

Pelatihan “The Fundamental of Smart Programming” PKBM Bina Karya Semarang

Indra Gamayanto¹, Hanny Haryanto², Sendi Novianto³

^{1,2,3}Departemen Sistem Informasi & Teknik Informatika, Fakultas Ilmu Komputer, Universitas
Dian Nuswantoro

E-mail: ¹indra.gamayanto@dsn.dinus.ac.id,

²hanny.haryanto@dsn.dinus.ac.id, ³sendi.novianto@dsn.dinus.ac.id

Abstrak

Bahasa program menjadi salah satu hal terpenting dalam pengembangan teknologi informasi, dimana Bahasa program akan dalam menghasilkan banyak hal yang bermanfaat bagi masyarakat. Pada pengabdian ini kita akan berusaha untuk meningkatkan kemampuan dalam Bahasa program, sehingga murid-murid akan dapat menghasilkan kompetensi yang mampu dalam menghadapi globalisasi dan perkembangan teknologi informasi yang sangat cepat. Kerangka kompetensi harus dapat diciptakan agar dapat menghasilkan sumber daya manusia yang memiliki kemampuan tinggi, dan hal ini harus dimulai dari dasar, bukan dari puncaknya. Pengabdian ini akan berfokus pada penerapan dasar programming yang inovatif dan dapat cepat dipelajari, hal ini dinamakan the fundamental of smart programming. Hasil dari pengabdian ini, murid akan dapat memiliki kemampuan dalam menggunakan Bahasa program dan dapat langsung diterapkan pada kehidupan mereka. Mereka juga akan mampu dapat menciptakan dari apa yang sudah mereka pelajari, sehingga mereka akan mampu mengembangkan apa yang sudah dipelajarinya.

Kata kunci: Smart programming, Inovasi, Implementasi, Kreatif, Penerapan

Abstract

The program language is one of the most important things in the development of information technology, where the program language will produce many things that are beneficial to society. In this service we will try to improve skills in the language of the program, so that students will be able to produce competencies capable of dealing with globalization and the very rapid development of information technology. A competency framework must be created in order to produce highly capable human resources, and this must start from the bottom, not the top. This service will focus on applying the basics of programming that are innovative and can be quickly learned, this is called the fundamentals of smart programming. As a result of this dedication, students will be able to have the ability to use the program language and can directly apply it to their lives. They will also be able to create from what they have learned, so they will be able to develop what they have learned.

Keywords: Smart programming, Innovation, Implementation, Creative, Application

1. PENDAHULUAN

Programming merupakan sesuatu yang penting dalam menghadapi perubahan dan perkembangan teknologi infomarmasi yang sedemikian cepatnya. Oleh karena itu, sangatlah penting untuk dapat memahami programming sebagai sebuah keahlian dalam menghadapi kecepatan artificial intelligence dan perkembangan lainnya. Pada pengabdian masyarakat ini, kami berfokus untuk mengembangkan anak- anak ke dalam pengetahuan programming yang sangat penting dan pembelajaran ini akan dapat meningkatkan kompetensi anak-anak tersebut dalam menghadapi kebutuhan industri dan kebutuhan dalam menghadapi era globalisasi dengan kecepatan tinggi.

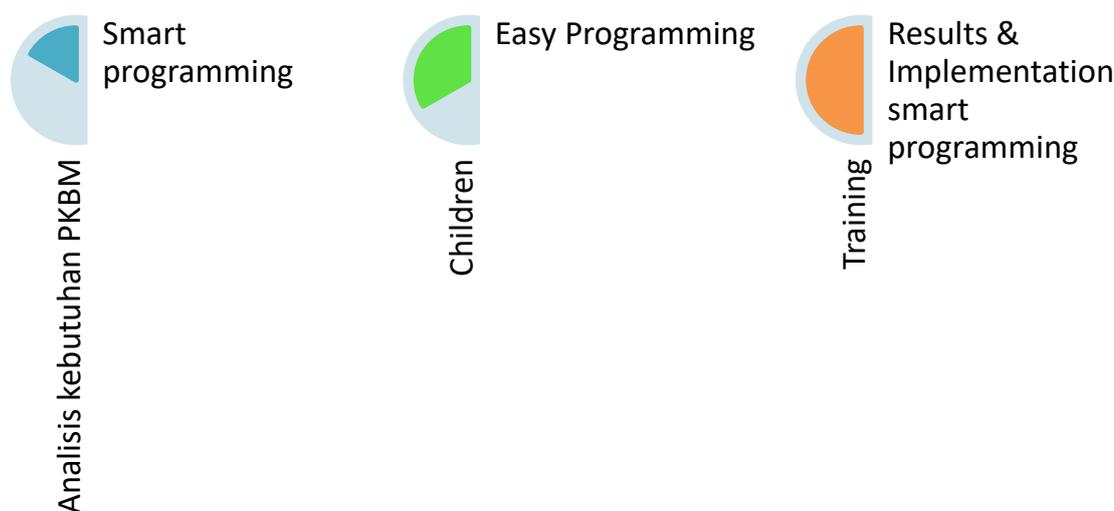
Programming mempunyai banyak manfaat, antara lain: (1) Pengkodean Meningkatkan

