

# IKHTISAR STRATEGI PEMBELAJARAN PEMROGRAMAN: SEBUAH INTEGRATIVE REVIEW

Manda Rohandi<sup>1\*</sup>, Ahmad Azhar Kadim<sup>2</sup>, Jemmy Pakaja<sup>3</sup>

<sup>1,2</sup>Pendidikan Teknologi Informasi, Universitas Negeri Gorontalo

\*Penulis korespondensi, email: [manda.rohandi@ung.ac.id](mailto:manda.rohandi@ung.ac.id)

## Abstract

*The industry's need for programmers is getting bigger, causing the interest to learn programming is also getting bigger. Many students experience difficulties and even fail in learning computer programming. This is due to a lack of understanding of the importance of learning programming, weak ability to think logically, lack of suitable learning materials, teaching methods that are still traditional, lack of practice in learning programming, testing methods that are still focused on knowledge of programming language syntax, and also computer programming related to several fields of technology. This article aims to see to what extent the programming learning strategies that have been implemented are effective in increasing the programming competence of students and teachers. The method in this article is the integrative literature review method. There were 20 articles that were search results and were selected based on predetermined inclusion and exclusion criteria. The findings in this study are the division of programming learning strategies into three groups, namely from the orientation of the programming elements, the learning strategies from the pedagogical point of view, and the learning strategies from other methods. All the learning strategies described in this article only concern hard skills and do not describe the soft skills needed by the industry. Therefore future research needs to discuss how to integrate soft skills into learning strategies and their application.*

**Keywords:** computer; soft skills; programming;

## Abstrak

Kebutuhan industri terhadap tenaga programmer yang semakin besar menyebabkan minat untuk mempelajari pemrograman juga semakin besar. Banyak siswa mengalami kesulitan dan bahkan gagal dalam mempelajari pemrograman komputer. Hal ini dikarenakan kurangnya pemahaman tentang pentingnya belajar pemrograman, lemah dalam kemampuan berpikir logis, kurangnya materi pembelajaran yang cocok, metode pengajaran yang masih tradisional, kurangnya praktik dalam pembelajaran pemrograman, metode pengujian yang masih dititikberatkan pada pengetahuan sintaks bahasa pemrograman, dan juga pemrograman komputer berhubungan yang dengan beberapa bidang teknologi. Artikel ini bertujuan untuk melihat sejauh mana strategi-strategi pembelajaran pemrograman yang telah di terapkan, efektif dalam meningkatkan kompetensi pemrograman siswa dan pengajar. Metode dalam artikel ini adalah metode integrative literature review. Terdapat 20 artikel hasil pencarian dan diseleksi berdasarkan kriteria inklusi dan eksklusi yang telah ditetapkan. Temuan dalam penelitian ini berupa pembagian strategi pembelajaran pemrograman kedalam tiga kelompok, yaitu dilihat dari orientasi elemen-elemen pemrograman, strategi pembelajaran dilihat dari pedagogiknya, dan strategi pembelajaran dilihat dari metode lainnya. Semua strategi pembelajaran yang dijabarkan dalam artikel ini, hanya menyangkut hardskill dan tidak menjabarkan softskill yang dibutuhkan industry. Oleh karena itu penelitian kedepannya perlu untuk membahas tentang bagaimana integrasi softskill kedalam strategi pembelajaran dan penerapannya.

**Kata kunci:** komputer; softskill; pemrograman;

© 2023 Information Technology Education FT UNG

## PENDAHULUAN

Perkembangan teknologi informasi yang semakin cepat, selalu di ikuti oleh perkembangan perangkat lunak maupun perangkat keras. Pengembangan kedua komponen teknologi

informasi ini, khususnya perangkat keras memerlukan ketersediaan alat dan bahan yang biasanya dilakukan oleh sektor industry. Adapun untuk pengembangan perangkat lunak, dapat dilakukan oleh sector industry atau dapat dikembangkan oleh seseorang yang hanya bermodalkan seperangkat PC atau laptop, sehingga berpotensi membuka lapangan pekerjaan yang luas dibidang pengembangan teknologi informasi. Menurut Shuermans dan Voskoglou (2019), pada awal 2019 terdapat 19 juta pengembang perangkat lunak didunia, tumbuh pesat diatas 20%. Meskipun demikian, menurut Berau of Labor Statistics, U.S, di Amerika sendiri jumlah pekerjaan untuk pengembang perangkat lunak sebanyak 1.365.000, dan di prediksi prospek pekerjaan pengembang perangkat lunak dari tahun 2018-2028 sebanyak 21% (tumbuh lebih cepat dari rata-rata).

Salah satu bagian dari pengembang perangkat lunak adalah programmer. Programmer merupakan ilmuwan computer yang terampil dalam menggunakan bahasa pemrograman untuk membangun program komputer yang dapat di eksekusi dan diterima oleh pemakai (Lateef, dkk. 2016). Menurut Berau of Labor Statistics, U.S, jumlah pekerjaan yang tersedia untuk computer programmers di amerika pada tahun 2018 sebanyak 250.300 lowongan, dengan gaji mencapai \$41.61/jam atau \$86.550/tahun. Meskipun gaji seorang programmers menjanjikan, namun untuk menjadi seorang programmer tidaklah mudah, diperlukan keahlian dalam menggunakan bahasa pemrograman, mempelajari konsep dan dapat menerapkannya kedalam suatu masalah, memiliki keterampilan dalam matematika, memiliki kemampuan dalam memecahkan masalah, memiliki keterampilan dalam berkomunikasi, memiliki keterampilan dalam menulis, dan memiliki rasa keingintahuan yang besar.

