



# Apa Itu Pemrograman?

*... dan apa itu pemrograman di era **Kecerdasan** **Artifisial**?*

Achmad Benny Mutiara

Mei 2025



# Pendahuluan

- Pertanyaan sederhana:
  - Apa artinya **pemrograman**? Atau Apa itu **pemrograman**?
- Pertanyaan ini mungkin tampak konyol karena dua alasan. Kita mungkin mendapati diri kita memiliki definisi dan konseptualisasi yang sangat jelas dan ringkas tentang (gagasan) pemrograman. Atau, kita mungkin memper-tanyakan nilai apa yang didapat dari membahas *pertanya-an sepele seperti itu*.



# Pendahuluan

- Namun, kita yakin jawaban atas pertanyaan ini dapat memberikan pencerahan tentang masa depan pemrograman di era **AI-generatif**.
- Kita mendekati jawaban atas pertanyaan tersebut dengan mene-lusuri sejarah komputasi serta pendapat di antara para pengajar pemrograman pengantar kontemporer.

# Secara Teori, Pemrograman adalah ...



- Karya Alan Blackwell membawa kita melalui definisi historis **pemrograman**
  - Awalnya, hal ini disajikan sebagai proses penyusunan “jadwal” untuk urutan operasi individual yang diperlukan untuk melakukan perhitungan.
  - Definisi selanjutnya memperluas hal ini ke proses penerjemahan dari bahasa yang mudah dipahami manusia ke bahasa yang mudah dipahami komputer.
- Jika kita, sebagai sebuah komunitas, mengadopsi definisi yang sempit ini, keseluruhan pemrograman mungkin akan menjadi **tidak relevan lagi** karena kemampuan **AI-generatif** yang terus berkembang.

# Secara Teori, Pemrograman adalah ...



- Namun, selama tahun 1970-an, **Donald Knuth**, dan **Edsger Dijkstra**, berpendapat bahwa pemrograman adalah sesuatu yang lebih dari itu. Mereka berpendapat bahwa
  - **pemrograman** adalah “**seni**” menyusun program, yang membutuhkan kesadaran akan estetika, penggunaan kecerdikan, dan kreativitas bawaan.
  - Dari perspektif ini, ***pemrograman lebih tentang menyusun potongan kode untuk memecahkan masalah daripada menulis baris instruksi individual.***

# Secara Teori, Pemrograman adalah ...



- Pada tahun 1985, **Peter Naur** lebih jauh memberikan nuansa pada pandangan pemrograman untuk pemecahan masalah. Secara khusus, ia berpendapat bahwa
  - "kehidupan" suatu program bergantung pada pengembangnya yang mengembangkan pemahaman tentang bagaimana program tersebut [yang mereka kembangkan atau kelola] mendukung masalah yang ada; yaitu, pemahaman tentang "ranah masalah."
  - Jadi, **pemrograman bukan sekadar apa** yang diwujudkan sebagai kode program; **tetapi juga** pengetahuan tentang bagaimana program berinteraksi dan berkembang di samping ranah masalah.

# Secara Teori, Pemrograman adalah ...



- Baru-baru ini, **Amy Ko dkk.** telah mengambil langkah lebih jauh dan mengemukakan perlunya juga mempertimbangkan dampak sosial dari program kita, termasuk keterbatasan, bias, dan pertimbangan etika yang tertanam dalam program tersebut.

# Secara Teori, Pemrograman adalah ...



- Dengan demikian, seseorang dapat mereduksi **pemrograman** menjadi
  - tindakan menulis instruksi, tetapi ketika mempertimbangkan seni dan konteks, **pemrograman** menjadi lebih tentang proses menyusun dan membangun sesuatu yang, dalam arti tertentu, ada dalam suatu domain.
  - Dengan demikian, definisi saat ini merangkum tindakan menulis program dan kecerdikan dalam menyusun bagian-bagian untuk memecahkan masalah.