Kreativitas. Coding dan pemrograman adalah aktivitas yang tepat bagi anak-anak untuk mengembangkan kreativitasnya. Mereka mulai memvisualisasikan desain, seni, dan musik sambil membuat situs web, aplikasi, permainan, dll. Anak-anak belajar menemukan solusi baru untuk tugas-tugas asing. Mereka beradaptasi dengan brainstorming, pemikiran luas dan konseptualisasi. Yang terpenting, rasa pencapaian yang mereka rasakan setelah menciptakan sesuatu yang unik sungguh tak ternilai harganya[1], (2) Pengkodean Mendorong Pemikiran Logis-Pengkodean meningkatkan sifat berpikir logis pada anak Anda, yang dapat diterapkan baik dalam kehidupan pribadi maupun pekerjaan. Berpikir logis adalah menganalisis situasi atau masalah dan mencari solusi potensial. Dalam pengkodean dan pemrograman, komponen kuncinya adalah dekomposisi, yang berarti – memecah suatu masalah kompleks menjadi bagian-bagian kecil yang lebih mudah dipahami dan diselesaikan. Setiap tugas memerlukan penguraian suatu tugas menjadi bagian-bagian yang lebih kecil. Anak-anak perlahan menyadari bahwa untuk mencapai hal-hal yang lebih besar, mereka perlu mengambil langkah-langkah kecil terlebih dahulu[2], (3) Pengkodean Membangun Keyakinan- Kebanyakan orang tua pernah merasakan bahwa anak mereka mendapatkan kepercayaan diri yang luar biasa selama proses pembelajaran coding. Mereka telah melihat perkembangan kepribadian anaknya secara keseluruhan. Mulai dari belajar coding, berinteraksi dengan guru, aktif mengerjakan proyek hingga menciptakan sesuatu yang baru, para siswa sangat menikmati prosesnya[3], (4) Coding Mengajarkan Berpikir Komputasi-Berpikir komputasional mengacu pada merancang ide untuk memecahkan masalah. Ini mengajarkan siswa bagaimana berpikir kritis dan logis. Berpikir komputasional mencakup serangkaian keterampilan dan proses yang memungkinkan siswa menavigasi masalah yang kompleks. Proses ini mempermudah penyelesaian masalah besar secara terstruktur. Saat coding, ketika siswa tidak menemukan solusi untuk suatu masalah, mereka mengubah polanya dan mencoba menyelesaikannya lagi. Ini menyajikan cara-cara partikel untuk mendekati dan menangani masalah. Mempelajari pemikiran komputasi membangun keterampilan metakognitif. Anak-anak pada usia yang sangat muda belajar bagaimana berpikir dengan benar[3],[4], (5) Coding Membangun Keterampilan Bercerita-Menciptakan sesuatu secara digital pada dasarnya adalah menceritakan sebuah kisah. Melalui coding, kami menciptakan cerita untuk orang-orang. Scratch, merupakan bahasa coding visual yang diajarkan kepada siswa pada mata kuliah coding. Hal ini memungkinkan anak-anak membuat permainan, cerita digital, animasi, dll. Bercerita adalah bagian penting dalam mengembangkan cerita animasi atau permainan, dll. Untuk merencanakan cerita, siswa terlibat dalam berbagai latihan curah pendapat. Hal ini juga memungkinkan mereka untuk menceritakan kisah mereka sendiri. Dengan menggunakan Scratch sebagai alat bercerita digital, siswa dapat membuat puisi, cerita pendek, mengilustrasikan soal matematika, dan banyak lagi. Dengan membaca, bertindak, atau memecahkan masalah dalam coding dan pemrograman, anak Anda juga dapat belajar bagaimana menyelesaikannya dalam kehidupan nyata[5], (6) Pengkodean Meningkatkan Kinerja Akademik-Siswa yang belajar coding dapat mengembangkan berbagai keterampilan untuk tujuan akademis mereka termasuk menulis, matematika, komputer, kreativitas, dan kegiatan ekstrakurikuler. Ini membantu meningkatkan kepercayaan diri mereka di sekolah dan keterlibatan sosial lainnya. Selain itu, coding mendorong pengembangan keterampilan penting, seperti pemecahan masalah, penyusunan strategi, perencanaan, pemikiran matematis, dll. Berbagai keterampilan tersebut akan membantu anak-anak dalam perjalanan akademis mereka[6], (7) Pengkodean Mengembangkan Kesabaran-Kesabaran adalah salah satu kualitas hebat yang harus dimiliki seseorang jika ingin mencapai sesuatu yang besar dalam hidup. Pengkodean membantu mengembangkan kualitas ini pada anak Anda sejak usia sangat muda. Pengkodean membutuhkan banyak kesabaran. Anda tidak akan membuat sesuatu pada saat pertama kali melakukannya. Mungkin ada perpecahan dan kegagalan. Beberapa langkah pertama akan didasarkan pada proses coba-coba. Hal ini secara keseluruhan mengembangkan kesabaran yang luar biasa pada anak-anak. Mereka termotivasi untuk terus melakukannya sampai mereka mendapatkan kesuksesan[7], (8) Coding Mengembangkan Pemikiran Algoritma-Berpikir algoritmik adalah menciptakan serangkaian langkah untuk menyelesaikan suatu tugas yang dapat diulangi oleh orang lain. Ini adalah metode yang secara jelas mendefinisikan prosedur yang diperlukan. Pada dasarnya, saat membuat algoritma, langkah-langkah dalam menyelesaikan

