Kemampuan mengenali pola dari sejumlah besar data sekaligus beradaptasi terhadap data baru adalah hal yang paling menarik, kuat, menggugah  dari machine learning. Sebab seperti kita tahu, pemrograman tradisional memiliki keterbatasan dengan sifatnya yang rigid, dengan sekumpulan aturan “if” dan “else” untuk memproses data atau menyesuaikan dengan masukan. Menurut Muller [3], penggunaan aturan untuk penentuan keputusan dengan cara ini memiliki dua kelemahan utama:

* Logika yang digunakan untuk membuat keputusan bersifat spesifik pada ranah dan masalah tertentu. Mengubah sedikit masalah, membuat kita mungkin perlu menulis keseluruhan sistem.
* Mendesain aturan untuk sistem memerlukan pemahaman yang mendalam tentang bagaimana suatu keputusan harus dibuat oleh seorang ahli.

Padahal permasalahan terus muncul seiring waktu serta arus informasi dan data pun terus berubah. Machine learning, dengan kemampuannya yang secara alami bersifat adaptif terhadap data dan masukan baru, menawarkan solusi untuk masalah tersebut. Sebagai gambaran, berikut adalah contoh dua kategori permasalahan yang dapat diselesaikan dengan baik oleh algoritma machine learning.

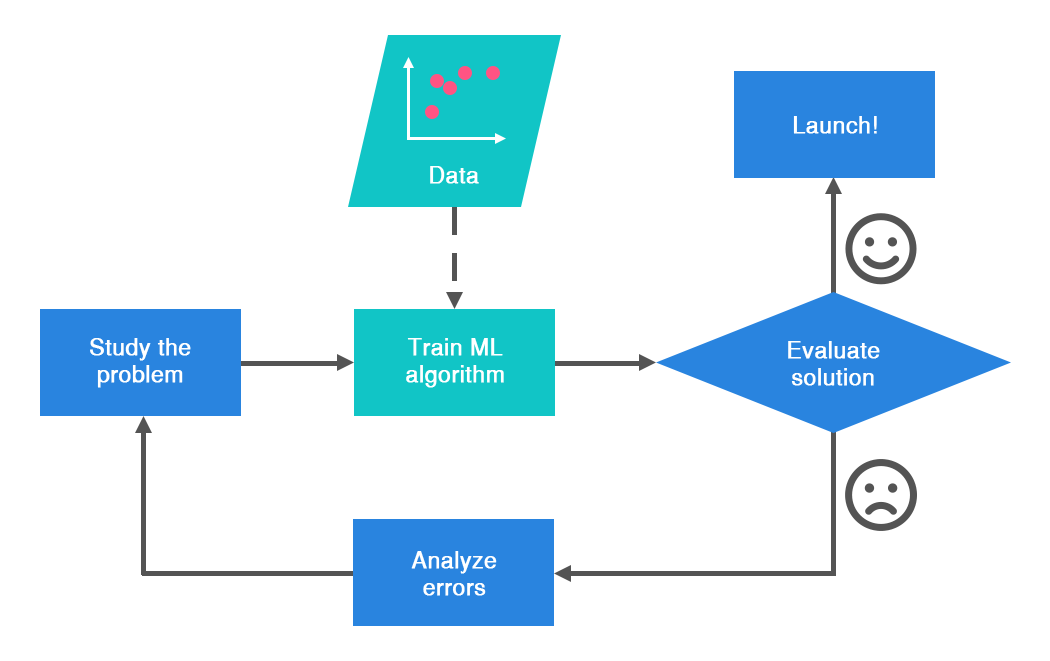
**Masalah yang Solusinya Membutuhkan Banyak Penyesuaian dan Aturan**

Bayangkan jika kita bertugas mengembangkan sebuah aplikasi filter spam dengan pemrograman tradisional. Langkah-langkah konvensional yang perlu kita lakukan adalah sebagai berikut.

Pertama, kita akan mendefinisikan bagaimana sebuah email termasuk kategori spam atau bukan. Misalnya, kita mengidentifikasi bahwa pada email spam umumnya terdapat kata-kata seperti “kaya”, “instan”, dan “murah”. Kemudian, kita menulis algoritma untuk setiap pola yang kita temukan pada email spam. Program pun akan menandai sebuah email spam jika menemui pola terkait. Terakhir, kita akan mengulangi kedua langkah tadi sampai program kita cukup baik untuk diluncurkan.

Karena kita menulis program menggunakan cara tradisional, hasilnya tentu daftar panjang berisi aturan-aturan rumit yang sulit untuk di-*maintain*, atau harus di-*update* secara berkala saat kita menemukan kosakata dan pola baru yang terkait dengan email spam.

Mari bandingkan jika kita menggunakan ML untuk mengembangkan filter spam tersebut. ML akan secara otomatis mempelajari pola kata-kata yang menentukan sebuah email spam atau bukan. Program dengan ML pun menjadi relatif lebih sederhana dan mudah untuk dipelihara. Seperti digambarkan oleh Geron [2], *flowchart* di bawah menunjukkan bagaimana alur pengembangan sebuah proyek Machine Learning.



**Masalah Rumit yang Tidak Bisa Diselesaikan dengan Pemrograman Tradisional**

Ada beberapa permasalahan yang sampai saat ini belum bisa dipecahkan dengan pendekatan pemrograman tradisional. Penyebabnya bisa jadi karena masalah tersebut terlalu rumit atau bisa juga karena belum diketahui algoritma pemrograman yang tepat untuk kasus tersebut.

Misal, kita ingin membangun sebuah sistem pengenalan suara. Menurut Geron [2], tahapan pertama yang perlu dilakukan adalah dengan membuat algoritma sederhana yang mampu membedakan kata “bagus” dan “indah”. Perhatikan bahwa kata “bagus” dimulai dengan suara bernada tinggi (“b”). Sebaliknya, kata “indah” dimulai dengan suara bernada rendah. Sehingga kita dapat membuat program untuk algoritma yang mengukur intensitas suara, kemudian menggunakannya untuk membedakan kedua kata tersebut.

Pertanyaannya, dapatkah hal ini kita terapkan pada ribuan jenis kata, yang diucapkan oleh jutaan orang, dalam berbagai macam bahasa, di lingkungan yang mungkin terdistraksi oleh suara-suara lainnya? Sangat rumit, bukan? Maka solusi terbaik yang dapat kita lakukan saat ini adalah, membuat algoritma yang dapat belajar dengan sendirinya melalui banyak data berupa sampel rekaman untuk setiap kata.

Tentu Anda dapat menerka, algoritma apa yang dapat menyelesaikan permasalahan ini? Betul, machine learning!