



Homepage Journal: <https://jurnal.unismuhpalu.ac.id/index.php/JKS>

Pengajaran Pemrograman Di Politeknik

Programming Teaching at Polytechnic

Sanif Sentosa^{1*}, Rafika Sari br Sembiring², Gloria Cahaya Julida Hutabarat³, Lenny Gurning⁴, Bahagia Tarigan⁵

¹Institut Bisnis Informasi Teknologi Dan Bisnis

²Universitas Mandiri Bina Prestasi

³Politeknik Unggulan Cipta Mandiri

⁴Politeknik Unggulan Cipta Mandiri

⁵Sekolah Tinggi Teologia Abdi Sabda

*Corresponding Author: E-mail: sanifsentosa549@gmail.com

Artikel Penelitian

Article History:

Received: 16 Dec, 2025

Revised: 18 Jan, 2026

Accepted: 24 Jan, 2026

Kata Kunci:

Pengajaran Pemrograman Di
Politeknik

Keywords:

Programming Teaching at
Polytechnic

DOI: [10.56338/jks.v9i1.10441](https://doi.org/10.56338/jks.v9i1.10441)

ABSTRAK

Terdapat banyak pendapat mengenai metode pengajaran pemrograman untuk mempersiapkan programmer yang berkualitas yang belajar secara otodidak. Programmer lulusan perguruan tinggi memiliki konsep berpikir dan standar yang berbeda dibandingkan programmer otodidak. Pendekatan pengajaran pemrograman dengan proses pengajaran yang nyaman dan menarik meningkatkan motivasi dan minat programmer untuk mempelajari lebih lanjut tentang pemrograman sebagai alternatif dalam pengajaran pemrograman. Pengajaran yang dimulai dengan konsep pemrograman dan diikuti oleh algoritma dan pemecahan masalah adalah pendekatan lain. Oleh karena itu, pendekatan lain dengan mengajarkan pemrograman secara langsung, diikuti dengan pengenalan algoritma dan pemecahan masalah dalam pemrograman, untuk menjadi solusi atas masalah tersebut. Selain itu, kemampuan logika, analisis, dan matematika diskrit menjadi dasar dalam pemecahan masalah.

ABSTRACT

There are many opinion about the teaching methode in programming to prepare the qualified programmer that learns autodidactly. Programmer as graduate of higher education has a thinking concept and the different standard than autodidacted programmer. The programming teaching approach by a convenient and interested teaching process increases the motivation and interest of programmer to study more about programming as alternative in programming teaching. The teaching beginning by the concept of programming and followed by algorithm and problem solving is another approach. Therefore, another approach by teaching the programming directly, followe by the introduction to algoritm and problem solving in programming, to be a solution on the problema. In addition, the logical, analysis and discrete mathematic capability to be a basic in problem solving.

PENDAHULUAN

Penelitian ini dibuat dengan tujuan untuk membuka wawasan tentang cara mengajarkan pemrograman dan untuk mendapatkan cara pengajaran pemrograman terbaik di politeknik. Filosofi pendidikan Politeknik dan Diploma yang berbasis kompetensi (vokasi) mengharuskan srtategi pengajaran pemrograman yang berbeda dengan pendidikan akademik. Kompetensi menekankan pada kemampuan melakukan/melaksanakan pekerjaan yang dilandasi pengetahuan keterampilan dan sikap

kerja. Dalam pengajaran pemrograman, kemampuan analisis menjadi hal yang sangat penting. Dari filosopinya, kemampuan analisis lebih ditekankan pada Strata Satu dibandingkan dengan Politeknik. Untuk itu perlu dikaji bentuk pengajaran pemrograman yang terbaik untuk diterapkan di Politeknik. Selain itu pengajaran pemrograman untuk peserta studi yang tidak memiliki dasar sama sekali juga menjadi pertimbangan dalam penerapan strategi tersebut. Faktor teknologi dan paradigma dalam sebuah bahasa pemrograman yang diajarkan juga menjadi hal yang perlu diperhatikan dalam metode pengajaran.