masalah jelas, efisien, dan terdefinisi dengan baik. Ini melatih siswa untuk dapat membayangkan dan dengan jelas mendefinisikan langkah-langkah individu yang diperlukan untuk menyelesaikan suatu tugas. Tidak diragukan lagi, coding memberdayakan anak-anak untuk belajar dengan menyenangkan. Hal ini juga menginspirasi mereka untuk berinteraksi dengan teknologi dengan cara yang positif dan produktif[8].

Hasil dari pengabdian ini adalah anak-anak mampu dalam mengembangkan kemampuan programmingnya dan meningkatkan kompetensinya, sehingga apa yang sudah dipelajari akan dapat membantu mereka di masa depan.

2. METODE



Gambar 1. Proses pengabdian masyarakat the fundamental of smart programming

Gambar 1, menjelaskan proses pengabdian masyarakat the fundamental of smart programming, dimana pertama-tama kita melakukan diskusi dengan pihak PKBM, untuk melanjutkan pelatihan programming kita menjadi smart programming, dimana tingkatan pelatihan ini menjadi lebih tinggi dan berada di tingkat intermediate. Selanjutnya, kita sudah menentukan bahwa akan diterapkan pelatihan smart programming yang berguna untuk meningkatkan kompetensi anak-anak, dan terakhir adalah hasil dan cara anak-anak tersebut mengimplementasikan apa yang sudah dipelajari

Adapun, kami melakukan survey terhadap 40 orang (meliputi guru yang berada di PKBM dan anak-anak yang mengikuti pelatihan untuk mengetahui atau memiliki indicator apakah pengabdian masyarakat ini sudah mencapai sasaran yang diinginkan)

1. Apakah anda tertarik untuk mempelajari Bahasa program?
 - a. Tidak
 - b. Tidak mau menjawab
 - c. Tertarik, tetapi pada game
 - d. Tertarik, tetapi kesulitan dalam memahaminya
 - e. Tertarik, tetapi membutuhkan dukungan dalam mengembangkannya

2. Apakah Anda merasa bahwa programming adalah kebutuhan utama?
 - a. Tidak
 - b. Tidak mau menjawab
 - c. Kami merasa, programming merupakan sebuah kebutuhan, tetapi terkendala masalah fasilitas
 - d. Kami merasa, programming merupakan sebuah kebutuhan, tetapi kami membutuhkan dukungan dana untuk pelatihan yang lebih tinggi
 - e. Kami merasa, programming merupakan sebuah kebutuhan, tetapi kami membutuhkan dukungan dana untuk mentoring

3. Apakah anda merasa bahwa programming akan memiliki masa depan yang baik?
 - a. Tidak
 - b. Tidak mau menjawab
 - c. Programming akan memiliki masa depan baik khususnya di bidang IT
 - d. Programming masih dipertanyakan perkembangannya karena masih belum adanya dukungan fasilitas yang memadai di sekolah
 - e. Programming memiliki masa depan yang jelas dan stabil, karena pasti akan selalu dibutuhkan

Notes: Dalam menjawab pertanyaan-pertanyaan, ini anak-anak di dampingi oleh guru dan pelatih untuk dapat menjawabnya

