

BAB I

PENDAHULUAN

1.1 Latar Belakang

Istilah *computational thinking* yang sebenarnya pertama kali diperkenalkan oleh Seymour Papert pada tahun 1980 kembali dipopulerkan oleh seorang ahli ilmu komputer asal Carnegie Mellon University bernama Jannette M. Wing pada tahun 2006, dimana *computational thinking* diartikan sebagai proses berpikir seperti seorang ilmuwan komputer dalam menghadapi masalah. Lebih lanjut *computational thinking* ini diartikan sebagai proses berpikir yang melibatkan penyelesaian masalah, perancangan sistem, dan pemahaman perilaku manusia, dengan digambarkan ke dalam konsep dasar ilmu komputer (Wing, 2006). Selanjutnya, *Computer Science Teachers Associate* (CSTA) dan *International Society for Technology in Education* (ISTE) mendefinisikan *computational thinking* dalam *Operational Definition of Computational Thinking for K-12 Education* sebagai proses pemecahan masalah yang meliputi abstraksi, dekomposisi, desain algoritma, pengumpulan dan analisis data dengan tugas seperti pengenalan pola, otomatisasi dan generalisasi (Sykora, 2021).

Keterampilan *computational thinking* kini semakin berkembang dan dianggap sebagai salah satu keterampilan esensial dan fundamental yang harus dimiliki oleh setiap pembelajar abad ke-21 termasuk siapapun dan bukan hanya untuk ilmuwan komputer saja. Bahkan Tran (2019) menganggap keterampilan *computational thinking* ini dianggap sangat penting diajarkan sejak usia dini karena sama seperti keterampilan membaca, menulis, dan aritmetika yang harus dimiliki oleh semua anak.

Berdasarkan analisis awal terhadap keterampilan *computational thinking* siswa di lingkungan Kecamatan Tenjo yang dilakukan dengan melakukan tes menggunakan instrumen *Computational Thinking Test (CTt)*, 90% siswa dikategorikan sebagai *Low CT Thinker* atau dengan kata lain mempunyai keterampilan *computational thinking* yang rendah. Dari hasil wawancara terhadap guru dan siswa, hal ini disebabkan karena secara umum siswa belum

pernah mengenal pembelajaran dan arah pembelajaran di kelas yang dapat mengasah keterampilan berpikirnya dalam hal keterampilan *computational thinking*.

Sementara itu sebagai upaya meningkatkan keterampilan *computational thinking*, pengintegrasian keterampilan ini ke dalam berbagai bidang pelajaran dalam pendidikan K-12 sendiri sekarang telah diterima secara luas karena dinilai dapat meningkatkan kualitas pembelajaran (Güven & Gulbahar, 2020). Terlihat bahwa keterampilan *computational thinking* ini telah dimasukkan ke dalam kurikulum nasional di berbagai negara termasuk di Malaysia sejak tahun 2017 (Chongo et al., 2021) dan Thailand pada tahun 2018 (Threekunprapa & Yasri, 2020). Bahkan keterampilan ini juga telah dimasukkan sebagai salah satu aspek penilaian matematika pada kerangka PISA Tahun 2021.

Proses integrasi dan upaya peningkatkan keterampilan *computational thinking* siswa pada pendidikan K-12 di beberapa negara dilakukan melalui berbagai cara, diantaranya dengan pendekatan *plugged-in computing* seperti pembelajaran pemrograman, pendekatan *unplugged* seperti latihan soal pemecahan masalah bernama *Bebras Challenge*, dan kombinasi keduanya. Diantara semua upaya tersebut, pembelajaran pemrograman dianggap sebagai salah satu cara yang paling efektif untuk meningkatkan keterampilan *computational thinking* (Zhang & Nouri, 2019) Selaras dengan pernyataan tersebut, hasil penelitian Wong & Cheung (2020) menunjukkan bahwa pembelajaran pemrograman ini memang telah terbukti efektif dalam meningkatkan keterampilan *computational thinking*. Bahkan beberapa negara termasuk Australia, Inggris, Estonia, Finlandia, Selandia Baru, Norwegia, Swedia, Korea Selatan, Amerika, Macedonia, Yunani dan Prancis meyakini hal tersebut dan mewajibkan pembelajaran pemrograman untuk diterapkan sebagai upaya untuk meningkatkan keterampilan *computational thinking* dimulai dari tingkat pendidikan dasar yang meliputi sekolah dasar dan menengah baik sebagai pembelajaran tersendiri maupun terintegrasi dengan pembelajaran yang lain (Mason & Rich, 2020; Pérez-Marín et al., 2020). Hal tersebut diperkuat dengan hasil penelitian Çiftci & Bildiren (2020) yang

menyebutkan bahwa pembelajaran pemrograman bukan hanya bisa dipelajari oleh profesional, tetapi dapat dipelajari juga oleh siapapun, berapapun usianya.

