

Hubungan antara Kemampuan Matematika dengan Keterampilan Pemrograman

Zulkipli¹

¹*Teknik Informatika, Sekolah Tinggi Teknologi Indonesia Tanjungpinang
Jalan Pompa Air No. 28 km 2,5 Tanjungpinang, Indonesia*

¹zulkipli@sttindonesia.ac.id

Abstrak— Perkembangan komputer berlangsung hingga lima generasi, mulai tahun 1940 hingga sekarang. Seiring dengan perkembangan teknologi yang pesat, maka inovasi perangkat lunak pada bidang tertentu harus dilakukan untuk memudahkan pemrosesan data. Ada suatu fenomena dimana beberapa Mahasiswa Informatika mempertanyakan “Mengapa mereka sebagai calon programmer masih harus mempelajari Mata Kuliah Matematika”. Padahal, tanpa mereka sadari salah satu yang mereka butuhkan untuk menjadi programmer yang baik adalah mampu berfikir logis, dengan mempelajari bagaimana cara berfikir matematika. Permasalahan yang telah dijelaskan tersebut memunculkan gagasan bagi peneliti untuk melihat bagaimana hubungan (korelasi) antara Kemampuan Matematika dengan Keterampilan Pemrograman. Penelitian ini adalah jenis penelitian kuantitatif dengan pendekatan deskriptif untuk mengetahui seberapa erat hubungan antara Kemampuan Matematika dengan Keterampilan Pemrograman. Sampel yang diambil berjumlah 60 mahasiswa Program Studi Teknik Informatika dan Sistem Informasi dari salah satu perguruan tinggi di Kepulauan Riau yang dipilih menggunakan teknik *Random Sampling*. Data dikumpulkan dengan menggunakan Angket berskala likert, berisi 19 Pernyataan mengenai Kemampuan Matematika dan 10 Pernyataan mengenai Keterampilan Pemrograman. Data yang diperoleh selanjutnya dianalisis menggunakan model analisis *correlate pearson product moment* Berdasarkan hasil penelitian, dinyatakan bahwa ada hubungan yang erat antara Kemampuan Matematika dengan Keterampilan Pemrograman.

Kata kunci— Perkembangan teknologi, kemampuan berfikir logis, kemampuan matematika, keterampilan pemrograman, korelasi.

Abstract— The development of computers lasted for five generations, starting from 1940 until now. Along with rapid technological developments, software innovations in certain fields must be carried out to facilitate data processing. There is a phenomenon where some Informatics Students question "Why do they as prospective programmers still have to study Mathematics Subjects". In fact, without them realizing, one of the things they need to become a good programmer is being able to think logically, by learning how to think mathematically. The problems that have been explained give rise to the idea for researchers to see the relationship (correlation) between Mathematics Ability and Programming Skills. This research is a type of quantitative research with a descriptive approach to find out how close the relationship is between Mathematics Ability and Programming Skills. The sample taken was 60 students from the Informatics Engineering and Information Systems Study Program from one of the universities in the Riau Islands who were selected using the Random Sampling technique. Data was collected using a Likert scale questionnaire, containing 19 statements regarding mathematical abilities and 10 statements regarding programming skills. The data obtained was then analyzed using the Pearson product moment correlation analysis model. Based on the research results, it was stated that there is a close relationship between Mathematics Ability and Programming Skills.

Keywords— Technological developments, logical thinking abilities, mathematical abilities, programming skills, correlation.

I. PENDAHULUAN

Perkembangan komputer berlangsung hingga lima generasi, mulai tahun 1940 hingga sekarang. Awalnya negara-negara yang terlibat dalam Perang Dunia II mengembangkan komputer untuk mengeksplorasi potensi strategis yang ada pada komputer. Hal ini lah yang menjadi pelopor meningkatnya pendanaan untuk mengembangkan komputer serta mengakselerasi kemajuan teknik komputer. Kemudian, komputer generasi berikutnya menggantikan bahasa mesin dengan bahasa rakitan yang diimplementasikan menggunakan singkatan-singkatan menggantikan *binary code*. Masuk dalam generasi ketiga, para ilmuwan berhasil menemukan chip tunggal semikonduktor yang memungkinkan komputer menggunakan *operating system*. Komputer generasi IV mulai didistribusikan secara komersil bersama paket *software* yang mudah

diimplementasikan oleh orang awam. Hingga sampai saat ini komputer generasi V masih terus dikembangkan.