3. HASIL DAN PEMBAHASAN

3.1. Hasil Survey

Tabel 1. Hasil survey, indicator pengabdian smart programming

Pertanyaan	Pilihan	Hasil	Indikator
1. Apakah anda tertarik untuk mempelajari Bahasa program?	a. Tidak	0	Dominan pada pilihan c
	b. Tidak mau menjawab	2	
	c. Tertarik, tetapi pada game	25	
	d. Tertarik, tetapi kesulitan dalam memahaminya	10	
	e. Tertarik, tetapi membutuhkan dukungan dalam mengembangkannya	3	
2. Apakah Anda merasa bahwa programming adalah kebutuhan utama?	a. Tidak	0	Dominan pada pilihan e
	b. Tidak mau menjawab	1	
	c. Kami merasa, programming merupakan sebuah kebutuhan, tetapi terkendala masalah fasilitas	12	
	d. Kami merasa, programming merupakan sebuah	10	

	kebutuhan, tetapi kami membutuhkan dukungan dana untuk pelatihan yang lebih tinggi e. Kami merasa, programming merupakan sebuah kebutuhan, tetapi kami membutuhkan dukungan dana untuk mentoring	17	
3. Apakah anda merasa bahwa programming akan memiliki masa depan yang baik?	a. Tidak b. Tidak mau menjawab c. Programming akan memiliki masa depan baik khususnya di bidang IT d. Programming masih dipertanyakan perkembangannya karena masih belum adanya dukungan fasilitas yang memadai di sekolah e. Programming memiliki masa depan yang jelas dan stabil, karena pasti akan selalu dibutuhkan	0 3 15 12 10	Dominan pada pilihan c

Tabel 1, menjelaskan bahwa banyak anak-anak berada pada indicator pilihan c, untuk pertanyaan 1, artinya banyak anak-anak yang menganggap bahwa programming lebih cenderung kepada game. Pada pertanyaan 2, pilihan lebih banyak pada 3, dimana anak-anak membutuhkan lebih banyak mentoring dan dana dalam mengembangkan kemampuan programmingnya, dan pada pertanyaan 3, anak-anak lebih memilih pada pilihan c, bahwa programming selalu identic dengan peluang kerja khususnya di bidang teknologi informasi

3.2. Foto kegiatan dan contoh materi



Gambar 2. Pengabdian masyarakat the fundamental of smart programming



Gambar 3. Contoh materi the fundamental of smart programming

3.3. Implementasi smart programming

Setelah melakukan pengabdian the fundamental of smart programming, maka berikut ini adalah Pengkodean Meningkatkan Keterampilan Pemecahan Masalah. Tidak diragukan lagi, coding adalah salah satu cara paling mudah bagi anak-anak untuk meningkatkan keterampilan pemecahan masalah mereka. Namun sebelum kita membahasnya, mari kita mundur selangkah dan memperjelas istilah tersebut. Pada definisi yang paling sederhana, pemecahan masalah mengacu pada kemampuan seseorang untuk mengatasi situasi yang kompleks atau baru dengan cara yang efisien. Seseorang dengan keterampilan pemecahan masalah yang terasah dengan baik merangkai keahlian yang berbeda seperti kreativitas, kecerdasan emosional, keterampilan penelitian, kolaborasi, dan pengambilan keputusan menjadi respons yang kohesif dan efektif. Dalam konteks pemrograman, pembuat kode terus-menerus ditantang dan diminta untuk memecahkan masalah. Praktik ini mengharuskan mereka untuk memecah masalah menjadi sub-masalah yang lebih mudah dikelola, kemudian melanjutkan melalui proses berulang dalam mengidentifikasi, membuat prioritas, dan menerapkan solusi. Oleh karena itu, anak-anak mulai mengasah keterampilan pemecahan masalah mereka segera setelah mereka menerima tantangan coding pertama[9].

Penelitian mendukung hal ini. Sebuah penelitian pada tahun 2013 yang dilakukan terhadap anak usia lima dan enam tahun menemukan bahwa ketika anak-anak berpartisipasi dalam lingkungan pemrograman, mereka memiliki “peluang untuk mengembangkan konsep matematika, pemecahan masalah, dan keterampilan sosial.” Selain itu, para peneliti mencatat bahwa peserta muda dalam penelitian ini sering kali menikmati pengalaman belajar mereka dan terlibat dalam permainan dan aktivitas pemrograman sesuai usia yang diberikan kepada mereka. Meski begitu, manfaat yang diberikan mungkin berbeda-beda sesuai usia dan perkembangan kognitif relatif. Misalnya, penelitian lain – yang dilakukan pada tahun 2014, dan melibatkan anak-anak berusia empat hingga lima tahun – mencatat bahwa partisipan menunjukkan peningkatan kemampuan kognitif non-verbal, namun menunjukkan “tidak ada perbedaan yang signifikan secara statistik dalam keterampilan pemecahan masalah mereka. Penelitian ini menunjukkan bahwa meskipun anak Anda tidak ingin mempelajari coding saat dewasa, mempraktekkan coding sejak dini akan membantu mereka membangun keterampilan kognitif yang sangat mudah ditransfer. Apakah mengherankan mengapa anak-anak harus membuat kode?[10]