Penerapan pembelajaran pemrograman di berbagai negara tersebut telah dicoba dengan menggunakan berbagai model pembelajaran, seperti penerapan pembelajaran berbasis permainan (*game-based learning*), robotik, dan pembelajaran berbasis desain. Pembelajaran pemrograman berbasis permainan memang terbukti secara signifikan dapat meningkatkan pemahaman konsep dan keterampilan *computational thinking* siswa (Rose et al., 2018; Weintrop et al., 2016). Selain itu, motivasi dan kegigihan mereka dalam mengikuti kegiatan pembelajaranpun turut meningkat (Israel-Fishelson & HersHKovitz, 2020). Namun secara bersamaan, pembelajaran pemrograman berbasis permainan ini membuat siswa cenderung bekerja sendiri untuk menyelesaikan tugas yang diberikan daripada berkolaborasi dengan temannya (Lin et al., 2020). Besarnya kemungkinan kegagalan fokus siswa pun membuat guru harus memberikan perhatian lebih pada proses penilaian dan pencapaian *computational thinking* siswa (Hooshyar et al., 2021) mengingat pembelajaran berbasis permainan ini akan memiliki hasil yang lebih baik apabila diintegrasikan ketika keterampilan *computational thinking* siswa sudah mencapai tingkat tertentu (Zhao & Shute, 2019).

Penelitian lain mencoba mengintegrasikan pembelajaran pemrograman dengan aktivitas *robotics*. Hal ini juga terbukti dapat meningkatkan motivasi (Chalmers, 2018) dan keterampilan *computational thinking* siswa, bahkan bagi siswa yang tidak memiliki informasi sebelumnya mengenai pemrograman karena aktivitasnya berorientasi pada langkah-langkah pengoperasian robot, meskipun memerlukan desain pembelajaran yang lebih kompleks dan terkadang siswa kesulitan dalam memahami istilah teknis pada rangkaian listrik (Durak et al., 2019). Namun bukan tanpa masalah, penggunaan aktivitas *robotics* menimbulkan konsekuensi pada peran guru yang semakin vital dan membutuhkan pembelajaran yang memiliki soliditas jaringan konsep melalui aktivitas pemecahan masalah (Çakır et al., 2021). Selain itu, tidak tersedianya perangkat *robotics* yang murah dan *open-sources*

(Bakala et al., 2021) menjadi sebuah permasalahan tersendiri dalam penggunaan aktivitas *robotic* pada pembelajaran pemrograman untuk meningkatkan keterampilan *computational thinking* siswa ini.

Dari berbagai model pembelajaran yang diterapkan dalam pembelajaran pemrograman, Model *Design-Based Learning for Computational Thinking (DBLM-CT)* sebenarnya memiliki kelebihan yang dapat mengatasi kelemahan dalam penerapan *game-based learning* dan *robotics*, seperti kecenderungan siswa yang kurang mampu berkolaborasi, adanya tingkat keterampilan prasyarat pada siswa, dan kebutuhan akan jaringan konsep yang solid melalui aktivitas pemecahan masalah. Pembelajaran pemrograman dengan *design-based learning* terbukti dapat mendukung peningkatan keterampilan *computational thinking* (Jun et al., 2017a; Matere, Weng, Astatke, Hsia, Matere, et al., 2021) dan motivasi siswa (Garneli & Chorianopoulos, 2018, 2019; Repenning, 2015). Penerapan *design-based learning* pada pembelajaran pemrograman juga dapat meningkatkan keterampilan pemecahan masalah dan berfikir algoritmik (Saritepeci, 2020a) sekaligus kemampuan siswa berkolaborasi (Jun et al., 2017; Matere et al., 2021).

Namun hasil penerapan model *design-based learning* dalam pembelajaran pemrograman masih menyisakan masalah. Dari tujuh aspek pada keterampilan *computational thinking*, hasil perolehan keterampilan dirasa masih kurang optimal pada beberapa aspek keterampilan seperti pada aspek keterampilan *loops* dan aspek keterampilan *conditionals*. Kegagalan siswa dalam meningkatkan keterampilan pada dua aspek keterampilan *computational thinking* tersebut disebabkan karena siswa tidak mampu untuk menghubungkan dan mengkoordinasikan fakta dan keterampilan yang telah dipelajari sebelumnya (Matere et al., 2021) karena aktivitas *design-based learning* terkadang mengabaikan hubungan antara aktivitas konstruksi desain dengan beberapa keterampilan yang ditargetkan (Li et al., 2021). Terlalu luasnya spektrum pemilihan ide desain yang dapat dipilih siswa menimbulkan kecenderungan keterampilan yang ditargetkan menjadi tidak tercapai.