Seiring dengan perkembangan teknologi yang pesat, maka inovasi perangkat lunak pada bidang tertentu harus dilakukan untuk memudahkan pemrosesan data [1][2]. Menyadari akan hal ini, bahasa pemrograman tentu menjadi hal yang harus dikuasai bagi para pengembang perangkat lunak [3][4][5]. Bahasa pemrograman yang mempunyai kosa kata terbatas digunakan untuk membangun komponen perangkat lunak, suatu tata bahasa yang terbatas secara eksplisit. Bahasa tersebut adalah mewakili secara simbolis berdasarkan rangkaian instruksi CPU. Hal yang perlu diingat bahwa pada saat pengembang *software* membangun suatu program yang diproduksi secara baik dan terstruktur maka kecepatan eksekusi program tersebut akan berjalan secara efisien [6][7].

Software pada saat ini sangat dibutuhkan untuk mengakomodasi kebutuhan manusia dalam beberapa hal [8][9][10], contohnya aplikasi pemesanan makanan, transportasi, hotel, tempat wisata, dan masih banyak lagi. Melihat perkembangan komputer dan perangkat lunak sejauh ini, dapat dikatakan bahwa pengembangan perangkat lunak akan berbanding lurus dengan aspek kebutuhan serta perkembangan jaman [11][12][13].

Pembuatan suatu perangkat lunak dibutuhkan analisis sistem yang baik, mampu membaca struktur data, serta *user requirement* [14][15]. Di sisi lain, ada beberapa yang berpendapat bahwa salah satu *soft skill* yang harus dimiliki programmer adalah *logical thinking* yang baik [16][17][18]. Hal ini berbeda dengan *hard skill* yang menuntut seseorang secara teknis harus mampu menulis kode secara prosedural, *logical thinking* dikaitkan dengan kemampuan memvisualisasikan bahasa realita ke dalam bahasa pemrograman. Kemampuan tersebut juga dapat membantu programmer untuk memanipulasi serta mengontrol output yang diinginkan. Kemampuan berfikir logis dapat diasah dengan mempelajari Matematika [19], yang memang menjadi salah satu tolok ukur apakah seseorang mampu berfikir logis atau tidak.

Pemahaman konsep ilmiah memegang peranan penting, salah satunya adalah menghubungkan suatu konsep dengan hal lain yang telah dipelajari. Oleh karena itu, kemampuan memahami konsep merupakan tujuan penting. Beberapa peneliti mengungkapkan bahwa kemampuan berfikir matematis dapat membantu seseorang mengaktualisasikan apa yang difikirkannya menjadi lebih sederhana menggunakan simbol-simbol, variabel, dsb [20][21]. Selain itu, Memahami salah satu materi matematika (logika dasar) dan penerapannya dalam menyusun algoritma akan menjadi modal awal yang cukup untuk mempelajari pemrograman [22][23]. Dengan mempelajari logika pemrograman yang disusun dalam algoritma terstruktur, seseorang dapat memecahkan masalah dalam pemrograman dengan lebih cepat dan efisien.

Proses berpikir matematis banyak memberikan kontribusi terhadap perkembangan berpikir seseorang secara keseluruhan, mengoptimalkan kemampuan seseorang [24]. Terkait dengan itu, seseorang dihadapkan pada berbagai hal yang menunjang kemampuan berpikir tingkat tinggi. Kemampuan berfikir matematis juga dianggap merepresentasikan bahwa seseorang memiliki kemampuan pemecahan masalah yang baik [25]. Kemampuan pemecahan masalah sangat penting bagi seseorang dalam mengembangkan perangkat lunak. Dasar yang harus dimiliki seorang programmer diantaranya keterampilan logika, algoritma, dan kemampuan pemrosesan secara sistematis. Programmer tanpa dibekali kemampuan logika dan algoritma yang baik tentu dapat dinyatakan sebagai

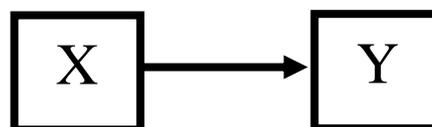
seseorang yang pemahamannya semu (*pseudo understanding*) [26][27].