Seperti yang ditulis oleh para peneliti pada penelitian sebelumnya mengenai masalah ini. Kami percaya bahwa anak-anak berusia sepuluh tahun dapat memperoleh manfaat langsung dari peluang untuk terlibat dalam pemikiran komputasional. Anak-anak dapat memulai dengan memecahkan teka-teki di atas kertas, melanjutkan di papan permainan, dan pada akhirnya menyelesaikannya. solusi mereka pada komputer. Melalui proses ini, pelajar dapat dibimbing melalui aktivitas berpikir algoritmik yang semakin kompleks yang dibangun dari pengetahuan

diam-diam dan kegembiraan mereka tentang permainan. Studi khusus ini menyimpulkan bahwa dengan melakukan aktivitas berpikir komputasi, anak-anak memperoleh pemahaman yang lebih baik tentang pola berpikir algoritmik. Pemahaman yang lebih baik ini dapat diterjemahkan dengan baik ke dalam pembelajaran matematika dan meningkatkan keterampilan pemecahan masalah anak. Kemajuan ini dapat memberikan dampak positif yang jelas terhadap kinerja mereka sebagai pelajar – dan, dalam jangka panjang, sebagai profesional[11].

Pengkodean Mendorong Ketekunan. Ilmu komputer adalah salah satu dari sedikit disiplin profesional yang hampir selalu mengalami kegagalan. Kegagalan tidak hanya dapat dikenali dengan segera - misalnya, sebuah program "rusak", dan tidak berfungsi sebagaimana mestinya - namun kesuksesan secara harafiah tidak dapat dicapai sampai semua kesalahan ditangani. Bahkan program yang paling sederhana pun memerlukan pembuat kode untuk memahami serangkaian masalah dan menyelesaikannya; jika tidak, kode tidak akan berjalan sebagaimana mestinya. Semua pengembang profesional telah menghabiskan waktu hingga larut malam untuk menyisir Stack Overflow, berjuang dengan sia-sia untuk mengidentifikasi cara paling efisien untuk menyatukan bagian-bagian aplikasi menjadi satu kesatuan yang kohesif. Tanyakan kepada pengembang mana pun apa itu “dereferensi penunjuk nol” — mereka akan bergidik karena frustrasi, mengingat berjam-jam yang dihabiskan untuk mencoba melacak bug yang sulit dipahami[12].

Namun dengan semua rasa frustrasi dan kegagalan ini, muncullah manfaat karakter yang nyata dan bertahan seumur hidup: ketekunan. “Data kami menunjukkan bahwa banyak individu berbakat tidak menepati komitmen mereka,” psikolog Angela Lee Duckworth pernah menyatakan dalam sebuah pembicaraan TED tentang nilai kegigihan. “Faktanya, dalam data kami, ketabahan biasanya tidak berhubungan atau bahkan berbanding terbalik dengan ukuran bakat.” Ketabahan, sebagaimana didefinisikan oleh Duckworth dan rekan-rekan psikolognya, mengacu pada kemampuan langka untuk bertahan melewati kegagalan yang berulang. Orang-orang yang paling sukses dalam hidup bukanlah mereka yang menghindari dari kegagalan, melainkan mereka yang memandang kegagalan sebagai sinyal yang bermanfaat. Ketika anak-anak mulai membuat program, mereka belajar bahwa kegagalan bersifat sementara, dan tidak harus membuat mereka frustrasi atau menghambat kemajuan. Bahkan keberhasilan kecil pun dapat memberikan dorongan yang dibutuhkan anak-anak untuk mengatasi masalah dalam program mereka. Seiring berjalannya waktu, ketekunan ini dapat meningkatkan ketabahan anak – dan menjadi salah satu indikator terpenting keberhasilan pendidikan dan karier mereka di masa depan[13].

Pengkodean dapat membantu anak Anda mengembangkan pola pikir berkembang : sebuah filosofi dimana kemampuan seseorang untuk belajar dapat ditingkatkan, disesuaikan sesuai kebutuhan, dan berdiri teguh melawan kegagalan yang melemahkan semangat. Mentalitas apa yang lebih baik yang bisa diterapkan seorang anak sejak dini?[14],[15]

Coding Mengajarkan Kreativitas. Kreativitas, kreativitas, kreativitas; ini adalah sesuatu yang sangat diperhatikan oleh setiap pendidik awal dan orang tua, namun kualitas ini tidak dihargai oleh sebagian besar orang dewasa yang bekerja. Penelitian yang dilakukan oleh tim psikolog UC Berkeley menunjukkan bahwa kreativitas cenderung menurun seiring bertambahnya usia. Pepatah tersebut benar adanya – orang dewasa cenderung kurang kreatif dibandingkan anak-anak. Pada saat kita mencapai usia senja, kita telah “melupakan” salah satu keterampilan terpenting dalam hidup. “Orang-orang kreatif memiliki respons yang cepat dan efektif yang membantu mereka mencapai tujuan hidup dan memungkinkan mereka menikmati perjalanan... [Ini] adalah keahlian dan struktur kepribadian unik dan individual yang dikembangkan sepanjang masa kanak-kanak dan disesuaikan pada masa remaja, ” Peneliti Notre Dame, Doireann O'Connor pernah mencatat[16].