# Secara Teori, Pemrograman adalah ...



- Penelitian dalam pendidikan komputasi menceritakan kisah yang sama:
  - hambatan dalam belajar **pemrograman** lebih dari sekadar menulis instruksi individual;
  - tantangannya lebih luas tentang membaca kode, menyusun kode, merencanakan struktur, memecahkan masalah, dan memvalidasi solusi.

# Dalam Praktik, Pemrograman adalah..



- Mari kita ubah perspektif dari teori ke praktik dan pertimbangkan apa yang dianggap oleh para pendidik sebagai pemrograman. Kita mungkin bertanya mengapa kita fokus pada para pendidik.
- Untuk definisi formal (praktis) tentang ‘pemrograman,’ tentunya kita harus beralih ke praktisi: pengembang, insinyur, dan programmer?

# Dalam Praktik, Pemrograman adalah..



- Namun, para pendidik sering kali menjadi pihak yang memfasilitasi pertemuan pertama mahasiswa dengan pemrograman. Akibatnya, perspektif pendidik akan menentukan bagaimana para pemula akan menghadapi pembelajaran pemrograman (di era **AI generatif**). Jadi, kita merasa penting untuk mengeksplorasi sudut pandang mereka karena hal itu dapat menentukan pemahaman generasi berikutnya tentang topik tersebut.

# Dalam Praktik, Pemrograman adalah..



- Oleh karena itu, kita baru-baru ini melakukan survei di mana kita mempertanyakan apa yang diyakini oleh para pengajar pemrograman pemula tentang pemrograman, dan menjadi seorang programmer yang baik, melalui dua pertanyaan terbuka.
- Kita secara khusus berfokus pada pendidikan tinggi di **Denmark**. karena informasi tentang mata kuliah, program, dan guru (siapa yang mengajar apa) bersifat publik.

# Dalam Praktik, Pemrograman adalah..



- Lebih jauh, kelas komputasi tidak wajib di sekolah dasar maupun di semua sekolah menengah, yang berarti bahwa pendidikan tinggi adalah garis depan pendidikan tempat mahasiswa pertama kali mengenal pemrograman.
- Dikombinasikan dengan ukuran negara yang terbatas, hal ini memungkinkan kita untuk menyelidiki **1.169** program secara komprehensif dan dengan demikian mengumpulkan tanggapan dari populasi pendidik yang luas dan representatif di seluruh disiplin ilmu.

# Dalam Praktik, Pemrograman adalah..



- Dengan menyelidiki kurikulum semua mata kuliah wajib pada program-program ini, kita mengidentifikasi program pendidikan yang memperkenalkan pemrograman. Ini merupakan "gambaran singkat" tentang bagaimana pemrograman wajib telah muncul dalam pendidikan tinggi di seluruh negara (Denmark).
- Hasilnya mencakup 130 mata kuliah wajib yang memperkenalkan pemrograman yang dibagikan di 175 program. Dengan mendistribusikan survei kita kepada 146 pendidik yang mengajar mata kuliah ini, kita memperoleh 81 tanggapan (55%): dari guru (26%), asisten kursus (67%), dan lainnya,

# Dalam Praktik, Pemrograman adalah..



- seperti kepala program studi (7%). Responden terlibat dalam 61% dari program yang diidentifikasi (106 dari 175) yang mencakup sebagian besar bidang disiplin ilmu (sebagaimana didefinisikan oleh daftar pendidikan pemerintah). Jadi, tidak mengherankan, mengingat cakupan disiplin ilmu yang luas, kita melihat keragaman gender yang lebih seimbang (dalam hal ini, sekitar 2:1) daripada rasio 4:1 yang terkait dengan TIK di Eropa. Akhirnya, kita menemukan bahwa 40% responden berasal dari negara lain selain Denmark. Oleh karena itu, kita menyimpulkan bahwa tanggapan yang dikumpulkan memang mewakili kelompok pendidik yang sangat beragam.