Berkaitan dengan hal tersebut, Ada suatu fenomena dimana beberapa Mahasiswa Informatika mempertanyakan “Mengapa mereka sebagai calon programmer masih harus mempelajari Mata Kuliah Matematika?”. Padahal, tanpa mereka sadari salah satu yang mereka butuhkan untuk menjadi programmer yang baik adalah mampu berfikir logis [28][29], dengan mempelajari bagaimana cara berfikir matematika. Permasalahan yang telah dijelaskan tersebut memunculkan gagasan bagi peneliti untuk melihat bagaimana hubungan (korelasi) antara Kemampuan Matematika dengan Keterampilan Pemrograman.

II. METODOLOGI PENELITIAN

Penelitian ini adalah jenis penelitian kuantitatif dengan pendekatan deskriptif untuk mengetahui seberapa erat hubungan antara Kemampuan Matematika dengan Keterampilan Pemrograman. Sampel yang diambil berjumlah 60 mahasiswa Program Studi Teknik informatika dan Sistem Informasi dari salah satu perguruan tinggi di Kepulauan Riau yang dipilih menggunakan teknik *Random Sampling*. Data dikumpulkan dengan menggunakan Angket berskala likert, berisi 19 Pernyataan mengenai Kemampuan Matematika dan 10 Pernyataan mengenai Keterampilan Pemrograman. Data yang diperoleh selanjutnya dianalisis menggunakan model analisis *correlate pearson product moment*, sebelumnya data harus teruji memenuhi dua syarat untuk dianalisis, yaitu data harus berdistribusi normal dan ada linearitas antara kedua variabel tersebut.

Dalam penelitian ini terdapat dua variabel yang dianalisis, yaitu Kemampuan Matematika sebagai variabel bebas (*Independent*) dan Keterampilan Pemrograman sebagai variabel terikat (*dependent*). Hubungan antara kedua variabel tersebut divisualisasikan sebagai berikut:



Gambar 1. Konstelasi Masalah Penelitian

Keterangan:

X = Kemampuan Matematika

Y = Keterampilan Pemrograman

Telah dijelaskan sebelumnya bahwa sampel dipilih menggunakan teknik *random sampling*, artinya pengambilan sampel dari suatu populasi (mahasiswa salah satu perguruan tinggi) dilakukan secara acak. Teknik tersebut digunakan untuk memudahkan peneliti untuk melakukan penelitian terhadap sebagian elemen objek penelitian. Teknik analisis

data yang digunakan yaitu analisis korelasi sederhana dengan taraf signifikansi $\alpha = 0,05$.

III. HASIL PENELITIAN

A. Uji Normalitas Kolmogorov-Smirnov

Kenormalan suatu data merupakan salah satu syarat untuk melakukan analisis korelasi. Uji Normalitas Kolmogorov-Smirnov ini dilakukan untuk mengetahui apakah data penelitian berdistribusi normal. Dasar pengambilan keputusan dalam uji ini adalah jika nilai signifikansi (Sig.) $> 0,05$ maka data penelitian berdistribusi normal, apabila nilai Sig. $< 0,05$ maka data penelitian tidak berdistribusi normal. Hasil Uji Normalitas ditunjukkan pada tabel berikut:

TABEL 1
HASIL UJI NORMALITAS ONE K-S TEST

	Unstandardized Residual
N	60
Normal Parameters ^{a,b}	
Mean	0E-7
Std. Deviation	3.01266027
Absolute	.104
Most Extreme Differences	
Positive	.104
Negative	-.073
Kolmogorov-Smirnov Z	.805
Asymp. Sig. (2-tailed)	.536

- a. Test distribution is Normal.
- b. Calculated from data.

Berdasarkan tabel output tersebut, diketahui bahwa nilai signifikansi Asymp. Sig (2-tailed) sebesar 0,536 lebih besar dari 0,05. Maka dapat disimpulkan bahwa data berdistribusi normal.