Banyak siswa mengalami kesulitan dan bahkan gagal dalam mempelajari pemrograman komputer. Hal ini dikarenakan pemrograman komputer berhubungan dengan beberapa bidang teknologi dan memerlukan pemahaman yang benar terhadap konsep yang abstrak (Lahtinen, dkk. 2005). Hal lainnya yang menyebabkan siswa kesulitan dalam mempelajari pemrograman adalah kurangnya pemahaman tentang pentingnya belajar pemrograman, lemah dalam kemampuan berpikir logis, kurangnya materi pembelajaran yang cocok, metode pengajaran yang masih tradisional, kurangnya praktik dalam pembelajaran pemrograman, metode pengujian yang masih dititikberatkan pada pengetahuan sintaks bahasa pemrograman (Zongda, dkk. 2011). Janpla dan Piriyasurawong (2018), menyebutkan bahwa kesulitan siswa dalam mempelajari pemrograman selain harus mengerti bagaimana program digunakan, mengetahui struktur dan makna dari penggunaan bahasa pemrograman dalam penulisan program, belajar perencanaan, memiliki keterampilan dalam mengembangkan, menguji dan bagaimana men-debug program, siswa juga harus menguasai bahasa Inggris yang merupakan bahasa yang digunakan dalam bahasa pemrograman. Kesulitan-kesulitan dalam pembelajaran pemrograman diatas sama dengan temuan yang didapatkan oleh Demilie (2019).

Artikel ini bertujuan untuk melakukan ikhtisar mengenai strategi-strategi yang telah digunakan dalam pembelajaran pemrograman komputer dan berhasil membantu mengatasi kesulitan siswa dalam pembelajaran pemrograman komputer.

## **METODE**

Artikel ini merupakan makalah integrative review yang bertujuan untuk menilai, mengkritik, dan mensintesis literatur dari topic penelitian tertentu yang memungkinkan kerangka teori dan perspektif yang baru (Torraco, 2005). Prosedur literature review pada

makalah ini dikerjakan dalam 4 langkah sebagaimana yang dilakukan oleh (Snyder, 2019), yaitu mendesain, melakukan review, menganalisis, serta membuat struktur dan menulis review.

### **Ketentuan Pencarian Artikel**

Sebelum benar-benar melakukan pencarian artikel, beberapa pencarian awal dilakukan dari Google Scholar untuk memperoleh beberapa artikel yang relevan untuk menentukan kata kunci dan mengoptimalkan pencarian sesuai ketentuan. Pada tahap selanjutnya, dilakukan pencarian pada basisdata artikel elektronik seperti, Google scholar, berdasarkan semua kata kunci yang relevan. Pencarian dilakukan berdasarkan kata kunci ‘metode belajar-mengajar pemrograman’, atau ‘strategi pembelajaran pemrograman’, yang terdapat pada judul, abstrak atau isi keseluruhan artikel.

### **Kriteria Inklusi dan Eksklusi Artikel**

Kriteria inklusi dalam makalah ini, adalah (1) artikel yang telah melalui penelitian empiris yang berasal dari jurnal peer-reviewed, karena merupakan temuan yang tervalidasi dan memiliki dampak yang besar dilapangan. (2) artikel yang berasal dari prosiding yang telah diseminarkan pada konferensi internasional yang fokus pada strategi pembelajaran pemrograman dan atau model pembelajaran secara umum yang berkaitan dengan artikel pembelajaran pemrograman. Adapun kriteria eksklusi dalam artikel ini, adalah (1) artikel yang berbahasa selain bahasa inggris, (2) bab pada buku, opini dan dokumen-dokumen yang tidak dipublikasikan, (3) opini atau tulisan-tulisan yang terdapat pada blog atau website yang tidak jelas.

### **Proses Pemilihan Artikel**

Berdasarkan kata kunci, artikel kemudian dicari berdasarkan kriteria inklusi dan eksklusi untuk mengidentifikasi artikel yang relevan. Artikel yang serupa dihilangkan redudansinya, selanjutnya artikel yang termasuk dalam kriteria eksklusi kemudian juga dieliminasi. Hasil eliminasi artikel didapatkan 20 artikel yang kemudian di review abstraknya untuk memastikan sesuai dengan batasan dan fokus dari penelitian pada makalah ini.

## **HASIL DAN PEMBAHASAN**

### **Hasil**

Mempelajari pemrograman menurut sebagian besar orang adalah hal yang tidak mudah. Diperlukan strategi belajar maupun mengajar yang sesuai, sehingga luaran keterampilan pemrograman yang diharapkan dapat terpenuhi. Dalam bab ini, strategi pembelajaran pemrograman dibagi kedalam tiga bagian, yaitu strategi pembelajaran dilihat dari orientasi elemen-elemen pemrograman, strategi pembelajaran dilihat dari pedagogiknya, dan strategi pembelajaran dilihat dari metode lainnya.

### **Strategi pembelajaran pemrograman dilihat dari elemen pemrograman**

Beberapa metode pembelajaran pemrograman telah dikembangkan untuk mencari metode yang sesuai dengan topic yang diajarkan. Szlavi dan Zsako (2003), memberikan daftar metode-metode yang sering digunakan dalam mengajarkan pemrograman, yaitu:

*Inverted: Journal of Information Technology Education, Vol 3, No2, Juli 2023*  
e-ISSN: 2828-3880, p-ISSN: 2828-2094

**Berorientasi metodologi algoritma**

Metode ini serupa dengan metode lainnya yang mencakup semua proses dalam pemrograman, seperti perencanaan algoritma dan struktur data, penulisan kode program, pengujian program, deteksi kesalahan dan perbaikan, control efisiensi dan control kualitas, dan terakhir dokumentasi. Metode ini cocok untuk diajarkan bagi mereka yang mempersiapkan diri untuk bekerja pada usaha informatika, atau pada pelatihan profesional di bidang informatika, dan atau pada pendidikan yang lebih tinggi dibidang informatika.

**Berorientasi data**

Serupa dengan metode sebelumnya, hanya lebih lebih utama ditekankan pada perbaikan jenis dan struktur data. Ide dasarnya adalah identifikasi masalah dipandang sebagai jenis spesifikasi dan mengkombinasikan perbaikan jenisnya dengan struktur algoritmik. Contohnya: masalah kartesian produk dikombinasikan dengan sekuensial, masalah gabungan dan alternative struktur data dikombinasikan dengan percabangan, masalah himpunan, urutan, hirarki dan struktur jaringan dikombinasikan dengan iterasi, dan masalah perulangan data atau jenis perulangan dikombinasikan dengan rekursif. Sama seperti metode yang pertama, metode ini juga cocok untuk orang yang ingin bekerja sebagai usahawan, mengikuti pelatihan profesional atau mengambil pendidikan yang lebih tinggi dibidang informatika.