Dalam sebuah diskusi di forum web, muncul pertanyaan bagaimana cara atau pendekatan terbaik dalam mengajarkan programming khususnya untuk seseorang dengan pengetahuan programming yang masih sedikit atau bahkan tidak ada sama sekali. Banyak pendapat muncul sebagai tanggapan untuk pertanyaan tersebut. Beberapa menyatakan bahwa mengajar programming bagi non-programmers sebaiknya dimulai dari hal-hal yang menyenangkan yang bisa diperoleh melalui programming atau dengan membuat proses pembelajaran programming menjadi sesuatu yang menyenangkan. Penggunaan game dalam pembelajaran programming menjadi salah satu alternatif untuk itu, misalnya Robocode atau Code Rall (IBM). Programming untuk mengembangkan aplikasi berbasis web menjadi salah satu alternatif berbasis web pages dengan ‘instant gratification tool’ seperti frontpage atau macromedia menjadi hal yang menyenangkan. Jika mereka mulai menginginkan fungsi yang lebih tinggi, maka akan muncul keinginan untuk mengembangkan kemampuan dan kreatifitas mereka, misalnya dengan mempelajari Javascript.

Pandangan lain menyatakan bahwa motivasi merupakan kunci keberhasilan pembelajaran sesuatu termasuk programming. Pengajaran seorang non-programmer menjadi programmer dapat dilakukan dengan membuat mereka memperoleh atau memiliki motivasi yang benar. Perlu dicari jalan untuk menjelaskan bahwa dalam mengatasi suatu permasalahan perlu melibatkan programming. Dengan melihat sesuatu yang dapat melakukan sesuatu yang diinginkan dengan lebih cepat, lebih baik dan lebih mudah, mereka pasti akan dengan cepat memperoleh motivasi yang tinggi untuk belajar programming.

Pemilihan programming language juga menjadi salah satu faktor yang menjadi salah satu faktor yang mungkin menjadi penentu keberhasilan pembelajaran programming. Sebuah pandangan menyatakan menyatakan bahwa pemula harus mengerti detil dari assembler dan C sehingga bahasa tersebut perlu dipelajari. Pandangan lain menyatakan bahwa high level language (mendekati bahasa manusia) sebaiknya diberikan lebih dahulu, misalnya Python. Dengan bahasa tersebut pemula dapat memahami konsep dasar sebelum mereka harus berurusan dengan kompleksitas pointer dan malloc’s dalam bahasa C. Bahkan dapat dimulai dengan HTML, CSS, kemudian scripting , dilanjutkan dengan C.

Selain pandangan-pandangan tersebut beberapa aspek seperti : paradigma, bahasa pemrograman, dan metodologi pemrograman, dapat juga digunakan dalam menghasilkan metode pengajaran programming yang baik.

TINJAUAN LITERATUR

Paradigma dan Bahasa Dalam Pemrograman

Beberapa hal yang perlu diperhatikan dalam pembelajaran pemrograman adalah paradigma, bahasa, permasalahan (problem) yang digunakan sebagai contoh , dan metodologi pemrograman. Dari keempat hal tersebut paradigma dan bahasa pemrograman merupakan isu yang paling perlu ditelaah. Terdapat empat paradigma dalam pemrograman yakni :

1. Berorientasi prosedur (procedural oriented)
2. Berorientasi fungsi (functional oriented)
3. Berorientasi logika (logic oriented)
4. Berorientasi obyek (object oriented)