B. Uji Linearitas

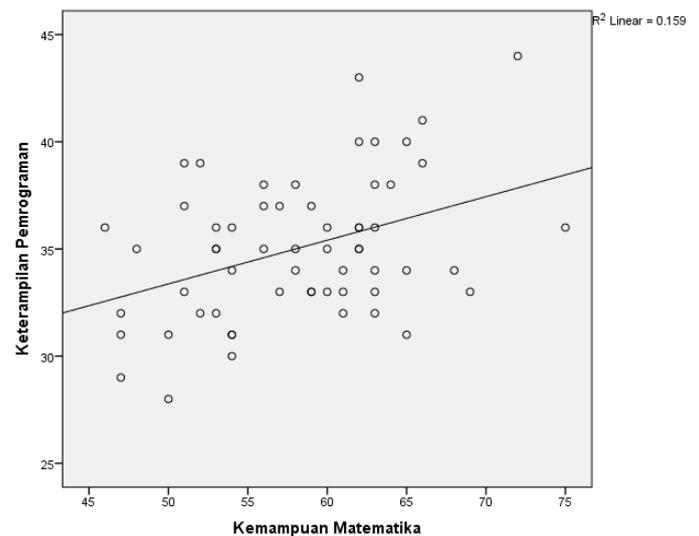
Ada beberapa dasar pengambilan keputusan dalam uji linearitas. Pertama, dengan membandingkan nilai signifikansi. Jika nilai Signifikansi Linearity Sig. $< 0,05$, maka ada hubungan yang linear secara signifikan antara variabel Kemampuan Matematika dengan variabel Keterampilan Pemrograman. Jika nilai Signifikansi Linearity Sig. $> 0,05$, maka tidak ada hubungan yang linear secara signifikan antara variabel Kemampuan Matematika dengan variabel Keterampilan Pemrograman. Dasar pengambilan keputusan kedua yaitu dengan membandingkan nilai Deviasion from Linearity untuk menunjukkan simpangan dari kelinierannya. Jika nilai Deviation from Linearity Sig. $> 0,05$, maka Simpangannya tidak signifikan (linear). Jika nilai Deviation of Linearity Sig. $< 0,05$, maka Simpangannya signifikan.

Kombinasi dari kedua dasar tersebut digunakan untuk menjustifikasi apakah antara data tersebut linear. Signifikansi Linearity nya signifikan dan Deviation from Linearity nya tidak signifikan, maka dapat dikatakan bahwa tersebut benar-benar Linear (Linear Kuat). Kemudian, Signifikansi Linearity nya signifikan dan Deviation from Linearity nya signifikan, maka dapat dikatakan data tersebut Linear Kuat tetapi kemungkinan data tersebut mengembung. Jika Signifikansi Linearity nya tidak signifikan (lemah) dan Deviation from Linearity nya signifikan, maka dapat dikatakan Data tersebut tidak Linear. Linearitas data ditunjukkan berdasarkan tabel berikut:

TABEL II
HASIL UJI LINEARITAS

	Sum of Squares	df	Mean Square	F	Sig.
(Combined)	366.650	22	16.666	2.282	.013
Keterampilan Pemrograman	101.359	1	101.359	13.880	.001
Deviation from Linearity	265.291	21	12.633	1.730	.071
* Kemampuan Matematika	270.200	37	7.303		
Total	636.850	59			

Berdasarkan olah data, Signifikansi Linearity $0,001 < 0,05$ (Linear) dan Deviation from Linearity $0,071 > 0,05$ (simpangan tidak signifikan). Maka, data tersebut dapat dijustifikasi Linear Kuat. Sifat kelinearan data tersebut diperkuat dengan melihat grafik berikut:



Gambar 2. Grafik plot yang menunjukkan kedua variabel terdapat hubungan yang linear

C. Hasil Uji Korelasi Pearson

Dua syarat untuk melakukan uji korelasi pearson product moment telah terpenuhi, dimana data tersebut dinyatakan berdistribusi normal dan data tersebut terdapat hubungan yang linear. Pengambilan keputusan apakah Kemampuan

Matematika dan Keterampilan Pemrograman terdapat korelasi didasarkan pada perbandingan antara nilai r hitung dengan r tabel. Jika r hitung lebih besar dibandingkan r tabel, maka kedua variabel memiliki hubungan yang erat. Di sisi lain, jika menunjukkan hasil sebaliknya, maka kedua variabel tidak memiliki hubungan. Berdasarkan hasil perhitungan data, informasi yang dapat dilihat adalah sebagai berikut:

TABEL III
HASIL UJI KORELASI PEARSON PRODUCT MOMENT

		Kemampuan Matematika	Keterampilan Pemrograman
Kemampuan Matematika	Pearson Correlation	1	.399**
	Sig. (2-tailed)		.002
	N	60	60
Keterampilan Pemrograman	Pearson Correlation	.399**	1
	Sig. (2-tailed)	.002	
	N	60	60

** . Correlation is significant at the 0.01 level (2-tailed).

Diketahui nilai r hitung untuk korelasi antara Kemampuan Matematika (X) dengan Keterampilan Pemrograman (Y) adalah $0,399 > r$ tabel $0,254$, maka dapat dijustifikasi bahwa ada hubungan yang erat antara Kemampuan Matematika dengan Keterampilan Pemrograman. Kemudian, karena nilai Pearson Correlations bernilai positif maka hubungan antara kedua variabel tersebut bersifat positif.