**Berorientasi spesifikasi**

Mirip dengan dua metode yang sebelumnya, dimana spesifikasi formal (teknik berbasis matematis yang bertujuan untuk membantu implementasi sistem dan perangkat lunak) sebagai bagian yang paling penting dalam proses pengembangan perangkat lunak, algoritmanya diturunkan secara otomatis melalui spesifikasi itu sendiri, yang kemudian kode program itu dibuat berdasarkan instruksi penulisan program yang kaku. Metode ini cocok diajarkan bagi mahasiswa universitas tingkat atas, karena hanya berhasil jika mahasiswa memiliki dasar pengetahuan matematika.

**Berorientasi tipe masalah**

Metode ini berbeda dengan tiga metode sebelumnya, pemrograman dipandang sebagai satu kesatuan aktivitas secara keseluruhan dan tidak dapat dibagi menjadi beberapa bagian, karena metode ini memiliki fitur yang penting, yang mana faktanya kita selalu berurusan dengan program keseluruhan. Untuk dapat memecahkan masalah dibutuhkan gagasan pemrograman yang baru dan elemen-elemen yang diciptakan untuk memecahkan masalah tertentu. Keuntungan dari metode yang berorientasi tipe masalah adalah pengetahuan yang baru yang didapatkan dari kebutuhan alami bukan dari dugaan. Metode ini merupakan metode yang cocok untuk semua tingkat pendidikan, yang mana tujuan dari metode ini adalah mengembangkan cara berpikir algoritmik dan pelatihan dasar.

**Berorientasi bahasa**

Metode ini merupakan salah satu metode tertua dalam pengajaran pemrograman dan dianggap sudah ketinggalan zaman. Tujuan utamanya adalah menghasilkan program yang efektif. Metode ini hanya terikat pada satu bahasa pemrograman tertentu, oleh karena itu kita biasa mendengar istilah java programmers, C programmers, dan lain-lain. Dasar dari metode ini adalah mengajarkan bahasa pemrograman dan memperkenalkan pengetahuan pemrograman melalui bahasa tersebut. Orang yang belajar dengan metode ini, biasanya akan merasa kesulitan apabila berpindah menggunakan bahasa pemrograman yang lain, hal ini karena orang tersebut sudah terbiasa dengan satu bahasa pemrograman tertentu.

### **Berorientasi instruksi**

Metode ini serupa dengan metode berorientasi bahasa, namun dengan pengecualian bahwa bahasa pemrograman yang digunakan adalah bersifat umum dan tidak terikat dengan satu bahasa pemrograman saja. Metode ini mendefinisikan elemen-elemen bahasa pemrograman secara umum menurut prinsip-prinsip Newman. Meskipun demikian metode ini serupa dengan metode sebelumnya dianggap memiliki kelemahan dan sudah ketinggalan zaman.

### **Berorientasi matematika**

Masalah dalam metode ini diambil dari matematika, dimana suatu masalah didasarkan pada masalah lainnya yang disesuaikan dengan prinsip matematika. Diperlukan pemahaman matematika yang tinggi dan kemampuan untuk menerapkannya. Kelemahan dari metode ini adalah tidak adanya jaminan bahwa struktur apakah akan menjadi logis atau sempurna dilihat dari sudut pandang pemrograman.

### **Berorientasi perangkat keras**

Metode ini menganggap bahwa pengetahuan algoritmik tidak akan dapat dimengerti tanpa pengetahuan bahasa pemrograman tingkat tinggi, dan pengetahuan bahasa pemrograman tingkat tinggi tidak dapat dimengerti tanpa pengetahuan bahasa mesin atau assembly, dan oleh karena itu, pengetahuan bahasa assembly tidak akan dapat dimengerti tanpa pengetahuan bagaimana perangkat keras bekerja. Metode ini mencoba membangun pengetahuan pemrograman secara bottom-up, yaitu mengetahui bagaimana perangkat keras bekerja, sehingga kita bias mengetahui bagaimana bahasa tingkat tinggi bekerja. Hal ini tentunya memiliki konsekuensi bahasa seseorang tidak harus memiliki pengetahuan pemrograman seperti pengetahuan algoritma dan pemodelan data. Meskipun demikian, metode ini dianggap juga sudah ketinggalan zaman.

### **Berorientasi model**

Pada metode ini, model di berikan kepada siswa (algoritma atau kode program), yang kemudian siswa mengambil informasi dari model tersebut, sehingga mereka dapat menghasilkan program baru dengan memodifikasi model yang diberikan. Eksperimen memainkan peran penting dalam metode ini. Siswa memodifikasi kode program, mengevaluasinya, dan jika hasilnya sudah sesuai dengan yang diharapkan, maka modifikasi tersebut dinilai berhasil. Adapun jika tidak sesuai harapan, maka eksperimen harus diulang kembali. Metode ini hanya bermanfaat bagi orang yang memiliki tingkat kecerdasan yang tinggi.

