BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan *Generative Artificial Intelligence* (GenAI) telah mengubah lanskap tugas-tugas pemrograman secara otomatis, termasuk cara *developer* menulis dan menyelesaikan kode program, serta memungkinkan *developer* pemula untuk menghasilkan kode hanya dengan mengartikulasikan kebutuhannya secara alami [1]. Hal ini sejalan dengan meningkatnya pemanfaatan AI dalam pengembangan perangkat lunak, dari 76% pada tahun 2024 menjadi 84% pada tahun 2025 [2], [3]. Salah satu teknologi yang paling sering digunakan adalah GitHub Copilot, asisten pemrograman berbasis *Large Language Model* (LLM) yang sejak awal dikembangkan khusus untuk lingkungan pemrograman (IDE), sehingga mampu menghasilkan rekomendasi kode secara *real time* langsung di dalam editor sesuai konteks pemrograman pengguna [1].

Berdasarkan Stack Overflow Developer Survey 2025, GitHub Copilot menempati posisi kedua secara umum (67,9%) setelah ChatGPT (81,7%) [3]. Selisih ini wajar karena ChatGPT digunakan secara serba guna untuk berbagai jenis tugas di luar pemrograman, sementara GitHub Copilot secara khusus dirancang untuk konteks penulisan kode di dalam IDE, dan pada kategori tersebut GitHub Copilot memiliki tingkat adopsi tertinggi, jauh di atas Google Gemini (47,4%), Claude Code (40,8%), Microsoft Copilot (31.3%) [3]. Pada *earnings call* kuartal keempat tahun fiskal 2025, Microsoft melaporkan bahwa GitHub Copilot telah digunakan oleh 20 juta pengguna di seluruh dunia [4]. Tingginya tingkat adopsi tersebut menunjukkan bahwa AI, khususnya GitHub Copilot sebagai asisten yang menyatu dengan alur kerja pemrograman, telah menjadi bagian dari aktivitas pemrograman, termasuk dalam proses pembelajaran bagi *developer* pemula.

Bagi *developer* pemula, GitHub Copilot memberikan berbagai kemudahan dalam menyelesaikan tugas pemrograman. Penelitian menunjukkan

bahwa penggunaan Copilot membantu mahasiswa menyelesaikan tugas lebih cepat, meningkatkan efektivitas penyelesaian masalah, serta memungkinkan implementasi fitur yang lebih banyak dibandingkan tanpa bantuan AI [5], [6]. Kehadiran Copilot memberikan akses terhadap solusi secara instan sehingga mampu mengurangi hambatan teknis yang sering dihadapi selama proses belajar pemrograman.

Namun, kemudahan tersebut juga memunculkan tantangan baru, terutama bagi *developer* pemula. *Developer* pemula dalam penelitian ini didefinisikan secara operasional mengacu pada klasifikasi Ferino dkk. [7], yaitu mahasiswa aktif program studi rumpun ilmu komputasi yang belum pernah berstatus praktisi profesional pengembang perangkat lunak secara independen di industri bukan berdasarkan lamanya belajar pemrograman secara akademis maupun lamanya menggunakan GitHub Copilot. Status nonpraktisi ini berkorelasi dengan kemampuan metakognitif dan regulasi diri yang masih berkembang [8], sehingga kelompok ini cenderung belum memiliki pengalaman yang memadai untuk mengevaluasi kualitas rekomendasi AI secara kritis. Uraian lebih rinci mengenai definisi operasional ini, termasuk batasan kasusnya, disajikan pada 2.2 dan 3.4.

Penelitian menunjukkan bahwa pemula sering menerima kode yang dihasilkan AI tanpa memahami logika di balik solusi tersebut [9]. Kondisi ini berpotensi menimbulkan *AI overreliance*, yaitu kecenderungan terlalu bergantung pada rekomendasi AI dalam menyelesaikan tugas, yang pada akhirnya dapat mengurangi keterlibatan kognitif selama proses pemecahan masalah dan memunculkan *illusion of competence*, yaitu keyakinan bahwa seseorang telah memahami konsep pemrograman meskipun pemahamannya sangat bergantung pada bantuan AI [10]. Dengan demikian, penggunaan GitHub Copilot tidak hanya memengaruhi hasil penyelesaian tugas, tetapi juga dapat memengaruhi cara *developer* pemula belajar, mengambil keputusan, dan membangun pemahaman terhadap konsep pemrograman.