IV. PEMBAHASAN

Hasil analisis data yang telah dilakukan, menyatakan bahwa nilai signifikansi Asymp. Sig (2-tailed) sebesar $0,536$ lebih besar dari $0,05$. Maka dapat disimpulkan bahwa data berdistribusi normal. Kemudian, Signifikansi Linearity $0,001 < 0,05$ (Linear) dan Deviation from Linearity $0,071 > 0,05$ (simpangan tidak signifikan). Maka, data tersebut dapat dijustifikasi Linear Kuat. Hasil akhir penelitian menyatakan bahwa nilai r hitung untuk korelasi antara Kemampuan Matematika (X) dengan Keterampilan Pemrograman (Y) adalah $0,399 > r$ tabel $0,254$, maka dapat dijustifikasi bahwa ada hubungan yang erat antara Kemampuan Matematika dengan Keterampilan Pemrograman.

Variabel bebas dalam penelitian ini adalah Kemampuan Matematika, sedangkan variabel terikatnya adalah Keterampilan Pemrograman dari mahasiswa. Hasil penelitian menunjukkan kemampuan Matematika mahasiswa memiliki hubungan yang erat dengan kemampuan pemrogramannya. Secara keseluruhan dari temuan di atas dapat diinterpretasikan bahwa pemahaman konsep matematika adalah kemampuan pemahaman seseorang dalam memecahkan masalah. Dengan cara ini seseorang dapat mengatasi aspek-aspek matematika dan menemukan kegunaannya dalam bidang matematika itu

sendiri, khususnya yang berkaitan erat dengan bidang teknologi informasi. Dengan memahami konsep matematika secara utuh, mahasiswa dapat memahami kegunaan matematika dalam bidang Sistem Informasi maupun Teknik Informatika. Oleh karena itu dapat dikatakan bahwa pemahaman konsep matematika memberikan dampak positif terhadap bidang teknik informatika.

Pernyataan tersebut bukan tanpa alasan, hal itu dapat terjadi dikarenakan beberapa hal. Sebagai contoh, kalkulus merupakan salah satu ilmu matematika yang diyakini berkaitan dengan informatika. Kalkulus bermanfaat untuk melatih logika dalam berpikir, dikarenakan setiap permasalahan yang ditemui di kehidupan sehari-hari khususnya dalam bidang informatika sebenarnya bisa diselesaikan dengan mengimplementasikan logika penyederhanaan. Melihat perkembangan *Artificial Intelligence*, konsep limit dalam kalkulus diimplementasikan untuk mengembangkan *Artificial Intelligence* (kecerdasan buatan). Pemanfaatan Konsep Limit dalam bidang Teknik Informatika untuk menciptakan *Artificial Intelligence* (kecerdasan buatan) adalah suatu kecerdasan buatan dimana *Artificial Intelligence* merupakan salah satu cabang teknik informatika yang dalam menyajikan pengetahuannya menggunakan simbol-simbol dalam mengolah informasi berdasarkan metode heuristik.

Ada beberapa yang menyebutkan bahwa beberapa bidang matematika yang memiliki hubungan erat dengan informatika diantaranya: 1. Matematika Diskrit, Ini adalah bidang matematika di mana elemen-elemen yang tidak berhubungan satu sama lain dibahas. Elemen-elemen tersebut dikatakan diskrit apabila terdiri dari beberapa unsur yang berbeda dan tidak berhubungan. Antonim dari diskrit adalah kontinu, yaitu kontinu satu sama lain; 2. Logika dasar, Dalam aplikasi informatika, logika sangat penting untuk menciptakan suatu sistem yang dapat berjalan dengan baik. Saat Anda mempelajari informatika, salah satu hal pertama yang Anda lakukan adalah mempelajari logika dasar; 3. Kalkulus, Kalkulus adalah studi tentang perubahan. Hal ini tentunya sangat berguna untuk komputasi yang penuh variasi dan fleksibilitas. Misalnya ketika kalkulus diterapkan pada pemrosesan citra digital, seperti deteksi tepi dan segmentasi; 4. Algoritma, adalah serangkaian instruksi untuk memecahkan suatu masalah. Selain itu, algoritma digunakan untuk pemrosesan data. Sejauh ini lebih umum diartikan sebagai pola yang dihasilkan dari tampilan beberapa data; 5. Statistika, Seperti yang telah disebutkan sebelumnya bahwa informatika erat kaitannya dengan data, sehingga statistika tentu saja merupakan ilmu yang penting untuk dipelajari. Statistika adalah ilmu tentang segala sesuatu tentang data, mulai dari pengumpulan, pengolahan, deskripsi, dan penyajiannya.