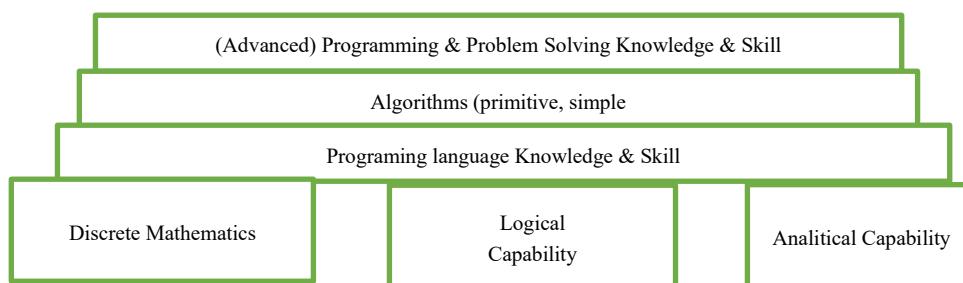
Setelah paradigma memiliki beberapa bahasa pemrograman yang sesuai. Pemilihan paradigma juga harus diikuti dengan pemilihan bahasa pemrograman yang digunakan. Karakteristik permasalahan, kemampuan programmer dan bahasa/tools yang digunakan akan berpengaruh pada paradigma yang akan dipilih. Problem dengan karakteristik logika matematika seperti aplikasi artificial intelligent akan membutuhkan pendekatan fungsi atau logic dengan bahasa LISP misalnya. Pendekatan berorientasi objek tentunya membutuhkan kemampuan analisis dan desain berorientasi objek sehingga bisa diimplementasikan dengan bahasa pemrograman yang berorientasi objek.

Pembelajaran programming dengan berbagai pendekatan tersebut, harus mempertimbangkan tingkat kesulitannya sehingga bagi pemula akan mengalami proses yang baik dalam rangka pengembangannya kemampuan pemrogramannya. Pembelajaran konsep dasar dalam pemrograman akan lebih mudah dipelajari dalam bahasa orientasi procedural/imperative, selain karena sudah terdapat high level language untuk pendekatan tersebut, kompleksitas permasalahan juga bisa dimulai dari yang lebih kecil. Teknologi berorientasi objek yang berkembang pesat merupakan pendekatan yang dapat digunakan berikutnya yang tentu saja sudah harus didahului kemampuan dalam analisis dan desain aplikasi berorientasi objek. Untuk lebih expert dengan kebutuhan problem solving yang lebih kompleks dapat dilakukan dengan paradigma berorientasi logika.

METODE

Metode Pada Pembelajaran Programming

Dalam beberapa pendekatan, pemrograman dimulai dengan algoritma yang merekomendasikan penggunaan pseudo-code sebagai pengganti sebuah bahasa pemrograman tertentu. Namun dalam programming knowledge and skill building blocks dalam gambar 1 menunjukkan bagaimana urutan pengajaran pemrograman. Pengajaran pemrograman harus didasari kemampuan matematis yang memadai (Matematika Diskrit, Logika Matematika dan Kemampuan Analitis). Dengan dasar tersebut mahasiswa mulai diajari dengan dengan kemampuan penguasaan suatu bahasa pemrograman tertentu diikuti dengan penguasaan konsep dasar pemrograman. Setelah kemampuan tersebut dimiliki, mahasiswa mulai dilatih untuk mengerti bagaimana langkah-langkah (algoritma) yang harus dilakukan dalam mengatasi permasalahan dengan menggunakan bahasa yang sudah dikuasai. Kemampuan bahasa pemrograman dan problem solving kemudian dapat dikembangkan dengan permasalahan yg lebih kompleks.



Gambar 1 Programing Knowledge and Skill Building Blocks

Metode penelitian yang akan digunakan dalam penelitian ini adalah R&D (Research & Development) penelitian formative dengan mengadaptasikan metode penelitian pengembangan yang diadaptasi oleh Tessmer (Akker, 1999). Penentuan subjek penelitian yang akan dilakukan dalam penelitian ini akan adalah purposed sampling dengan pertimbangan subjek pernah mempelajari bahasa pemrograman Python sehingga tahu tentang apa yang dibutuhkan selama pembelajaran. Subjek akan diambil dari mahasiswa Pendidikan Teknik Komputer. Lima Subjek penelitian dengan purposed sampling akan digunakan untuk uji validasi one-to-one. Untuk uji validasi expertsubjek penelitian adalah 2 orang

dosen ahli pemrograman. Adapun teknik pengumpulan data yang digunakan adalah kuisioner, studi kepustakaan dan dokumentasi.