Hasil penelitian Scherer *et al.* (2020) merekomendasikan agar pembelajaran pemrograman perlu dirancang untuk mengakomodasi gaya belajar dan kemampuan berpikir siswa sesuai dengan usianya. Selain itu Kong *et al.* (2018) lebih dulu menyarankan bahwa pembelajaran pemrograman juga perlu mengarahkan siswa agar menerapkan konsep dan praktis *computational thinking* untuk mengatasi masalahnya sehari-hari. Hal ini tentunya berkaitan dengan perkembangan kognitif dan karakteristik belajar siswa usia belajar yang menurut Piaget (Madjid, 2014: 10) berada pada tahap operasional konkret dimana memiliki kecenderungan belajar sebagai berikut: 1) konkret; 2) integratif/*holistic*; dan 3) hierarkis. Sementara itu penelitian Matere *et al.* (2021) merekomendasikan agar penerapan pembelajaran pemrograman dilakukan melalui aktivitas instruksi yang berulang agar dapat meningkatkan memori siswa dalam mempertahankan konsep yang dipelajari yang mereka butuhkan ketika merancang program.

Berdasarkan kajian di atas, sebagai upaya untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar, pembelajaran pemrograman dengan menerapkan model *design-based learning* dapat menjadi solusi yang tepat meskipun masih terdapat beberapa kekurangan yang membuat peningkatan keterampilan siswa belum optimal. Untuk itu diperlukan sebuah desain pembelajaran pemrograman menggunakan model *design-based learning* yang dikemas dalam tema-tema konkret yang menargetkan keterampilan-keterampilan *computational thinking* secara hierarkis dengan aktivitas desain yang berulang di setiap temanya sebagaimana yang dikembangkan dalam penelitian ini.

1.2 Pembatasan Penelitian

Penelitian ini menghasilkan beberapa produk yang terkait pada pengembangan desain pembelajaran pemrograman dengan model *design-based learning* bagi siswa sekolah dasar. Produk-produk tersebut diantaranya: 1) rancangan pembelajaran pemrograman yang mengacu kepada *Creative Computing Curriculum*; dan 2) bahan ajar cetak berisi materi dan lembar kerja pembelajaran pemrograman berbasis model *design based learning model for computational thinking*.

1.3 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, maka rumusan masalah dalam penelitian ini diantaranya:

- a. Bagaimana mengembangkan desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar?
- b. Bagaimana kelayakan desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar?
- c. Bagaimana menguji efektivitas desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah di paparkan di atas, maka penelitian ini bertujuan untuk :

- a. Mengembangkan desain pembelajaran pemrograman menggunakan model *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar.
- b. Menguji kelayakan desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar.
- c. Menguji efektivitas desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* siswa sekolah dasar.

1.5 State of The Art

Pembelajaran pemrograman dinilai sebagai cara yang paling banyak digunakan dan efektif dalam upaya meningkatkan keterampilan *computational thinking* siswa (Wong & Cheung, 2020; Zhang & Nouri, 2019). Dalam praktiknya pembelajaran pemrograman ini telah dicoba dengan menggunakan model *game-based learning* dan terbukti dapat meningkatkan keterampilan *computational thinking* siswa (Rose et al., 2018; Weintrop et

al., 2016). Namun secara bersamaan, penggunaan model ini membuat siswa cenderung untuk bekerja sendiri dalam menyelesaikan tugasnya (Lin et al., 2020) dan memiliki potensi yang besar pada kegagalan fokus pada diri siswa (Hooshyar et al., 2021) mengingat pembelajaran dengan model ini dinilai akan memiliki hasil yang lebih baik apabila siswa sudah mencapai tingkat tertentu pada keterampilan *computational thinking* (Zhao & Shute, 2019).

Selain dengan menggunakan model *game-based learning*, pembelajaran pemrograman yang coba diintegrasikan dengan aktivitas robotik juga terbukti dapat meningkatkan keterampilan *computational thinking* karena siswa cenderung memiliki motivasi yang lebih tinggi bahkan bagi siswa yang tidak memiliki informasi sebelumnya mengenai pemrograman karena aktivitasnya yang berorientasi kepada pengoperasian robot (Chalmers, 2018). Namun bukan tanpa masalah, penggunaan aktivitas robotik ini membutuhkan jaringan konsep yang solid melalui aktivitas pemecahan masalah yang seringkali gagal dipenuhi oleh guru (Çakır et al., 2021). Tidak tersedianya perangkat robotik yang murah dan *open-sources* juga menimbulkan permasalahan tersendiri (Bakala et al., 2021).