### **Strategi pembelajaran pemrograman dilihat dari pedagogik**

Metode-metode pembelajaran pemrograman diatas lebih menekankan pada strategi pengembangan materi pembelajaran pemrograman. Terdapat beberapa pendekatan dan strategi pembelajaran pemrograman yang lebih ditekankan pada kemampuan pedagogik, sebagaimana yang dilakukan oleh Mironova, dkk (2016). Penelitian ini bertujuan untuk medemonstrasikan sebuah pendekatan dan beberapa strategi mengajar pemrograman untuk siswa non-IT. Dalam penerapannya, perkuliahan dilakukan melalui metode tatap muka, pengerjaan tugas secara berkelompok atau berpasangan, dan siswa dapat belajar secara mandiri melalui aplikasi e-learning Moodle yang telah disiapkan. Materi yang diberikan disesuaikan dengan siswa yang berlatarbelakang non-IT. Modul awal berupa materi yang berkaitan dengan text processing, presentation dan spreadsheets, yang menggunakan MS. Word, Power Point dan Excel. Dasar-dasar pemrograman yang merupakan bagian dari matakuliah informatika menggunakan Scratch sebagai tool yang sangat berguna untuk memberikan pemahaman visual tentang konsep dan terminology dari pemodelan dan pemrograman seperti data, proses, percabangan atau iterasi. Materi perkuliahan selanjutnya menggunakan Visual Basic atau Phyton. Visual basic dipilih karena tampilannya yang mirip dengan MS. Excel, sedangkan Phyton dipilih karena tidak memerlukan deklarasi variabel yang dapat memudahkan pemula dalam mempelajarinya. Perkuliahan bertujuan untuk menghasilkan luaran pembelajaran yang dapat memahami pendekatan berorientasi objek dan memperoleh keterampilan yang dibutuhkan dalam membangun algoritma. Dari hasil penelitiannya didapatkan bahwa untuk siswa non-IT tahun pertama haruslah focus pada model, algoritma dan visualisasinya, daripada pengajaran sintaks dan teknik penulisan program. Selain itu, didapatkan bahwa pembelajaran aktif sangat membantu tidak hanya untuk siswa dalam mendapatkan pengetahuan baru dan menggapai hasil yang lebih baik, namun juga membantu pengajar dalam mengembangkan dan meningkatkan kemampuan pedagogiknya ke tingkat yang lebih tinggi. Sedangkan strategi dan metode pengajaran pemrograman lainnya untuk pelajar IT diusulkan oleh Djenic dan Mitic (2017), yang mana menggunakan 4 strategi pembelajaran yaitu, Direct instruction, pembelajaran kolaboratif, situated learning, dan self-directed learning. Dalam pembelajaran direct instruction, pengajar melakukan aktivitas: memberikan materi, mempresentasikan dan mereview elemen-elemen program dan algoritmanya; memberikan instruksi sintaks program; memberikan demonstrasi program yang telah selesai dibuat. Siswa melakukan aktivitas: mengikuti pengajar secara interaktif; memberikan pertanyaan tentang elemen-elemen dan algoritma, serta program yang telah selesai dibuat. Dalam pembelajaran kolaboratif, pengajar membuat kelompok; memberikan instruksi untuk memecahkan masalah secara kelompok; dan memimpin diskusi tentang materi yang diajarkan. Adapun siswa secara berkelompok memecahkan masalah; aktif dalam berdiskusi tentang masalah dan materi; mencatat diskusi tentang masalah dan perkuliahan. Situated learning, pengajar memberikan instruksi untuk menggunakan tools pengembang disekitarnya; instruksi dalam latihan dan memecahkan masalah program; dan instruksi dalam menulis program. Sedangkan siswa melakukan mastering menggunakan tools yang ada; programme solving; menulis program; mengerjakan simulasi terhadap program yang telah dipecahkan sebelumnya; melakukan penelitian terhadap program yang telah dipecahkan sebelumnya. Dan yang terakhir adalah self-directed learning, yang mana pengajar memberikan arahan dan rekomendasi penelitian terhadap subjek tertentu. Siswa kemudian mencari semua bagian subjek dan melakukan review baik sebagian atau seluruhnya dan melakukan praktik professional. Dari hasil penelitiannya didapatkan bahwa semua metode yang digunakan dapat memberikan motivasi buat pengajar pemrograman untuk dapat meningkatkan

kemampuan mengajarnya dan menyesuaikan strategi mengajarnya sesuai dengan generasi siswa yang modern. Strategi pembelajaran yang diusulkan oleh Kheleel, dkk (2017), lebih fokus pada mengidentifikasi kebutuhan pembelajaran pemrograman berdasarkan masalah pembelajaran yang tidak efektif dan kurangnya minat dan motivasi mengikuti pembelajaran pemrograman, dan memverifikasinya melalui dua perspektif (ahli dan siswa). Berdasarkan hasil analisa yang didapatkan dari ahli dan mahasiswa untuk memverifikasi kebutuhan pembelajaran pemrograman, didapatkan hasil sebagai berikut: (1) Secara umum ahli dan mahasiswa memberikan umpan balik setuju terhadap semua kebutuhan pembelajaran pemrograman; (2) Responden memberikan tambahan kebutuhan berupa keterampilan berpikir melalui kerjasama tim yang sangat penting bagi mahasiswa yang baru belajar konsep bahasa pemrograman, yang berguna dalam membangun dasar yang kuat dimasa yang akan datang saat mahasiswa menggunakan bahasa pemrograman yang lain. Motivasi pembelajaran melalui game rule, elemen game seperti penggunaan drag and drop dan soal multiple choice dapat digunakan. Sebagai kesimpulan, semua kebutuhan pembelajaran pemrograman mengindikasikan penerimaan yang positif dilihat dari 2 perspektif. Beberapa saran untuk meningkatkan kemampuan pemrograman adalah mahasiswa harus berlatih sesering mungkin, contoh dan solusi harus disediakan lebih banyak, mahasiswa harus diberikan berbagai macam jenis tugas, dan menyediakan lingkungan belajar mahasiswa yang lebih menyenangkan dan lebih santai. Sekumpulan kebutuhan pembelajaran bahasa pemrograman didapatkan, seperti: (1) Real-time reaction; (2) Response to event; (3) Simplicity, power; (4) cognitive efficiency; (5) practices; (6) motivation through game rules; (7) used updated reference and basic concepts; (8) motivation students through fun game elements. Serupa dengan dua penelitian sebelumnya, Sakibayev dkk, (2018) juga melakukan hal yang sama namun dengan cara yang berbeda. Tujuan penelitian yang dilakukan adalah untuk menemukan faktor-faktor yang dapat memberikan dampak yang signifikan pada pengembangan kemampuan programming siswa dan kemudian memformulasikan hipotesis yang disajikan sebagai pondasi teoritis untuk faktor-faktor tersebut. Peneliti mengamati praktek dan kebiasaan belajar yang dilakukan oleh siswa yang memiliki kemampuan yang tinggi dalam pemrograman dan membandingkannya dengan kebiasaan dan praktek siswa dengan kemampuan yang rendah dalam pemrograman. Kemudian menggunakan hasil observasi tersebut untuk memformulasikan teori dan prinsip pendidikan yang dapat diterapkan pada pengajaran pemrograman untuk pengembangan kemampuan pemrograman yang efektif. Terdapat lima hipotesis yang didapatkan dari pengumpulan data, yaitu: (1) memecahkan masalah matematika adalah cara yang paling efektif dalam meningkatkan kemampuan pemrograman, karena pemrograman komputer memiliki pondasi matematika, meskipun tidak semua masalah matematika bisa meningkatkan kemampuan pemrograman; (2) minat pada sains adalah salah satu syarat untuk pengembangan kemampuan pemrograman siswa yang efektif. Membaca buku sains dan menyelesaikan masalahnya, memerlukan penalaran logis dan analitik, mengetahui inti dan struktur dari sesuatu, dan mengerti hubungan sebab dan akibat. Kebanyakan siswa dengan kemampuan pemrograman yang tinggi suka menonton dan membaca buku sains; (3) membaca sejarah teknologi komputer juga sangat penting dalam pengembangan kemampuan pemrograman. Kebanyakan siswa dengan kemampuan pemrograman yang tinggi menghabiskan waktunya dalam membaca dan menonton film dokumenter tentang berbagai topik yang berkaitan dengan sejarah teknologi computer; (4) memecahkan masalah catur dapat meningkatkan kemampuan pemrograman siswa. Mayoritas siswa dengan kemampuan pemrograman yang tinggi tergabung dalam perkumpulan catur. (5) Menulis essay sederhana dalam topik ilmiah dapat meningkatkan kemampuan pemrograman siswa. Mayoritas siswa dengan kemampuan pemrograman tinggi dapat menulis essay sederhana dalam topik ilmiah. Selain hipotesis diatas peneliti