Berbagai penelitian terdahulu telah membahas penggunaan GitHub Copilot dari aspek produktivitas, efektivitas penyelesaian tugas, kualitas kode, maupun performa pengguna [5], [6], [11]. Penelitian mengenai aspek emosional dan efikasi diri juga mulai berkembang, tetapi umumnya masih menggunakan pendekatan kuantitatif melalui instrumen terstandar [12]. Akibatnya, penelitian terdahulu lebih banyak menjelaskan apa dampak penggunaan GitHub Copilot, sementara bagaimana *developer* pemula mengalami, memaknai, dan membangun interaksi dengan GitHub Copilot selama proses belajar pemrograman masih belum dipahami secara mendalam. Padahal, pengalaman tersebut dapat berbeda pada setiap individu dan dipengaruhi oleh berbagai faktor, seperti tujuan penggunaan, tingkat kepercayaan terhadap AI, pengalaman belajar sebelumnya, serta konteks penyelesaian tugas yang dihadapi. Kesenjangan tersebut menunjukkan bahwa pengalaman penggunaan GitHub Copilot oleh *developer* pemula masih memerlukan kajian yang lebih mendalam agar diperoleh pemahaman yang lebih utuh mengenai interaksi manusia dengan AI dalam konteks pembelajaran pemrograman.

1.2. Rumusan Masalah

Penggunaan GitHub Copilot semakin luas dalam pembelajaran pemrograman karena mampu membantu *developer* pemula menyelesaikan tugas secara lebih efektif. Namun, di balik manfaat tersebut, penggunaan GitHub Copilot juga berpotensi memengaruhi cara *developer* pemula berinteraksi dengan AI, membangun pemahaman, mengambil keputusan, serta mempertahankan keterlibatan dalam proses belajar. Berbagai penelitian terdahulu umumnya berfokus pada pengukuran produktivitas, efektivitas, dan performa penggunaan GitHub Copilot, sedangkan pengalaman penggunaan GitHub Copilot oleh *developer* pemula yang mencakup pola interaksi, pengalaman emosional, pemaknaan terhadap proses belajar, serta faktor-faktor yang memengaruhi keterlibatan mereka masih belum dipahami secara mendalam. Oleh karena itu, diperlukan pemahaman yang lebih komprehensif mengenai pengalaman penggunaan GitHub Copilot oleh *developer* pemula.

1.3. Pertanyaan Penelitian

Berdasarkan pernyataan pada rumusan masalah di atas, maka pertanyaan penelitian yang dikemukakan yaitu:

1. Bagaimana pola interaksi *developer* pemula dengan GitHub Copilot?
2. Bagaimana pengalaman emosional *developer* pemula terhadap penggunaan GitHub Copilot?
3. Bagaimana pemaknaan *developer* pemula terhadap pengaruh penggunaan GitHub Copilot pada proses belajar mereka?
4. Faktor apa saja yang mendorong atau menghambat keterlibatan aktif *developer* pemula saat menggunakan GitHub Copilot?

1.4. Batasan Penelitian

Agar penelitian ini lebih terfokus dan mendalam, ruang lingkupnya dibatasi sebagai berikut:

1. Penelitian ini tidak mengukur atau menganalisis kualitas objektif dari kode yang dihasilkan GitHub Copilot, seperti kebenaran sintaksis, efisiensi, atau keamanan kode. Fokus penelitian adalah pada pengalaman subjektif pengguna dalam berinteraksi dengan alat tersebut.
2. Penelitian ini tidak mencari hubungan kausal atau pengaruh statistik antar faktor, melainkan berfokus pada identifikasi dan pemaknaan faktor-faktor tersebut berdasarkan pengalaman subjektif partisipan yang dieksplorasi secara kualitatif.
3. Penelitian ini dibatasi pada *developer* pemula, yaitu individu atau mahasiswa yang sedang mempelajari pemrograman dan masih berada pada tahap pembentukan keterampilan dasar, yang belum memiliki model mental dan strategi pemrograman yang matang sebagaimana *developer* berpengalaman, serta belum memasuki karier profesional penuh waktu sebagai *developer*.

4. Hasil penelitian ini bersifat kontekstual dan tidak dimaksudkan untuk digeneralisasikan secara statistik, sesuai dengan pendekatan kualitatif yang digunakan.

1.5. Tujuan Penelitian

Sesuai dengan pertanyaan penelitian yang telah dirumuskan, maka tujuan dari penelitian ini adalah:

1. Menganalisis pola interaksi *developer* pemula dengan GitHub Copilot.
2. Mengeksplorasi pengalaman emosional *developer* pemula terhadap penggunaan GitHub Copilot.
3. Menganalisis pemaknaan *developer* pemula terhadap pengaruh penggunaan GitHub Copilot pada proses belajar mereka.
4. Mengidentifikasi dan mengkategorikan faktor-faktor yang mendorong serta menghambat keterlibatan aktif *developer* pemula saat menggunakan GitHub Copilot.