Informatika berkorelasi terhadap data, mulai dari proses hingga penyimpanan data. Hal tersebut merupakan alasan

matematika yang berhubungan erat dengan komputasi dibutuhkan dalam dunia informatika. Jika disebutkan, ada beberapa bidang matematika yang konsepnya diimplementasikan dalam informatika, diantaranya Matematika Diskrit, yang membahas set dan relasi, teori graf, logika, hingga peluang. Kemudian, Logika Dasar, yang sangat penting mengembangkan maupun menciptakan suatu sistem agar berjalan dengan baik dan benar. Kalkulus yang sebelumnya telah disebutkan salah satu manfaatnya. Serta algoritma, yang konsepnya digunakan sebagai serangkaian instruksi untuk memecahkan suatu masalah. Lebih dari itu, algoritma diimplementasikan pada pengolahan data. Sebagai contoh, rekomendasi konten yang muncul pada aplikasi media sosial merupakan algoritma berdasarkan riwayat atau jenis konten yang paling sering dilihat oleh penggunanya.

Matematika peranannya dalam dunia informatika adalah mendorong seseorang untuk memiliki keterampilan berpikir kritis, logis, analitis dan sistematis [30][31]. Dalam perkembangan teknologi informasi, matematika telah memberikan kontribusi tersendiri. Perkembangan yang pesat di bidang teknologi informasi dan komunikasi saat ini didasarkan pada perkembangan matematika di bidang teori bilangan, aljabar, analisis, teori probabilitas dan matematika diskrit. Menguasai dan menciptakan teknologi masa depan memerlukan pemahaman matematika yang kuat sejak dini [32], karena matematika pada dasarnya dirancang untuk menumbuhkan kemampuan berpikir logis, analitis, sistematis, kritis, dan kreatif, serta keterampilan kerjasama dan kemampuan menggunakan matematika untuk pemecahan masalah.

Berbicara perihal apakah kemampuan matematika memang begitu penting untuk merepresentasikan keterampilan pemrograman atau menguasai bahasa pemrograman. Ada yang berpendapat bahwa menguasai ilmu pemrograman memerlukan kemampuan matematika [33][34]. Di sisi lain, ada yang berpendapat bahwa pemrograman identik dengan penggunaan angka, tipe data, variabel, operasi, dsb. Mereka menyatakan bahwa untuk mengolah data, kemampuan minimal yang harus dimiliki adalah memahami konsep himpunan. Ada juga yang menambahkan bahwa kemampuan matematika akan memudahkan aktualisasi pemikirannya ke dalam pengembangan perangkat lunak [35].

V. KESIMPULAN

Berdasarkan hasil penelitian, dapat dinyatakan bahwa kemampuan matematika berhubungan erat dengan keterampilan pemrograman. Tentu saja pernyataan ini akan memunculkan pro dan kontra. Ada yang berpendapat keterampilan pemrograman tidak terlalu membutuhkan kemampuan matematika, dikarenakan penguasaan dalam pemrograman hanya sebatas penguasaan bahasa program serta

prosedur yang disesuaikan dengan fokus peruntukannya. Pendapat lain menyatakan bahwa kemampuan matematika penting dikarenakan konsep-konsep dalam matematika dapat membantu pengguna memahami pemrograman secara utuh, tidak hanya prosedural, melainkan pemahaman tentang pemrograman yang luas.