Lalu, bagaimana cara kita menjaga kemampuan kreatif anak seiring bertambahnya usia? Para ahli menyarankan agar anak-anak belajar coding. Sebagai praktiknya, pengkodean memosisikan pengembang sebagai “pencipta” atau “pembangun dunia” dalam lingkungan pemrograman — menurut definisi, disiplin ini memerlukan pemikiran kreatif dan pola pikir eksperimental. Dengan pemrograman, anak-anak terus-menerus didorong untuk bereksperimen. Begitu mereka memahami fungsi dasar, mereka dapat terus bertanya pada diri sendiri, Bagaimana jika saya mencobanya? Apakah itu akan berhasil? Komputer memberikan umpan balik langsung, yang selanjutnya mendorong para pembuat kode muda untuk menemukan solusi yang bisa diterapkan terhadap tantangan apa pun yang mereka hadapi. Proyek pengkodean yang sesuai dengan usia mendorong anak-anak untuk bercerita, menghidupkan karakter, dan mengembangkan program unik, yang memberdayakan mereka untuk menggabungkan pemikiran logis dan sistematis dengan pembelajaran yang kreatif, artistik, dan intuitif[17].

Coding Membantu Mengajarkan Literasi Digital. Mengapa anak-anak harus belajar coding? Teknologi ada di sekitar kita; hal ini mendukung pekerjaan kita, memfasilitasi komunikasi, memungkinkan konektivitas sosial, dan mendukung hampir semua industri dalam perekonomian. Di dunia modern, kemampuan untuk memahami dan berinteraksi dengan teknologi bukan hanya hal yang baik — namun juga merupakan suatu kebutuhan dalam menghadapi masyarakat yang semakin terdigitalisasi. Anak-anak perlu mengetahui teknologi; dengan kata lain, mereka harus melek digital. Literasi digital merupakan istilah yang luas. Secara umum, ini mengacu pada kemampuan seseorang untuk memahami dan berinteraksi dengan teknologi dengan cara yang bijaksana dan bermakna. Topik-topik seperti keamanan internet, cyberbullying, jejak digital, dan etika online semuanya termasuk dalam bidang literasi digital. Misalnya, anak-anak yang melek digital dapat meneliti topik secara online, menemukan informasi yang relevan, dan kemudian berpikir kritis tentang sumber yang mereka identifikasi. Dari sana, mereka dapat merumuskan pendapat mereka sendiri dan, jika diperlukan, memberikan tanggapan. Mereka juga dapat mengelola kehadiran digital mereka dan berkomunikasi dengan orang lain melalui saluran online[18].

Perlu dicatat bahwa banyak anak – dan khususnya remaja – sudah lebih mahir dalam kehidupan digital dan lebih siap untuk menangani kelebihan informasi yang timbul dari interkoneksi online yang konstan dibandingkan generasi sebelumnya. Meskipun coding belum menjadi kebutuhan sehari-hari bagi pengguna internet, coding mungkin akan menjadi kebutuhan sehari-hari dalam beberapa tahun ke depan. Semua komunikasi online sudah difasilitasi melalui program berkode; beberapa game, seperti Minecraft, memungkinkan pengguna membuat kode fitur mereka sendiri. Tampaknya masuk akal untuk berpikir bahwa pemrograman akan menjadi sebuah literasi yang diperlukan, seperti halnya menavigasi platform media sosial atau penelitian internet[19].

Meminjam kutipan dari Annette Lee , penulis Coding Literacy, “Jika Anda tidak tahu cara memprogram, Anda dapat menjalani hidup dengan baik. Namun hal ini akan segera berubah.” Tampaknya yang terbaik bagi orang tua untuk mendorong anak-anak mereka menjadi yang terdepan dengan belajar coding sejak dini[20].

Coding adalah Keterampilan Membangun Karir. Pentingnya pemrograman tidak bisa dianggap remeh dalam perekonomian kita, yang kini sebagian besar didorong oleh teknologi. Biro Statistik Tenaga Kerja melaporkan bahwa lowongan pekerjaan untuk pengembang perangkat lunak diperkirakan akan tumbuh sebesar 21 persen antara tahun 2018 dan 2028 – yang berarti empat kali lipat angka rata-rata untuk semua karier. Selain itu, tidak semua perusahaan mengharuskan pengembangnya memperoleh gelar sarjana empat tahun ; banyak yang kekurangan bakat sehingga mereka menerima siapa pun yang memiliki keterampilan dan potensi kerja yang diperlukan.