# Dalam Praktik, Pemrograman Adalah..

- Dengan mengkategorikan respons menurut penggunaan kata kunci, kita menemukan bahwa
  - 36% mendefinisikan pemrograman dalam hal kemampuannya untuk memecahkan masalah.
  - 35% berfokus pada penulisan baris kode/instruksi dalam definisi pemrograman mereka.
  - Sementara beberapa jawaban berfokus pada kualitas pemrograman lainnya, yang disebutkan di atas tampaknya menjadi atribusi dominan (sempit) pemrograman.
  - Dimensi lain yang terlibat: pentingnya komputer (28%), algoritma (10%), penggunaan bahasa pemrograman (9%), eksekusi kode (6%), atau otomatisasi (5%).





# Dalam Praktik, Pemrograman Adalah..

- Menurut pengajar pengantar pemrograman:
  - definisi pemrograman sebagian besar berfokus (secara sempit) pada kemampuan untuk menginstruksikan komputer.
  - Dengan kata lain, pengajar mengabaikan untuk menekankan konteks pemrograman yang lebih luas yang dibahas sebelumnya.
  - Masuk akal jika responden menanggapi dengan impuls daripada refleksi yang dipertimbangkan dengan baik. Kita juga dapat berspekulasi bahwa tanggapan tersebut hanya mengartikulasikan "penyebut umum terkecil", yang lebih mudah diakses oleh audiens yang kurang memahami komputasi; seperti dengan sengaja menyederhanakan tindakan pemrograman agar lebih mudah dipahami oleh audiens luar.



# Dalam Praktik, Pemrograman Adalah..

- Namun, pengamatan tersebut masih berlaku: kita cenderung merenungkan pemrograman sebagai 'tindakan menulis program' dan tidak mempertimbangkan kualitas yang lebih luas yang tidak dapat diraba.
- Namun, satu hal yang pasti: tidak ada definisi hegemon tunggal.

# Seorang Programmer yang Baik adalah ...



- Masa depan **pemrograman** tidak semata-mata ditentukan oleh definisi kita tentang **pemrograman**;
- Kita juga harus mempertimbangkan peran seorang programmer.
- Saat ini, konsep seorang '**programmer**'
  - harus lebih luas daripada pengembang profesional yang bekerja di kantor, berkembang hingga mencakup pengembang pengguna akhir atau **programmer** vernakular/percakapan,
  - tidak harus mengidentifikasi diri sebagai '**programmer**' meskipun mereka mungkin melakukan hal yang persis seperti itu: pemrograman

# Seorang Programmer yang Baik adalah ...



- Singkatnya,
  - sekelompok ‘**programmer**’, yang setiap hari mengembangkan dan memelihara program, skrip, dan perangkat lunak yang menjalankan fungsi bisnis penting, berpotensi dalam lingkungan pemrograman yang tidak konvensional dan
  - sekelompok individu yang harus berkolaborasi secara simbiosis dengan pengembang profesional untuk mengembangkan dan memelihara perangkat lunak, yang memerlukan beberapa tingkat pengetahuan teknis.

# Seorang Programmer yang Baik adalah ...



- Mengingat definisi programmer yang (sekarang) lebih luas, penting untuk mengakui perbedaan potensial antara programmer vernakular yang baik dan programmer profesional.
- Perhatikan bahwa responden survei kita mengajarkan pemrograman dengan berbagai tujuan; khususnya, melatih baik insinyur perangkat lunak profesional maupun programmer percakapan. Tanpa konteks lebih lanjut, kita bertanya kepada mereka apa artinya menjadi programmer yang baik.

# Seorang Programmer yang Baik adalah ...



- Meskipun ada perbedaan antara ‘melakukan pemrograman,’ dan ‘menjadi programmer (yang baik)’, kita menganggap ini sebagai pertanyaan yang berguna untuk merenungkan lebih jauh tentang apa yang kita ajarkan kepada generasi berikutnya.
- Dengan melakukan kategorisasi yang sama seperti pertanyaan terbuka sebelumnya, para pendidik berpendapat bahwa
  - pemecahan masalah memainkan peran penting (20%).

# Seorang Programmer yang Baik adalah ...



- Namun, dan yang lebih menarik, kualitas seperti
  - kemampuan untuk mengabstraksi dan menguraikan (20%),
  - menyusun dan membuat kode yang dapat dibaca (20%),
  - bernalar dan bekerja secara logis (11%), dan kreativitas (9%) muncul dalam jawaban mereka—sesuatu yang tidak kita lihat ketika mereka mendefinisikan pemrograman.
- Lebih jauh, hanya ada sedikit atau tidak ada penyebutan tentang pentingnya menulis instruksi untuk komputer.