REFERENSI

- [1] Y. K. Dwivedi *et al.*, "Opinion Paper: 'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *Int. J. Inf. Manage.*, vol. 71, p. 102642, Aug. 2023, doi: 10.1016/j.ijinfomgt.2023.102642.
- [2] S. Kraus, P. Jones, N. Kailer, A. Weinmann, N. Chaparro-Banegas, and N. Roig-Tierno, "Digital Transformation: An Overview of the Current State of the Art of Research," *SAGE Open*, vol. 11, no. 3, p. 215824402110475, Jul. 2021, doi: 10.1177/21582440211047576.
- [3] A. Hindle, E. T. Barr, M. Gabel, Z. Su, and P. Devanbu, "On the naturalness of software," *Commun. ACM*, vol. 59, no. 5, pp. 122–131, Apr. 2016, doi: 10.1145/2902362.
- [4] R. Duke, E. Salzman, J. Burmeister, J. Poon, and L. Murray, "Teaching programming to beginners - choosing the language is just the first step," in *Proceedings of the Australasian conference on Computing education*, Dec. 2000, pp. 79–86. doi: 10.1145/359369.359381.
- [5] J. Erickson, K. Lyytinen, and K. Siau, "Agile Modeling, Agile Software Development, and Extreme Programming," *J. Database Manag.*, vol. 16, no. 4, pp. 88–100, Oct. 2005, doi: 10.4018/jdm.2005100105.
- [6] D. E. Knuth, "Structured Programming with go to Statements," *ACM Comput. Surv.*, vol. 6, no. 4, pp. 261–301, Dec. 1974, doi: 10.1145/356635.356640.
- [7] Boehm, "Improving Software Productivity," *Computer (Long. Beach. Calif.)*, vol. 20, no. 9, pp. 43–57, Sep. 1987, doi: 10.1109/MC.1987.1663694.
- [8] A. Nurseptaji, "IMPLEMENTASI METODE WATERFALL PADA PERANCANGAN SISTEM INFORMASI PERPUSTAKAAN," *J. Dialekt. Inform.*, vol. 1, no. 2, pp. 49–57, May 2021, doi: 10.24176/detika.v1i2.6101.
- [9] F. D. Macías-Escrivá, R. Haber, R. del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7267–7279, Dec. 2013, doi: 10.1016/j.eswa.2013.07.033.
- [10] Braa, Hanseth, Heywood, Mohammed, and Shaw, "Developing Health Information Systems in Developing Countries: The Flexible Standards Strategy," *MIS Q.*, vol. 31, no. 2, p. 381, 2007, doi: 10.2307/25148796.
- [11] H. Unphon and Y. Dittrich, "Software architecture awareness in long-term software product evolution," *J. Syst. Softw.*, vol. 83, no. 11, pp. 2211–2226, Nov. 2010, doi: 10.1016/j.jss.2010.06.043.
- [12] J. Bolliger and T. Gross, "A framework based approach to the development of network aware applications," *IEEE Trans. Softw. Eng.*, vol. 24, no. 5, pp. 376–390, May 1998, doi: 10.1109/32.685260.
- [13] C. Lüthje, "Characteristics of innovating users in a consumer goods field," *Technovation*, vol. 24, no. 9, pp. 683–695, Sep. 2004, doi: 10.1016/S0166-4972(02)00150-5.
- [14] A. Voutama and E. Novalia, "Perancangan Aplikasi M-Magazine Berbasis Android Sebagai Sarana Mading