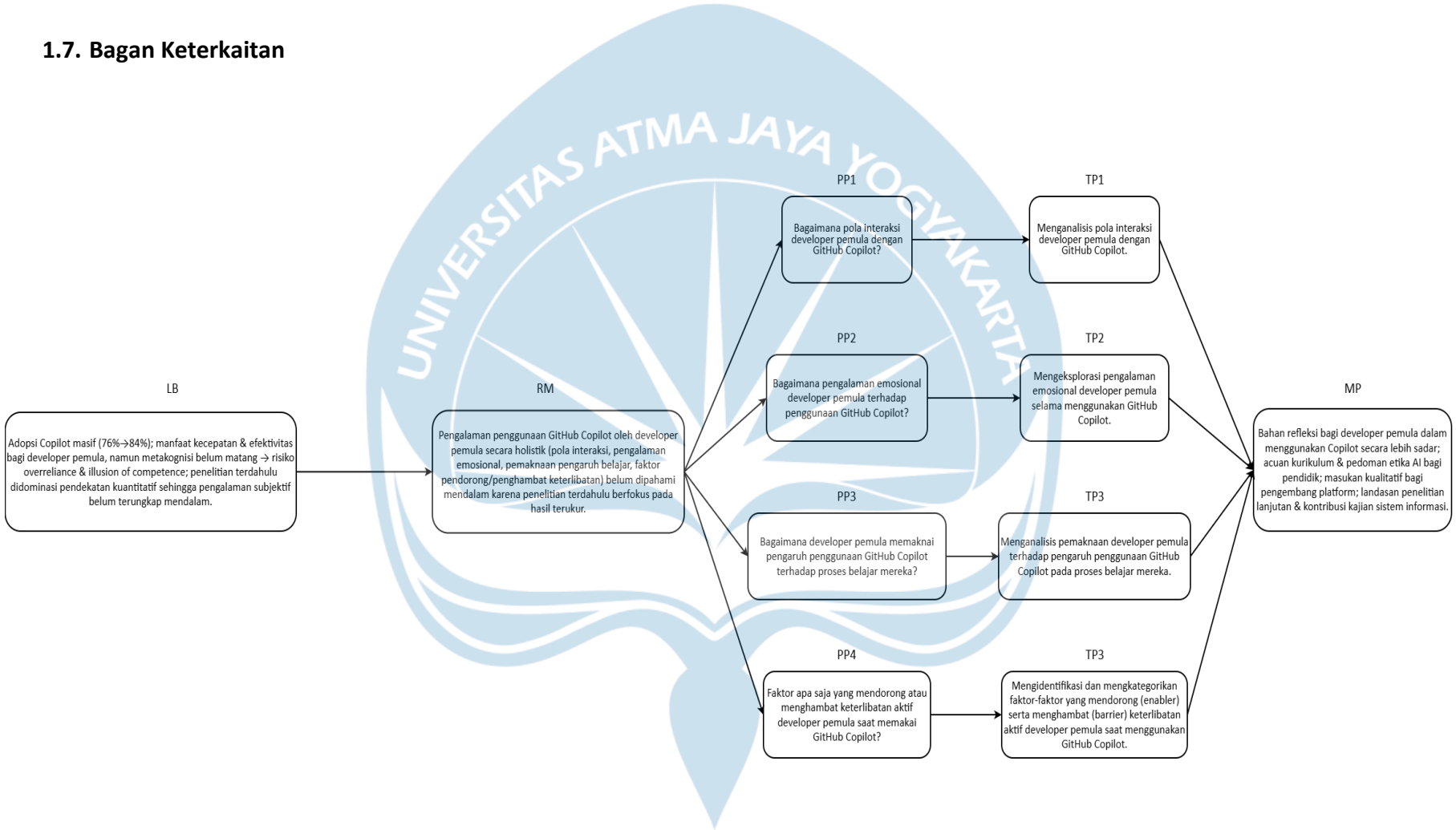
1.6. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat baik secara praktis maupun akademis.

Secara praktis, temuan penelitian ini dapat menjadi bahan refleksi bagi *developer* pemula dalam menggunakan GitHub Copilot secara lebih sadar dan efektif. Bagi pendidik dan institusi pendidikan, hasil penelitian ini dapat digunakan sebagai acuan dalam merancang kurikulum serta pedoman etika penggunaan AI dalam pembelajaran pemrograman. Bagi *developer* platform, penelitian ini memberikan masukan kualitatif mengenai pengalaman pengguna pemula yang dapat dimanfaatkan untuk pengembangan produk.

Secara akademis, penelitian ini memperkaya kajian sistem informasi, khususnya terkait interaksi manusia dengan asisten pemrograman berbasis AI, serta menjadi landasan bagi penelitian lanjutan mengenai adopsi AI dalam pendidikan dan pengembangan keterampilan.

1.7. Bagan Keterkaitan



Gambar 1.1. Bagan Keterkaitan