Di sisi lain, penerapan model *design-based learning* memiliki kelebihan yang dapat mengatasi permasalahan pada penerapan model *game-based learning* dan aktivitas robotik seperti kecenderungan siswa yang kurang mampu berkolaborasi dan kebutuhan akan aktivitas pemecahan masalah. Dimana pembelajaran pemrograman dengan model *design-based learning* terbukti dapat mendukung peningkatan keterampilan *computational thinking* (Jun et al., 2017a; Matere, Weng, Astatke, Hsia, Matere, et al., 2021) dan motivasi siswa (Garneli & Chorianopoulos, 2018, 2019; Repenning, 2015). Penerapan *design-based learning* pada pembelajaran pemrograman juga dapat meningkatkan keterampilan pemecahan masalah dan berfikir algoritmik (Saritepeci, 2020a) sekaligus kemampuan siswa berkolaborasi (Jun et al., 2017; Matere et al., 2021).

Namun penerapan model *design-based learning* ini dinilai belum optimal dalam upaya meningkatkan semua aspek pada keterampilan *computational thinking* siswa khususnya pada siswa usia sekolah dasar. Dari tujuh aspek

pada keterampilan *computational thinking*, hasil perolehan keterampilan dirasa masih kurang optimal pada beberapa aspek seperti aspek keterampilan *loops* (keterampilan menjalankan rangkaian perintah pemrograman secara berulang) dan keterampilan *conditionals* (keterampilan menjalankan instruksi pengambil keputusan berdasarkan kondisi tertentu). Kegagalan siswa dalam meningkatkan keterampilan pada dua aspek keterampilan *computational thinking* tersebut disebabkan karena siswa tidak mampu untuk menghubungkan dan mengkoordinasikan fakta dan keterampilan yang telah dipelajari sebelumnya (Matere et al., 2021) karena aktivitas *design-based learning* terkadang mengabaikan hubungan antara aktivitas konstruksi desain dengan beberapa keterampilan yang ditargetkan (Li et al., 2021). Terlalu luasnya spektrum pemilihan ide desain yang dapat dipilih siswa menimbulkan kecenderungan keterampilan yang ditargetkan menjadi tidak tercapai.

Kedua permasalahan tersebut sejalan dengan rekomendasi hasil penelitian Scherer *et al.* (2020) yang merekomendasikan agar pembelajaran pemrograman perlu dirancang untuk mengakomodasi gaya belajar dan kemampuan berpikir siswa sesuai dengan usianya. Selain itu Kong *et al.* (2018) lebih dulu menyarankan bahwa pembelajaran pemrograman juga perlu mengarahkan siswa agar menerapkan konsep dan praktis *computational thinking* untuk mengatasi masalahnya sehari-hari. Mengingat pembelajaran pemrograman dengan *design-based learning* seringkali memasukan semua aspek keterampilan *computational thinking* dalam satu rangkaian desain karya dimana urutan pentargetan keterampilan menjadi lebih fleksibel bergantung kepada setiap desain karya yang ingin siswa buat.

Hal ini tentunya berkaitan dengan perkembangan kognitif dan karakteristik belajar siswa usia belajar yang menurut Piaget (dalam Madjid, 2014: 10) berada pada tahap operasional konkret dimana memiliki kecenderungan belajar sebagai berikut: 1) konkret; 2) integratif/*holistic*; dan 3) hierarkis. Sementara itu penelitian Matere et al. (2021) merekomendasikan agar penerapan pembelajaran pemrograman dengan *design-based learning* dilakukan melalui aktivitas instruksi yang berulang agar dapat meningkatkan

memori siswa dalam mempertahankan konsep yang dipelajari yang akan mereka butuhkan ketika merancang sebuah program.

Berdasarkan penjabaran di atas, dapat diketahui bahwa perlunya sebuah desain pembelajaran yang dapat mengatasi permasalahan penargetan keterampilan yang terlalu fleksibel dan berpotensi tidak memfasilitasi ketujuh aspek keterampilan *computational thinking* dan kesulitan siswa untuk menghubungkan fakta dan konsep dengan keterampilan yang telah dipelajari sebelumnya. Oleh karena itu pada penelitian ini, peneliti akan mengembangkan sebuah desain pembelajaran pemrograman dengan model *design-based learning* yang dikemas dengan tema-tema konkret yang menargetkan keterampilan-keterampilan *computational thinking* secara hierarkis dengan aktivitas desain yang berulang di setiap temanya.

1.6 Road Map Penelitian

Penelitian pengembangan desain pembelajaran pemrograman menggunakan *design-based learning* untuk meningkatkan keterampilan *computational thinking* bagi siswa sekolah dasar ini dibagi menjadi tiga tahapan pencapaian sebagaimana tergambar pada *road map* penelitian di bawah ini:



Gambar 1.1 Road Map Penelitian