1. Tahap preliminary Dilakukan beberapa analisis yaitu: analisis materi, dan analisis fungsional.
2. Tahap self evaluation Dilakukan observasi dan studi literature yang bertujuan untuk melihat bagaimana cara kerja sistem pembelajaran yang telah dikembangkan sebelumnya seperti w3school, Python Tutor, CodeRunner dan analisis materi yang akan digunakan dalam pembelajaran. Yang akan dijadikan acuan dalam pembuatan prototype.
3. Uji validasi atau uji kelayakan sistem pembelajaran Uji validasi dilakukan oleh 2 pakar (expert review) dan peserta didik (one-to-one) yang terdiri dari minimal 5 peserta didik dan dilaksanakan secara paralel. Hasil dari pengujian tersebut akan kemudian akan di analisis.

Dalam memvalidasi atau menguji kelayakan hasil pengembangan media pembelajaran, instrumen penelitian ini mengadaptasikan aspek penilaian dari Afandi (2015, p. 82).

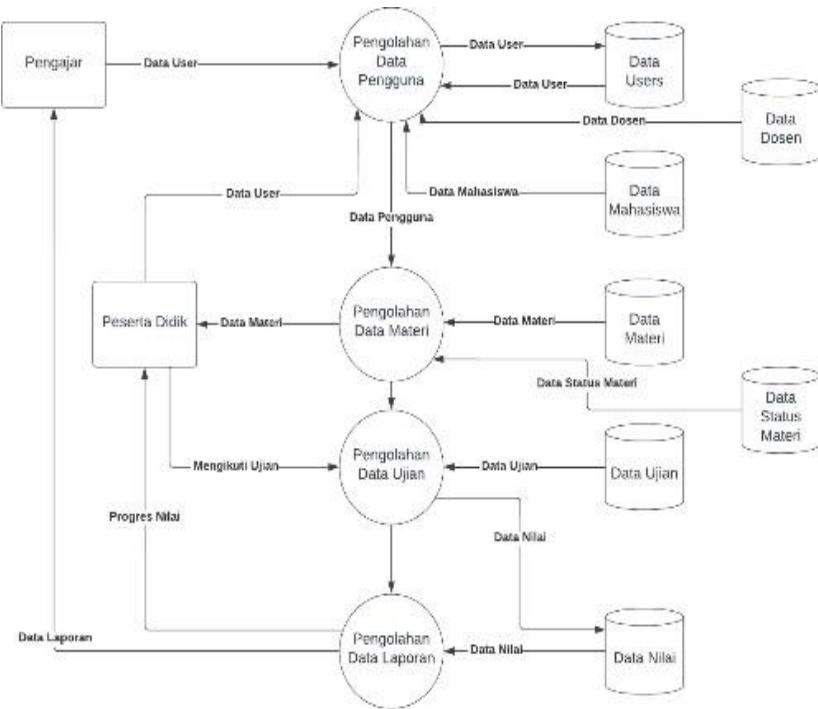
HASIL DAN PEMBAHASAN

Alur proses pada sistem pembelajaran ini akan mengadaptasi pembelajaran adaptif. juga menggunakan *Data Flow Diagram* (DFD) untuk menampilkan aliran data pada sistem, dan *Entity Relationship Diagram* (ERD) untuk menampilkan hubungan atau relasi antar entitas. Gambar 2 sampai gambar 6 adalah rancangan *Data Flow Diagram* (DFD) dari sistem pembelajaran adaptif.



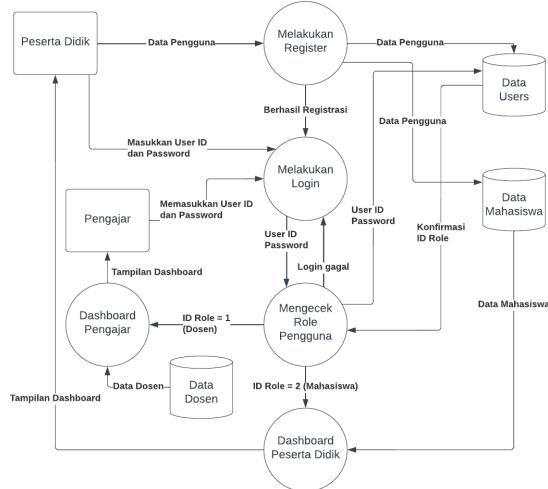
Gambar 2. Diagram Konteks

Konsep dari sistem pembelajaran adaptif digambarkan secara umum pada Diagram Konteks yang mengacu pada proses yang dilakukan sistem pembelajaran secara keseluruhan.