- Sekolah Menengah Atas,” *J. Tekno Kompak*, vol. 15, no. 1, p. 104, Feb. 2021, doi: 10.33365/jtk.v15i1.920.
- [15] R. D. Irawan, M. Adha, M. P. Sadana, Z. D. Kusnaa Washilatul Arba’ah, and E. Utami, “PEMODELAN HASIL REKAYASA KEBUTUHAN PERANGKAT LUNAK SISTEM JURNAL ELEKTRONIK TERINTEGRASI ‘IDEOGRAM,’” *JIKO (Jurnal Inform. dan Komputer)*, vol. 7, no. 1, p. 13, Feb. 2023, doi: 10.26798/jiko.v7i1.653.
- [16] L. F. Capretz and F. Ahmed, “Making Sense of Software Development and Personality Types,” *IT Prof.*, vol. 12, no. 1, pp. 6–13, Jan. 2010, doi: 10.1109/MITP.2010.33.
- [17] I. Garcia, C. Pacheco, F. Méndez, and J. A. Calvo-Manzano, “The effects of game-based learning in the acquisition of ‘soft skills’ on undergraduate software engineering courses: A systematic literature review,” *Comput. Appl. Eng. Educ.*, vol. 28, no. 5, pp. 1327–1354, Sep. 2020, doi: 10.1002/cae.22304.
- [18] E. Aivaloglou and F. Hermans, “How Kids Code and How We Know,” in *Proceedings of the 2016 ACM Conference on International Computing Education Research*, Aug. 2016, pp. 53–61. doi: 10.1145/2960310.2960325.
- [19] J. S. Ab, G. Margono, and W. Rahayu, “The Logical Thinking Ability: Mathematical Disposition and Self-Regulated Learning,” *J. Phys. Conf. Ser.*, vol. 1155, p. 012092, Feb. 2019, doi: 10.1088/1742-6596/1155/1/012092.
- [20] T. Dreyfus, “Advanced Mathematical Thinking Processes,” in *Advanced Mathematical Thinking*, Dordrecht: Springer Netherlands, 2002, pp. 25–41. doi: 10.1007/0-306-47203-1_2.
- [21] D. Tall *et al.*, “Symbols and the bifurcation between procedural and conceptual thinking,” *Can. J. Sci. Math. Technol. Educ.*, vol. 1, no. 1, pp. 81–104, Jan. 2001, doi: 10.1080/14926150109556452.
- [22] A. Chasannudin, L. Nuraini, and N. A. Luthfiya, “Pelatihan Aplikasi Scratch Untuk Meningkatkan Kemampuan Computational Thinking Pada Guru,” *Kifah J. Pengabd. Masy.*, vol. 1, no. 2, pp. 153–168, Dec. 2022, doi: 10.35878/kifah.v1i2.502.
- [23] J. Estevez, G. Garate, and M. Grana, “Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch,” *IEEE Access*, vol. 7, pp. 179027–179036, 2019, doi: 10.1109/ACCESS.2019.2956136.
- [24] K. E. Stanovich, “The Comprehensive Assessment of Rational Thinking,” *Educ. Psychol.*, vol. 51, no. 1, pp. 23–34, Jan. 2016, doi: 10.1080/00461520.2015.1125787.
- [25] R. S. Nickerson, “The Teaching of Thinking and Problem Solving,” in *Thinking and Problem Solving*, Elsevier, 1994, pp. 409–449. doi: 10.1016/B978-0-08-057299-4.50019-0.
- [26] A. Robins, J. Rountree, and N. Rountree, “Learning and Teaching Programming: A Review and Discussion,” *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, Jun. 2003, doi: 10.1076/csed.13.2.137.14200.
- [27] J. Duraes, H. Madeira, J. Castelhana, C. Duarte, and M. C. Branco, “WAP: Understanding the Brain at Software Debugging,” in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, Oct. 2016, pp. 87–92. doi: 10.1109/ISSRE.2016.53.
- [28] P. Martin-Löf, “Constructive Mathematics and Computer Programming,” 1982, pp. 153–175. doi: 10.1016/S0049-237X(09)70189-2.
- [29] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl, “GOLOG: A logic programming language for dynamic domains,” *J. Log. Program.*, vol. 31, no. 1–3, pp. 59–83, Apr. 1997, doi: 10.1016/S0743-1066(96)00121-5.
- [30] A. P. Wierzbicki, “Modelling as a way of organising knowledge,” *Eur. J. Oper. Res.*, vol. 176, no. 1, pp. 610–635, Jan. 2007, doi: 10.1016/j.ejor.2005.08.018.
- [31] M. Kordaki and P. Kakavas, “DIGITAL STORYTELLING AS AN EFFECTIVE FRAMEWORK FOR THE DEVELOPMENT OF COMPUTATIONAL THINKING SKILLS,” Mar. 2017, pp. 6325–6335. doi: 10.21125/edulearn.2017.2435.
- [32] E. R. McClure *et al.*, “STEM starts early: Grounding science, technology, engineering, and math education in early childhood,” 2017, [Online]. Available: <http://joanganzcooneycenter.org/publication/stem-starts-early/>
- [33] C. S. Cheah, “Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review,” *Contemp. Educ. Technol.*, vol. 12, no. 2, p. ep272, May 2020, doi: 10.30935/cedtech/8247.
- [34] I. Harel and S. Papert, “Software Design as a Learning Environment,” *Interact. Learn. Environ.*, vol. 1, no. 1, pp. 1–32, Mar. 1990, doi: 10.1080/1049482900010102.
- [35] D. Batory, J. N. Sarvela, and A. Rauschmayer, “Scaling step-wise refinement,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 6, pp. 355–371, Jun. 2004, doi: 10.1109/TSE.2004.23.