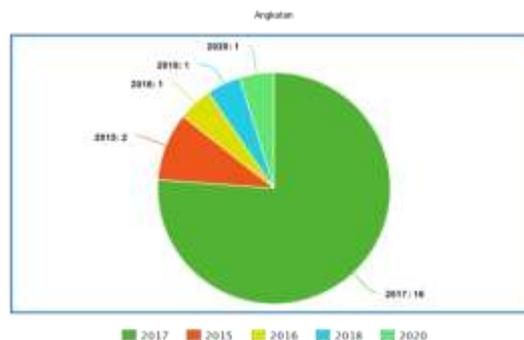
Gambar 7 efektifitas user interface pada aplikasi Code Learner menyatakan sulit dipelajari.

Berdasarkan Gambar 7 menunjukkan data bahwa 15 peserta atau 71,42% menyatakan bahwa user interface aplikasi code learner mudah dan sangat mudah untuk di pelajari, 5 orang atau 23,80% menyatakan cukup mudah, dan 1 orang atau 4,76%

8. Hasil

Hasil pengujian yang dilakukan terhadap Mahasiswa dan Dosen dengan mengambil target kasus Mahasiswa dan Dosen dari Fakultas Teknik, Universitas Muhammadiyah Jakarta.

Berikut adalah hasil pengujian yang telah dilakukan:



Gambar 6. Target peserta uji dari Mahasiswa

Berdasarkan Gambar 6 diperlihatkan bahwa peserta pengujian aplikasi Code Learner pada level mahasiswa melibatkan target mahasiswa dari berbagai angkatan seperti 2015, 2016, 2017, 2018, dan 2020.

Dimulai dengan menguji efektifitas user interface aplikasi Code Learner, melibatkan 21 peserta mahasiswa. Dari hasil pengujian user interface diperoleh hasil sebagai berikut.



Gambar 8 efektifitas Code Learner dalam pembelajaran Logika Pemrograman.

Gambar 8 menunjukkan efektifitas aplikasi Code Learner dalam kaitannya untuk memfasilitasi pengguna mahasiswa dalam mempelajari Logika Pemrograman. Dari total 21 peserta pengguna yang mengikuti pengujian 18 orang atau sebanyak 85,71% memberikan feedback bahwa Code Learner memberikan pemahaman yang lebih mudah bagi pengguna untuk memahami Logika Pemrograman berbanding dengan metode belajar konvensional di kelas, sedang 14,28% menyatakan sebaliknya.

Untuk membantu pembelajaran praktek, aplikasi Code Learner juga memfasilitasi feature error handling untuk membantu pengguna belajar dan memahami kejadian error dan saran bagaimana mengatasinya. Pengguna menguji soal pada kasus Logika Pemrograman dengan skenario error, dan pengguna diberikan notifikasi terkait error dan cara penanganannya. Sebanyak 18 atau 85,71% pengguna memberikan response bahwa feature ini membantu mereka dalam memahami lebih dalam tentang logika pemrograman dan cara penanggulangan error, dan sebanyak 3 pengguna atau sebanyak 14,28% memberikan respond sebaliknya.

Berdasarkan beberapa hasil uji terhadap aplikasi Code Learner diperoleh hasil bahwa lebih dari 80% features pada aplikasi Code Learner telah dapat digunakan dan membantu pengguna dalam memahami teknis pemrograman.

Daftar Pustaka

Andry, J. F. (2017). Pengembangan Aplikasi Backup Dan Restore Secara Otomatisasi Menggunakan Sdlc Untuk Mencegah Bencana. *Jurnal Muara*

Sains, Teknologi, Kedokteran Dan Ilmu Kesehatan, 1(1), 29–38. <https://doi.org/10.24912/jmstkik.v1i1.389>

Ayunda Pininta Kasih. (2020, June 24). *Survei UNICEF: 66 Persen Siswa Mengaku Tak Nyaman Belajar di Rumah Halaman all* - *Kompas.com*. <https://edukasi.kompas.com/read/2020/06/24/090832371/survei-unicef-66-persen-siswa-mengaku-tak-nyaman-belajar-di-rumah?page=all#page2>

Dirjen-Dikti. (2020). *Buku Panduan Penyelenggaraan Pembelajaran Semester Gasal 2020/2021 di Perguruan Tinggi*. 1–2.

Dwanoko, Y. S. (2016). IMPLEMENTASI SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) DALAM PENERAPAN PEMBANGUNAN APLIKASI PERANGKAT LUNAK. *Jurnal Teknologi Informasi*, 7(2), 83–94.

Fauziah, F., Apriansyah, A., Saputra, T. I., & Wijaya, Y. F. (2019). Desain Mesin Compiler untuk Penganalisa Leksikal, Sintaksis, Semantik, Kode Antara dan Error Handling Pada Bahasa Pemrograman Sederhana. *Journal of Applied Informatics and Computing*, 3(1), 01–07. <https://doi.org/10.30871/jaic.v3i1.1153>

Febrianza, G. G., & Purnamasari, K. K. (2017). Teknik Informatika – Universitas Komputer Indonesia *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 1, 1–9.

- <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-gungunfebr-36422>
- Koli, M. A. H., Marindani, E. D., & Hartoyo, A. (2015). Rancang Bangun Robot Bawah Air Mini ROV (Remotely Operated Vehicles) Berbasis Mikrokontroler ATmega16. *Jurnal Teknik Elektro Universitas Tanjungpura*, 2(1), 1–10.
- Kusniyati, H., & Putro, A. W. J. (2018). Rancang Bangun Aplikasi Reminder Pengumpulan Soal Ujian Di Smk Jakarta 1 Berbasis Web. *Petir*, 11(2), 191–202. <https://doi.org/10.33322/petir.v11i2.349>
- Muktiadi, R., & Wibowo, S. A. (2014). Analisis Leksikal untuk Mengidentifikasi Kata Kerja (Lexical Analysis to Identify a Verb). *Juita*, III(1), 43–46.
- Sadiman, A. S. (2008). *Pemanfaatan ICT untuk Pendidikan Pencegahan HIV/AIDS Penerapan Model Pembelajaran Interaktif Mendorong Penerapan e-Learning di Sekolah*. XII(1), 1–115.
- Wibowo, A. T., Akhlis, I., & Nugroho, S. E. (2015). Pengembangan LMS (Learning Management System) Berbasis Web untuk Mengukur Pemahaman Konsep dan Karakter Siswa. *Scientific Journal of Informatics*, 1(2), 127–137. <https://doi.org/10.15294/sji.v1i2.4019>
- Widiyono, A. (2020). Efektifitas Perkuliahan Daring (Online) pada Mahasiswa PGSD di Saat Pandemi Covid 19. *Jurnal Pendidikan*, 8(2), 169–177. <https://doi.org/10.36232/pendidikan.v8i2.458>