Gambar 3. DFD Level 0

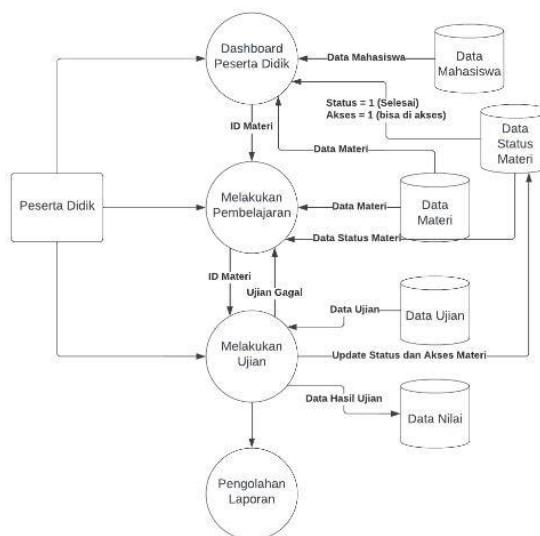
Gambar 3 menjelaskan, alur proses pengelolaan data pengguna, pengelolaan data materi, pengelolaan data ujian, serta pengelolaan data laporan.



Gambar 4. DFD Level 1 Pengolahan Data Pengguna

Gambar 4 merupakan DFD Level 1 yang menggambarkan proses pengolahan data pengguna. Pengelolaan data pengguna mencakup 5 proses, yang pertama proses melakukan registrasi data pengguna bagi peserta didik yang belum mendaftar kalau sudah berhasil peserta didik bisa langsung melanjutkan ke proses melakukan *login* jika gagal maka harus melakukan proses melakukan registrasi ulang. Proses melakukan *login* pengguna (peserta didik dan pengajar) diminta untuk memasukkan data *User ID* dan *Password* jika benar maka akan dilanjutkan ke proses mengecek *role* pengguna. Pada proses mengecek *role* pengguna apabila *role* pengguna adalah 1 maka sistem pembelajaran akan melanjutkan ke proses *dashboard* pengajar, sedangkan apabila *role* pengguna adalah 2 maka akan dilanjutkan ke proses *dashboard* peserta didik. Semua proses yang dilakukan untuk pengelolaan data pengguna dilakukan pada saat pengguna dalam kondisi belum *login* ke sistem pembelajaran, kecuali pada proses *dashboard* pengajar dan *dashboard* peserta didik.

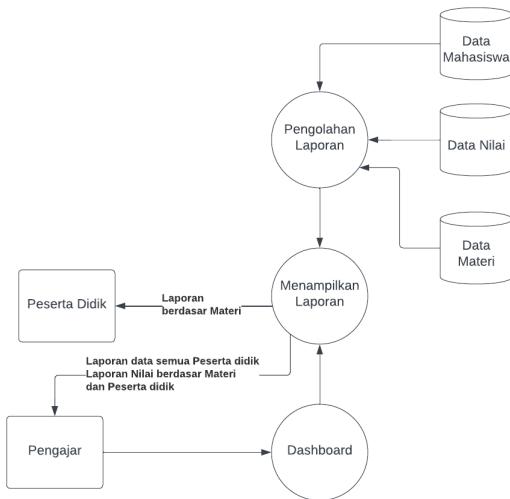
Proses melakukan *login* dan pengecekan *role* pengguna dilakukan untuk mendapatkan session otentifikasi untuk masuk ke dalam sistem pembelajaran adaptif, proses pengecekan *role* akan mengakses data peserta didik dan pengajar dengan memberikan nilai balik (*return*) otentifikasi pada pengguna sesuai dengan *role* yang dipunya oleh pengguna.