menerapkan metode spaced learning dalam proses pembelajaran untuk meningkatkan kemampuan pemrograman siswa. Hasil penelitian menemukan bahwa kelompok eksperimental yang terdiri dari 8 orang siswa dengan kemampuan pemrograman rendah memiliki nilai rata-rata C. dari verifikasi percobaan dilakukan pada minggu ke 6 tahun akademik siswa menunjukkan perkembangan awal kemampuan pemrograman komputer. Mulai pada minggu ke 10, siswa menunjukkan peningkatan perkembangan yang signifikan dilihat dari sebagian siswa telah mendapatkan nilai A atau A-. Sejalan dengan itu siswa menunjukkan peningkatan akademik pada matapelajaran yang lain. Spaced learning jika diterapkan pada pendidikan pemrograman komputer sangat efektif dalam meningkatkan nilai matapelajaran, karena pengulangan beberapa kali dalam pembelajaran memungkinkan siswa menyerap materi dengan usaha yang minimal. Marietjie (2018), melakukan pendekatan lain, yaitu menerapkan framework Cooperative, Responsibility dan Dependency (CRD) dalam konteks Problem-based learning (PBL) untuk pembelajaran pemrograman dan mengujinya menggunakan data penelitian yang didapatkan. Peneliti ingin mencari keterampilan dan kompetensi apasaja yang diterapkan oleh siswa dalam mengelola CRD timnya dalam proyek pemrograman. Framework kompetensi CRD dan hubungannya dengan keterampilan PBL yang diusulkan adalah kompetensi cooperative berhubungan dengan praktek yang berpusat pada siswa dan bertanggung jawab pada konstruksi pemecahan masalah yang sebenarnya, kompetensi cognitive berhubungan dengan pembelajaran aktif dan mendukung konstruksi pengetahuan, kompetensi metacognitive berhubungan dengan pengembangan keterampilan PBL, kompetensi kapabilitas berhubungan dengan penggunaan kelompok kecil, dan kompetensi strategik berhubungan dengan penyebutan masalah dunia nyata sebenarnya dan dari berbagai segi. Temuan dalam penelitian ini adalah PBL memiliki potensi untuk membantu perkembangan pembelajaran aktif, mendukung konstruksi pengetahuan dan sangat berguna dalam mengembangkan kemampuan yang krusial dalam mendapatkan pekerjaan. Meskipun siswa mengalami beberapa tantangan dan bantuan, siswa sangat aktif terlibat dalam mengatur timnya juga dalam hal mengerjakan tugas. Sebagai tambahan temuan dalam penelitian ini adalah perlunya kebutuhan dalam optimalisasi tanggungjawab dalam kerja tim. Temuan dalam penelitian ini sejalan dengan penelitian yang dilakukan oleh Kim (2019) yang bertujuan mengkaji creative problem-based learning (cPBL), flipped learning dan hubungan antara kreativitas dan penulisan kode program. Kim menyatakan bahwa tujuan dari pendidikan software coding jangan hanya fokus pada elemen gramatikal dari bahasa pemrograman, namun juga harus focus pada pengembangan kreativitas, mendefinisikan masalah dan memecahkannya, kemampuan mengarahkan diri sendiri dan bekerjasama. Hasil dari penelitian Kim menyatakan bahwa pembelajaran kolaboratif dan ditambah self-directed practice pada cPBL didalam kelas dapat membantu siswa lebih dalam memahami konsep pemrograman. Meskipun demikian, masih banyak siswa yang menganggap software coding terlalu sulit untuk meningkatkan kepercayaan diri pada kreatifitas pemrograman sebuah kode dalam satu semester. Oshanova, dkk (2019), menjelaskan bahwa untuk meningkatkan kemampuan berpikir adalah dengan berpikir secara komputasional atau berpikir secara algoritmik. Tujuan dari penelitiannya adalah mempelajari cara memecahkan masalah algoritmik sederhana menggunakan cara berpikir algoritmik. Untuk memotivasi dan meningkatkan efisiensi dalam mempelajari ilmu komputer, diciptakan game moment saat memecahkan masalah dari materi pemrograman dasar. Hal ini juga sejalan dengan apa yang dilakukan oleh khaleel, dkk dalam memotivasi siswa dalam belajar. Selain itu, penggunaan game moment dapat meningkatkan kualitas keterampilan siswa berdasarkan minat kognitifnya. Sebagaimana yang dijelaskan oleh Szlavi dan Zsako (2003), Sakibayev dkk, (2018), dan Oshanova, dkk (2019), bahwa salah