Namun inilah alasan mengapa anak-anak harus membuat kode meskipun mereka tidak tertarik untuk menjadi pengembang penuh waktu: mereka masih dapat memperoleh manfaat secara

profesional dengan mempelajari satu atau dua bahasa pengkodean. Keahlian pemrograman sangat berharga, terlepas dari latar belakang pendidikan atau minat pekerjaan anak Anda. Dalam beberapa kasus, bahkan mungkin kemampuanlah yang pada akhirnya mengangkat mereka melampaui pesaing mereka di pasar kerja. Kendala terbesarnya hanyalah menguasai bahasa pemrograman pertama tersebut — namun begitu mereka menguasainya, mereka akan memiliki posisi yang baik untuk mempelajari berbagai bahasa, kerangka kerja, dan teknologi di kemudian hari.

Coding Meningkatkan Kepercayaan Diri dan Keterampilan Komunikasi. Saatnya untuk menghilangkan prasangka kesalahpahaman yang populer; coding bukanlah aktivitas tunggal. Kecuali pekerja lepas dan operasi satu orang, sebagian besar pemrogram perangkat lunak bekerja dalam tim dan secara kolektif berupaya mencapai tujuan pengembangan. Sebuah studi yang dipublikasikan di *Teknologi Informasi dan Perangkat Lunak* menemukan bahwa kinerja pengembang sangat terkait erat dengan dinamika tim sehingga untuk meningkatkan kinerja tim, bisnis perlu mengatasi aspek lainnya. Seperti yang ditulis para peneliti: “Meningkatkan pengalaman kinerja memerlukan integrasi faktor-faktor lunak, seperti komunikasi, semangat tim, identitas tim, dan nilai-nilai, ke dalam proses pengembangan secara keseluruhan.”

Tentu; pengkodean tidak harus menjadi upaya kelompok, namun memberikan konteks yang sangat baik untuk kolaborasi. Mendaftarkan anak Anda ke kelas pemrograman bisa menjadi cara yang bagus untuk mengembangkan keterampilan membangun tim dan komunikasi — dan memiliki keterampilan komunikasi yang baik akan memberdayakan anak-anak untuk menavigasi lingkungan sosial, berkolaborasi dengan orang lain, berprestasi secara akademis, dan mencapai tujuan mereka. Yang terpenting, anak-anak belajar mengembangkan kepercayaan diri dan citra diri yang sehat. Proyek pengkodean mengajarkan anak Anda bahwa masalah apa pun dapat dipecahkan, bahwa hambatan apa pun dapat diatasi dengan upaya kreatif dan kerja sama tim yang berulang-ulang. Mungkinkah ada alasan yang lebih baik mengapa anak-anak harus belajar coding? Mungkin tidak.

Pemikiran Terakhir: Mengapa Anak-Anak Harus Membuat Kode. Dapat dikatakan bahwa coding adalah salah satu hobi paling konstruktif untuk diterapkan oleh anak Anda, terutama karena coding dapat membantu mereka mengembangkan ketekunan, komunikasi, kreativitas, dan pemikiran tingkat tinggi. Ya, belajar coding juga mempersiapkan anak Anda untuk mendapatkan peluang karier yang berpotensi menguntungkan, namun hal ini menawarkan lebih dari itu. Orang tua modern tidak bisa salah dengan mendorong anak mereka untuk membuat program. Mulailah dengan beberapa sumber daya di atas dan perkenalkan anak Anda pada program yang sesuai dengan perkembangannya hari ini.

4. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil setelah melakukan pengabdian masyarakat ini, antara lain:

1. Programming menjadi kebutuhan teknis utama yang harus dimiliki oleh anak-anak sejak dini, agar dapat mampu menghadapi perkembangan teknologi informasi yang sedemikian cepatnya, sehingga hal ini harus segera di eksekusi agar anak-anak siap dalam menghadapi kebutuhan industri
2. Programming merupakan sebuah kemampuan yang dibutuhkan karena sampai hari ini, banyak programmer yang dicari untuk dapat menciptakan dan melakukan inovasi untuk dapat memenuhi semua aktivitas
3. Programming dibutuhkan untuk dapat mengembangkan dan menciptakan aplikasi-aplikasi yang inovatif untuk dapat bermanfaat dan berguna untuk masyarakat