# Seorang Programmer yang Baik adalah ...



- Dengan demikian, ketika para pendidik diminta untuk mempertimbangkan kualitas seorang **programmer** yang baik, baik yang bersifat umum maupun profesional, tanggapan mereka, pada tingkat yang lebih luas, mencakup kualitas yang tidak kentara yang terkait dengan definisi literatur tentang pemrograman.
- Meskipun kata kunci yang disebutkan telah berubah, sekali lagi tampaknya tidak ada konsensus mengenai mana yang paling penting.



# Pemrograman di Era **AI** adalah ...



- Sementara definisi dalam literatur dengan jelas menekankan kualitas pemrograman yang lebih tidak kentara, para pendidik tidak melakukannya—setidaknya sampai kita mempertimbangkan apa artinya menjadi seorang programmer (yang baik).
- Lebih jauh, tampaknya tidak ada konsensus di antara para pendidik. Meskipun perbedaan ini tidak selalu menjadi masalah dalam komunitas tempat kita sangat memahami seluk-beluk definisi tersebut, hal itu bisa jadi menjadi masalah bagi mereka yang sekarang mempelajari pemrograman.

# Pemrograman di Era **AI** adalah ...



- Apakah kita hanya memberi tahu mahasiswa bahwa pemrograman adalah tindakan menulis instruksi?
- Apakah kita bahkan menjelaskan apa itu 'pemrograman'?
- Jika tidak, masuk akal untuk mengharapkan mahasiswa yang tidak terbiasa dengan domain tersebut untuk mencoba mengalihdaya-kan semua tugas mereka ke **AI-generatif**.

# Pemrograman di Era **AI** adalah ...



- Seperti yang dinyatakan oleh Michael Caspersen, *“[...] hal ini akan menjadi bahaya bagi mahasiswa yang tidak disiplin yang mencari jalan keluar yang mudah.”*
- Kita percaya bahwa kurangnya konsensus di antara perspektif yang dikumpulkan dan disajikan di sini merupakan bukti tidak adanya percakapan semacam itu.
- Dengan ini, kita tidak bermaksud untuk menemukan definisi tunggal tentang pemrograman, dan kita juga tidak menganggapnya produktif.

# Pemrograman di Era **AI** adalah ...



- Sebaliknya, kita berharap agar semua orang mempertimbangkan dengan saksama apa itu ‘pemrograman’, dan, pada gilirannya, menjelaskannya secara eksplisit kepada orang-orang di sekitar kita.
- Kita berpendapat bahwa pemrograman menyangkut kualitas yang tidak dapat diraba yang dijelaskan di sini, yang jauh lebih sulit untuk dialihdayakan ke **AI-generatif**.

# Pemrograman di Era **AI** adalah ...



- Salah satu cara untuk mencapai hal ini adalah dengan memberi insentif kepada mahasiswa yang **tidak menggunakan AI-generatif** pada tahap awal pendidikan pemrograman untuk mengembangkan pemahaman yang mendalam tentang pemrograman; misalnya, dengan menggunakan penilaian mahasiswa tertulis atau lisan yang diawasi (untuk mendidik para ahli).
- Pendekatan yang berbeda adalah dengan meningkatkan fokus pada pengajaran pemahaman program analitis sambil menerapkan **AI-generatif** untuk mensintesis instruksi, mirip dengan Porter dan Zingaro (untuk mendidik para editor).

# Pemrograman di Era **AI** adalah ...



- Meskipun ada banyak arah yang potensial, semuanya dimulai dengan berdiskusi dengan mahasiswa dan rekan kerja tentang apa itu '**pemrograman**'.
- Oleh karena itu, kita mohon agar para pendidik dan pendukung mengomunikasikan **kualitas pemrograman** yang tidak berwujud yang akan terus menjadi hal yang penting—bahkan di era **AI**.