satu strategi atau metode dalam mempelajari pemrograman adalah dengan menguasai matematika. Belajar matematika dapat meningkatkan kemampuan berpikir komputasional (CT). Kong (2020), melakukan penelitian tentang pengembangan guru dalam pembelajaran dan desain berpikir komputasional yang bertujuan untuk menjawab 4 rumusan permasalahan. Adapun rumusan permasalahan itu adalah 1) bagaimanakah partisipasi dalam program pengembangan guru mempengaruhi pemahaman guru pengajar terhadap konsep computational thinking; 2) bagaimanakah partisipasi dalam program pengembangan guru mempengaruhi pemahaman guru pengajar terhadap praktek computational thinking; 3) bagaimanakah partisipasi dalam program pengembangan guru mempengaruhi guru pengajar pada 4 materi-pengetahuan yang berhubungan dengan dimensi Technological Pedagogical Content Knowledge (TPACK) dalam pengajaran CT melalui pendidikan pemrograman; 4) bagaimanakah guru pengajar merasakan perkembangan mereka sepanjang program, yang diartikulasikan dalam refleksi pasca-program mereka. Penelitian ini memberikan hasil positif bagi tiga rumusan masalah penelitian, pertama ditemukan bahwa kegiatan ini secara signifikan menstimulasi dan terus meningkatkan pengetahuan peserta terhadap praktek dan konsep CT dengan dampak yang besar. Kedua, kemajuan yang signifikan ditemukan untuk semua CT yang berhubungan dengan dimensi TPACK, dengan dampak yang luas untuk peningkatan yang terus-menerus pada Content Knowledge (CK), Technological Content Knowledge (TCK) dan TPACK. Salah satu kontribusi yang signifikan dari penelitian ini adalah validasi empiris terhadap program pengembangan guru yang efektif dalam CT yang berhubungan dengan pemrograman. Hasil penelitian ini juga sejalan dengan penelitian sebelumnya, bahwa TPACK dinamis dan context-sensitive. Penelitian ini berkontribusi terhadap desain program yang sama di semua daerah, dalam merespon kekhawatiran global terhadap bagaimana menyediakan pengembangan profesionalisme yang efektif bagi guru dalam mengajarkan CT yang berkaitan dengan pemrograman. Meskipun demikian, Pedagogical Content Knowledge (PCK) merupakan satu-satunya dimensi yang tidak mengalami peningkatan dalam framework TPACK, sehingga perlu penekanan yang lebih dalam kedepannya, khususnya pada Teacher Development Course 2 (TDC2). Untuk pengembangan PCK pada guru selanjutnya, peneliti menyarankan TDC lebih menitikberatkan pada bagaimana menuntun siswa berfikir sebelum menulis kode program, seperti penggunaan cerita sederhana untuk menjelaskan konsep CT.

### **Strategi pembelajaran pemrograman dilihat dari metode lainnya**

Garcia dan De la rosa (2016) melakukan penelitian yang bertujuan untuk mengevaluasi interaksi dari sekelompok siswa sekolah menengah kolombia menggunakan aplikasi yang dinamakan RoBlock dalam konteks mobile robotics untuk mempelajari algoritma dan pemrograman. Dalam penelitiannya, digunakan beberapa metode pendekatan pengajaran algoritma dan pemrograman yang telah diusulkan seperti pendekatan desain game, pengembangan storytelling, dan jenis pendekatan lain. Meskipun pendekatan ini menawarkan solusi untuk konteks tertentu, pendekatan ini terbatas kepada akuisisi kemampuan bahasa pemrograman tertentu dan menggunakan aplikasi khusus yang juga memerlukan konfigurasi teknis khusus. Metode pendekatan diatas hanya merupakan pelengkap dari pembelajaran, sehingga siswa masih membutuhkan pengajar untuk dapat belajar optimal. Peneliti menciptakan aplikasi yang dinamakan RoBlocks V.1. Aplikasi ini dikembangkan untuk pembelajaran algoritma dan pemrograman yang menyediakan mekanisme untuk mengerjakan dan memvalidasi query. Aplikasi ini kemudian dibandingkan dengan aplikasi scratch untuk melihat apakah aplikasi RoBlocks dapat digunakan dalam mempelajari algoritma dan pemrograman. Temuan dalam penelitian

menunjukkan bahwa siswa yang bekerja menggunakan RoBlock menunjukkan interval antara 70% sampai 106%, dengan rata-rata 91% dengan waktu interaksi selama lima jam. Disamping itu, hasil yang menarik juga menunjukkan adanya hubungan antara performa siswa yang bekerja dengan RoBlocks dan konsep pembelajaran pemrograman. Berbeda dengan penelitian yang dilakukan oleh Garcia dan De la Rosa yang mengembangkan aplikasi berbasis web, Issong (2014), mengambil pendekatan pengembangan rekayasa perangkat lunak dalam mengajarkan pemrograman computer. Penelitiannya bertujuan untuk menggunakan proses Agile dalam pengajaran pemrograman untuk siswa tahun pertama. Sasarannya adalah untuk membantu pemula dalam mengembangkan kemampuan pemrogramannya, mengusulkan teknologi pengajaran yang dapat memaksimalkan peluang keterlibatan siswa, meningkatkan pengajaran sebagai refleksi oleh guru terhadap materi yang mereka ajarkan dan apasaja yang dipelajari oleh siswa. Pendekatan baru dalam pengajaran pemrograman komputer untuk pemula, menggunakan prinsip agile dan praktek pemrograman berpasangan. Berikut adalah integrasi antara manifesto agile dengan metodologi pengajaran pemrograman yang diusulkan:

- *Individual and interactions over process and tools sejalan dengan pair programming*
- *Working software over comprehensive documentation sejalan dengan coding over passive listening*
- *Customer collaboration over contract negotiation sejalan dengan active teacher/tutor involvement*
- *Respond to changes over following plan sejalan dengan course content flexibility and immediate corrections*

Untuk memudahkan pemahaman dan pembacaan, hasil penelitian dideskripsikan terlebih dahulu, dilanjutkan bagian pembahasan. Subjudul hasil dan subjudul pembahasan disajikan terpisah. Bagian ini harus menjadi bagian yang paling banyak, minimum 60% dari keseluruhan badan artikel.