DAFTAR PUSTAKA

- [1] A. K. Lew, M. Huot, S. Staton, and V. K. Mansinghka, “ADEV: Sound Automatic Differentiation of Expected Values of Probabilistic Programs,” *Proc. ACM Program. Lang.*, vol. 7, no. POPL, p. 5:121-5:153, Jan. 2023, doi: 10.1145/3571198.
- [2] L. Dive, I. Holmes, and A. J. Newson, “Is It Just for a Screening Program to Give People All the Information They Want?,” *Am. J. Bioeth.*, vol. 23, no. 7, pp. 34–42, Jul. 2023, doi: 10.1080/15265161.2023.2207510.
- [3] M. V. Hermenegildo, J. F. Morales, and P. Lopez-Garcia, “Some Thoughts on How to Teach Prolog,” in *Prolog: The Next 50 Years*, D. S. Warren, V. Dahl, T. Eiter, M. V. Hermenegildo, R. Kowalski, and F. Rossi, Eds., in *Lecture Notes in Computer Science*. Cham: Springer Nature Switzerland, 2023, pp. 107–123. doi: 10.1007/978-3-031-35254-6_9.
- [4] A. Fraser, L. Knoll, and D. Hevenstone, “Contested Social Impact Bonds: welfare conventions, conflicts and compromises in five European Active-Labor Market Programs,” *Int. Public Manag. J.*, vol. 26, no. 3, pp. 339–356, May 2023, doi: 10.1080/10967494.2022.2089792.
- [5] M. B. Carstensen, E. Sørensen, and J. Torfing, “Why we need bricoleurs to foster robust governance solutions in turbulent times,” *Public Adm.*, vol. 101, no. 1, pp. 36–52, 2023, doi: 10.1111/padm.12857.
- [6] E. H. Hur, K. Ardeleanu, T. W. Satchell, and L. Jeon, “Why are they leaving? Understanding Associations between early childhood program policies and teacher turnover rates,” *Child Youth Care Forum*, vol. 52, no. 2, pp. 417–440, Apr. 2023, doi: 10.1007/s10566-022-09693-x.
- [7] “[2303.17125] Understanding the Usability of AI Programming Assistants.” <https://arxiv.org/abs/2303.17125> (accessed Sep. 04, 2023).
- [8] “Using Large Language Models to Enhance Programming Error Messages | Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1.” <https://dl.acm.org/doi/abs/10.1145/3545945.3569770> (accessed Sep. 04, 2023).
- [9] S. S. Lu, R. Tian, and D. K. W. Chiu, “Why do people not attend public library programs in the current digital age? A mix method study in Hong Kong,” *Libr. Hi Tech*, vol. ahead-of-print, no. ahead-of-print, Jan. 2023, doi: 10.1108/LHT-04-2022-0217.
- [10] E. Matthes, *Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming*. No Starch Press, 2023.
- [11] M. Law, “Conflict-Driven Inductive Logic Programming,” *Theory Pract. Log. Program.*, vol. 23, no. 2, pp. 387–414, Mar. 2023, doi: 10.1017/S1471068422000011.
- [12] S. M. Beaudrie and S. Loza, “Insights into SHL program direction: student and program advocacy challenges in the face of ideological inequity,” *Lang. Aware.*, vol. 32, no. 1, pp. 39–57, Jan. 2023, doi: 10.1080/09658416.2021.1949333.
- [13] J. P. Comings, “Persistence: Helping Adult Education Students Reach Their Goals,” in *Review of Adult Learning and Literacy, Volume 7*, Routledge, 2007.
- [14] J. Patterson-Price and A. Pass, “‘Not for Serious Purpose’: Discrepancy between Parent and Child Motivation for Participation in a Community Dance Program,” *J. Dance Educ.*, vol. 23, no. 2, pp. 114–122, Apr. 2023, doi: 10.1080/15290824.2021.1908547.
- [15] G. De Toni, B. Lepri, and A. Passerini, “Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis,” *Mach. Learn.*, vol. 112, no. 4, pp. 1389–1409, Apr. 2023, doi: 10.1007/s10994-022-06293-7.
- [16] I. Štempeľová and O. Takáč, “IMPLEMENTATION OF ELEMENTS OF INTELLIGENT AGRICULTURE IN INFORMATICS LESSONS WITH REGARDS TO THE SUPPORT OF INTERSUBJECT RELATIONSHIPS,” *INTED2023 Proc.*, pp. 5968–5973, 2023, doi: 10.21125/inted.2023.1571.
- [17] A. R. Bazzi, L. C. Shaw, K. B. Biello, S. Vahey, and J. K. Brody, “Patient and Provider Perspectives on a Novel, Low-Threshold HIV PrEP Program for People Who Inject Drugs

- Experiencing Homelessness,” *J. Gen. Intern. Med.*, vol. 38, no. 4, pp. 913–921, Mar. 2023, doi: 10.1007/s11606-022-07672-5.
- [18] T. Dingsøy, F. O. Bjørnson, J. Schrof, and T. Sporse, “A longitudinal explanatory case study of coordination in a very large development programme: the impact of transitioning from a first- to a second-generation large-scale agile development method,” *Empir. Softw. Eng.*, vol. 28, no. 1, p. 1, Nov. 2022, doi: 10.1007/s10664-022-10230-6.
- [19] F. Kereki, *Mastering JavaScript Functional Programming: Write clean, robust, and maintainable web and server code using functional JavaScript and TypeScript*. Packt Publishing Ltd, 2023.
- [20] M. J. B. Ludvik, *Outcomes-Based Program Review: Closing Achievement Gaps In- and Outside the Classroom With Alignment to Predictive Analytics and Performance Metrics*. Taylor & Francis, 2023.