Gambar 5. DFD Level 1 Pengolahan Data Materi dan Ujian

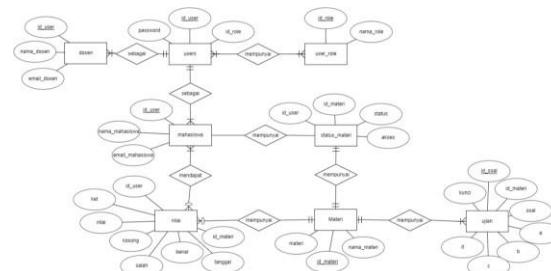
Gambar 5 merupakan DFD Level 1 yang menggambarkan proses dari *Dashboard* peserta didik, peserta didik melakukan pembelajaran, hingga peserta didik melakukan ujian yang kemudian dilanjutkan proses pengolahan laporan yang akan dibahas pada gambar 6 Pengelolaan materi, dan ujian ini juga menginterpretasikan salah satu model dalam pembelajaran adaptif yaitu model Instruksional. Data yang diolah berupa data peserta didik, beserta progress yang dilakukan peserta didik selama pembelajaran. Sistem juga akan merekam jumlah percobaan yang dilakukan peserta didik pada suatu ujian materi untuk mengetahui progres dan pada percobaan ke berapakah peserta didik mulai berhasil menjawab soal dengan benar dari materi yang disajikan.

Data yang diperoleh dari keseluruhan ini akan disimpan dalam basis data untuk dijadikan laporan nilai dan progres peserta didik pada proses selanjutnya, yaitu proses pengelolaan data laporan.



Gambar 6. DFD Level 1 Proses Pengolahan Laporan

Gambar 6 merupakan gambaran DFD Level 1 yang menggambarkan proses pengelolaan data laporan. Pengelolaan data laporan akan menginterpretasikan model pembelajaran adaptif yaitu model siswa, dimana peserta didik dapat memantau sampai dimana progres mereka dalam belajara pemrograman *Python*. Baik peserta didik dan pengajar akan mendapat data laporan namun dalam tingkat akses yang berbeda. Pada pengajar akses yang didapat yaitu laporan dari data semua peserta didik, dan Laporan Nilai berdasarkan materi dan peserta didik secara detail. Sedangkan pada peserta didik, akses yang didapat berdasarkan dari nilai yang hanya masing – masing peserta didik dapatkan.



Gambar 7. ERD Sistem Pembelajaran *Python*

Sistem pembelajaran adaptif yang dikembangkan memiliki 8 entitas, yaitu, *users*, *user_role*, dosen, mahasiswa, materi, status_materi, ujian, dan nilai. Entitas *users* digunakan untuk menyimpan data pengguna dari pengajar dan peserta didik, entitas *user_role* digunakan untuk menyimpan nama *roles* yang ada pada sistem pembelajaran, yaitu pengajar dan peserta didik. Entitas Dosen adalah entitas dari *users* yang digunakan untuk menyimpan data pengguna dengan *roles* sebagai pengajar. Sedangkan entitas mahasiswa adalah entitas dari *users* yang digunakan untuk menyimpan data pengguna dengan *roles* sebagai peserta didik. Entitas status_materi

digunakan untuk menyimpan progres belajar peserta didik dan berapa materi yang sudah bisa diakses oleh peserta didik. Entitas materi digunakan untuk menyimpan nama materi dan materi yang terdapat di dalam sistem pembelajaran. Entitas Ujian digunakan untuk menyimpan soal ujian beserta jawaban dan kunci jawaban setiap materi yang terdapat pada sistem pembelajaran. Sedangkan entitas nilai digunakan untuk mencatat progres nilai dan menghitung percobaan peserta didik dalam menyelesaikan materi.

KESIMPULAN

Politeknik menggunakan metoda pengajaran pemrograman yang sekaligus memperkenalkan pemrograman (programming) dan penyelesaian persoalan (problem solving) melalui bahasa pemrograman tertentu. Kemampuan dasar dalam pemrograman seperti matematika diksrit dan Logika Matematika perlu diberikan kepada mahasiswa sebelum mmepelajari bahasa pemrograman. Pemilihan bahasa bagi pemula menjadi faktor penting dalam penguasaan konsep dasar pemrograman yang menjadi penentu keberhasilan pembelajaran pemrograman.