## **Pembahasan**

Strategi pembelajaran pemrograman dapat dilakukan dengan berbagai metode, teknik dan taktik, yang semuanya bertujuan selain untuk meningkatkan pengetahuan siswa dan juga dapat membantu pengajar dalam mengembangkan dan meningkatkan kemampuan pedagogiknya ke tingkat yang lebih tinggi. Dalam proses pembelajaran pemrograman yang fokus pada pendekatan elemen-elemen atau materi dalam kuliah pemrograman, sebaiknya memilih salah satu atau menggabungkan beberapa metode yang diusulkan oleh Szlavi dan Zsako (2003). Meskipun terdapat kekurangan dalam penerapannya, yaitu tidak dijelaskannya model pembelajaran dalam penerapan strategi pembelajarannya. Adapun untuk strategi pembelajaran pemrograman untuk meningkatkan kemampuan pedagogik siswa dan pengajar, dapat menggunakan model pembelajaran interaktif seperti problem-based learning, collaborative learning, creative learning, spaced learning dan self-directed learning (Djenic dan Mitic, sakibayev dkk, Marietjie, Khaleel, dkk, Kim). Pembelajaran interaktif sangat membantu dalam mentransformasi peran pengajar dari translator informasi menjadi koordinator atau organizer proses pendidikan dan memungkinkan pembentukan kompetensi kompleks untuk spesialisasi profesional dimasa yang akan datang, melalui aktifitas kerja siswa yang mirip dengan pekerjaan professional (Yakovleva dan Yakovlev, 2014). Meskipun demikian, Kim dalam penelitiannya mendapatkan hasil

*Inverted: Journal of Information Technology Education, Vol 3, No 2, Juli 2023*

e-ISSN: 2828-3880, p-ISSN: 2828-2094

terdapat beberapa siswa masih merasa kesulitan dalam mempelajari pemrograman. Untuk mengasah kemampuan siswa dalam memecahkan masalah dalam pemrograman, sebaiknya menggunakan strategi pembelajaran yang diusulkan oleh Kong, dkk serta Oshanova, yaitu dengan menggunakan kemampuan berpikir komputasional atau algoritmik. Meningkatkan kemampuan berpikir komputasional terbukti dapat membantu siswa dan pengajar dalam meningkatkan minat kognisinya. Adapun kekurangan dalam penerapan strategi ini adalah tidak adanya pengaruh terhadap perkembangan pedagogi pengajar. Bagi siswa yang berlatar belakang non-IT atau siswa yang baru pada tingkat pertama, sebaiknya pembelajaran berjenjang seperti yang dilakukan oleh Mironova, dkk. Materi yang diberikan pada awal pelajaran adalah pelajaran dasar-dasar komputer seperti aplikasi pengolah kata, presentasi dan spreadsheets, kemudian dilanjutkan dengan praktik pemrograman visual menggunakan tools scratch untuk memberikan pemahaman konsep dan terminologi pemrograman serta tidak berorientasi pada sintaks program, dan yang terakhir belajar sintaks bahasa pemrograman menggunakan VB atau python. Strategi pembelajaran pemrograman lainnya yang dapat diterapkan adalah pendekatan pengembangan perangkat lunak (RPL) seperti yang diusulkan oleh Issong. Manifesto Agile pada RPL di hubungkan dengan teknik dan taktik dalam pengajaran pemrograman untuk mengoptimalkan interaksi siswa dan pengajar didalam kelas. Strategi ini memiliki kelemahan, yaitu sulit untuk diterapkan karena dibutuhkan fasilitas yang lengkap dan kehadiran pengajar untuk menuntun siswa setiap saat dalam kelas. Teknik pembelajaran pemrograman lainnya yang dapat digunakan dalam membantu meningkatkan kemampuan siswa dalam pemrograman adalah dengan bantuan tools pembelajaran seperti Scratch atau RoBlock yang berbasis web (Garcia dan De la rosa). Aplikasi tersebut digunakan sebagai pelengkap dalam proses pembelajaran untuk meningkatkan minat dan performa siswa pada tingkat dasar atau yang baru belajar pemrograman. Untuk meningkatkan hasil pembelajaran, perlu dikombinasikan strategi-strategi pembelajaran diatas, sehingga hasil yang didapatkan lebih optimal.

Meskipun semua strategi pembelajaran diatas sebagian besar berhasil meningkatkan keterampilan dan kemampuan siswa serta pengajar dalam pemrograman, namun semua strategi tersebut hanya menjelaskan tentang peningkatan *hardskill* dalam bidang pemrograman, tidak menjelaskan tentang peningkatan *softskill* secara implisit. *Softskill* sangat dibutuhkan oleh siswa, karena akhir dari pendidikan adalah dunia kerja. Ahmed, dkk (2013), menjelaskan bahwa *softskill* yang dibutuhkan industri untuk programmer ada 7 keterampilan. Ketujuh keterampilan tersebut adalah (1) terbuka dan dapat beradaptasi terhadap perubahan; (2) kemampuan untuk bekerja secara mandiri; (3) dapat bekerja dalam kelompok; (4) memiliki keterampilan organisasional; (5) memiliki keterampilan dalam menganalisa dan memecahkan masalah; (6) memiliki keterampilan interpersonal; dan yang paling keterampilan yang paling dibutuhkan adalah (7) keterampilan komunikasi. Diperlukan pembahasan secara implisit bagaimana mengintegrasikan *softskill* kedalam strategi pembelajaran dan penerapannya.

Terdapat beberapa artikel yang membahas tentang strategi pembelajaran pemrograman namun tidak dibahas dalam artikel ini. Dengan melakukan ikhtisar tentang strategi-strategi pembelajaran pemrograman, dapat membantu pengajar dalam memilih strategi yang cocok dalam pengajaran pemrograman, sehingga mempermudah siswa dalam memahami materi serta dapat memotivasi dan meningkatkan hasil belajar siswa.

## SIMPULAN

Kesulitan dan bahkan kegagalan siswa dalam mempelajari pemrograman dapat disebabkan oleh pemilihan strategi pengajaran yang tidak sesuai. Dalam artikel ini, dijelaskan tentang beberapa strategi pembelajaran pemrograman yang dapat diterapkan untuk meningkatkan kemampuan dan keterampilan pemrograman siswa dan pengajar. Strategi tersebut dapat dilihat dari orientasi elemen-elemen pemrograman, strategi pembelajaran dilihat dari pedagogiknya, dan strategi pembelajaran dilihat dari metode lainnya. Setiap strategi yang dijelaskan memiliki kelebihan dan kekurangan dalam mengimplementasikannya, oleh karena itu, untuk mengoptimalkan hasil pembelajaran perlu untuk mengkombinasikan strategi-strategi tersebut sesuai dengan tempat dan keadaan. Semua strategi pembelajaran yang dijabarkan dalam artikel-artikel yang di review, hanya menyangkut hardskill dan tidak menjabarkan softskill yang dibutuhkan industry. Oleh karena itu penelitian kedepannya perlu untuk membahas tentang bagaimana integrasi softskill kedalam strategi pembelajaran dan penerapannya.

## DAFTAR PUSTAKA

- Ahmed, F., Capretz, L. F., Bouktif, S., and Campbell, P. (2013). Softskill and Software Development: A Reflection from Software Industry. *International Journal of Information Processing and Management*. Vol 4 No 3 pp 171-191 <http://www.researchgate.net/publication/25647867>. Diakses 8 Juli 2020
- Berau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Software Developers. Tersedia pada laman: <http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>. diakses 7 Juli 2020
- Berau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Computer Programmers. Tersedia pada laman: <http://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm>. diakses 7 Juli 2020
- Demilie, W. B., (2019). Causes of failure of University students in computer programming courses: the case of Wachemo University. *International Journal of Scientific Research in computer science, engineering and information technology*. Vol. 5. Issue 5. pp. 123-132.
- Djenic, S., and Mitic J. (2017). Teaching Strategies and Methods in Modern Environments for Learning of Programming. 14<sup>th</sup> International Conference of Cognition and Exploratory Learning in Digital Age
- Garcia, P.G.F., and De la rosa, F. (2016). RoBlock – Web App for Programming Learning. *iJET*. Vol. 11 Issue 12 <https://doi.org/10.3991/ijet.v11i12.6004>
- Issong, Bassey. (2016). A Methodology for Teaching Computer Programming: First Year Students Perspective. *International Journal of Modern Education and Computer Science*. Vol 9 pp 15-21 DOI: 10.5815/ijmeecs.2014.09.03
- Janpla, S., and Piriyasurawong, P. (2018). The development of problem-based learning and concept mapping using a block-based programming model to enhance the competency of under graduated students in computer science. *TEM Journal*, Vol. 7, Issue 4, pp. 708-716
- Kim, Geun-Hyung. (2019). A Learning Model for Software Coding Education. *Journal of Problem Based-Learning*. Vol 6 No. 2 pp 67-75 <https://doi.org/10.24313/jpbl.2019.00164>
- Khaleel, F. L., Ashaari N. S., Wook, T. S. M. T., and Ismail A., (2017). Programming Learning Requirements Based on Multi Perspectives. *International Journal of Electrical and Computer Engineering*. Vol. 7 No. 3 pp 1299-1307

- Kong, S-C., Lai M., and Sun D. (2020). Teachers Development in Computational Thingking: Desaign and Learning Outcomes of Programming Concepts, Practices, and Pedagogy. *Computer & Education*. Vol. 151 <https://doi.org/10.1016/j.compedu.2020.103872>
- Lahtinen, E., Ala-Mutka, K., and Jarvinen, H-M., (2005). A Study of Difficulties of Novice Programmers. Tersedia di laman <http://www.researchgate.net/publication/220808194>. Diakses 8 Juli 2020
- Lateef, U. O., Ogunsanwo, G., Owoade, A. (2016). Introduction to computer programming (BASIC). Tersedia di laman <http://www.researchgate.net/publication/317182495>. Diakses 8 Juli 2020
- Marietjie Havenga (2018) Problem-based Projects in Computer Programming: Students' Cooperation, Responsibilities and Dependencies, *African Journal of Research in Mathematics, Science and Technology Education*, 22:2, 254-264, DOI: 10.1080/18117295.2018.1483596
- Mironova O., Amitan I, Vilipold J., dan Saar M. (2016). Active Learning Methods in Programming for Non-IT Students. *International Conference e-Learning*
- Sheuermans S. and Voskoglou C. (2019). The Global Developer Population 2019. Tersedia pada laman: <http://sdata.me/GlobalDevpop19>. Diakses 7 juli 2020.
- Snyder, Hannah. (2019). Literature Review as a research methodology: an overview and guidelines. *Journal of Business Research*. Vol. 104. pp 333-339
- Szlávi, Péter & Zsakó, László. (2003). Methods of teaching programming. *Teaching mathematics and Computer Science*. 1. 247-258. 10.5485/TMCS.2003.0023.
- Torraco, R. J. (2005). Writing Integrative Literature Reviews: Guidelines and Examples. *Human Resource Development Review*. Vol. 4. pp 356-367