Pengajaran algoritma diperlukan sebelum mahasiswa menggunakan pemrograman dalam mencari solusi dari suatu permasalahan (problem solving). Perlu adanya pengajaran paradigma pemrograman (khususnya object oriented) sebelum pengajaran bahasa pemrograman yang berbasis paradigma tersebut. Perlu diformulaiskan langkah-langkah yang dapat menghasilkan programmer lulusan sekolah informatika yang berbeda dengan programmer yang belajar secara otodidak serta mempunyai kemampuan yang sesuai dengan kebutuhan pasar.

DAFTAR PUSTAKA

- Afandi, R. (2015). Pengembangan media pembelajaran permainan ular tangga untuk meningkatkan motivasi belajar siswa dan hasil belajar IPS di sekolah dasar. JINoP (Jurnal Inovasi Pembelajaran), 1, 77-89. doi:<https://doi.org/10.22219/jinop.v1i1.2450>
- Akker, J. V. D. (1999). Principles and methods of development research. Design Approaches and Tools in Education and Training, 1-14. doi:https://doi.org/10.1007/978-94-011-4255-7_1
- Altadmri, A., & Brown, N. C. C. (2015). 37 Million Compilations. Paper presented at the Proceedings of the 46th ACM Technical Symposium on Computer Science Education. doi:<https://doi.org/10.1145/2676723.2677258>
- Alzahrani, N., Vahid, F., Edgcomb, A., Nguyen, K., & Lysecky, R. (2018). Python Versus C++. Paper presented at the Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18. doi:<https://doi.org/10.1145/3159450.3160586>
- Ateeq, M., Habib, H., Umer, A., & Rehman, M. U. (2014). C++ or Python? Which One to Begin with: A Learner's Perspective. Paper presented at the 2014 International Conference on Teaching and Learning in Computing and Engineering. doi:<https://doi.org/10.1109/LaTiCE.2014.420>
- Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? ACM SIGSOFT Software Engineering Notes, 41(6), 1-6. doi:<https://doi.org/10.1145/3011286.3011301>
- Bruhn, R. E., & Burton, P. (2003). An Approach to Teaching Java Using Computers. SIGCSE Bulletin, 35, 94 - 99. doi:<https://doi.org/10.1145/960492.960537>
- Cheah, C. S. (2020). Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. Contemporary Educational Technology, 12(2). doi:<https://doi.org/10.30935/cedtech/8247>
- Cui, L. (2016). Research on Educational Reform of Java Programming. doi:<https://doi.org/10.2991/mcei16.2016.147>

- Ebert, M., & Idea, A. G. (2017). Increase Active Learning in Programming Courses. 2017 IEEE Global Engineering Education Conference (EDUCON), 848-851. doi:<https://doi.org/10.1109/EDUCON.2017.7942946>
- Enbody, R. J., Punch, W., & McCullen, M. (2009). Python CS1 as Preparation for C++ CS2. Proceedings of the 40th ACM technical symposium on Computer science education - SIGCSE '09, 41. doi:<https://doi.org/10.1145/1508865.1508907>
- Gavrilovic, N., Arsic, A., Domazet, D., & Mishra, A. (2018). Algorithm for adaptive learning process and improving learners' skills in Java programming language. Computer Applications in Engineering Education, 26(5), 1362- 1382. doi:<https://doi.org/10.1002/cae.22043>
- Gomes, A., & Mendes, A. (2007). An environment to improve programming education. ACM International Conference Proceeding Series, 285. doi:<https://doi.org/10.1145/1330598.1330691>
- Hassinen, M., & Mayra, H. (2006). Learning programming by programming. Proceedings of the 6th Baltic Sea conference on Computing education research Koli Calling 2006 - Baltic Sea 06. doi:<https://doi.org/10.1145/1315803.1315824>
- Ismail, M., Ngah, N., & Umar, I. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. The Turkish Online J Edu Technol, 9.
- Kara, N., & Sevim, N. (2013). Adaptive Learning Systems: Beyond Teaching. Contemporary Educational Technology, 108-120. doi:<https://doi.org/10.30935/cedtech/6095>