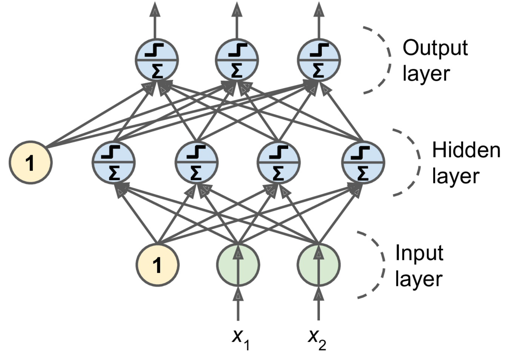
Multi Layer Perceptron (MLP) adalah sebuah jaringan saraf yang terdiri dari satu layer input, satu atau lebih hidden layer, dan satu output layer. MLP yang memiliki banyak *hidden layer* disebut juga *Deep Neural Network* (DNN).

Gambar berikut [21] adalah arsitektur dari sebuah multilayer perceptron dengan dua input, satu hidden layer yang memiliki 4 neuron, dan tiga output neuron. Dalam ilustrasi ini juga kita bisa melihat bias, meskipun biasanya bias ini bersifat implisit. Setiap layer (kecuali output layer) memiliki satu neuron bias dan terhubung sepenuhnya dengan lapisan berikutnya. Pada arsitektur ini sinyal hanya mengalir dalam satu arah (dari input ke output), jadi arsitektur ini adalah contoh dari*feedforward neural network* (FNN).



Sebuah jaringan saraf tiruan seperti MLP mirip dengan seorang bayi baru lahir yang belum mengerti apa-apa. Dengan belajar lah seorang bayi dapat meningkat pengetahuannya tentang dunia sekitar. Begitu juga dengan MLP. Ketika kita melatihnya pada sebuah data, kita ingin agar MLP membuat kesalahan yang sangat minim pada prediksinya.

Selama bertahun-tahun banyak peneliti yang kesulitan menemukan cara untuk melatih MLP. Barulah pada tahun 1986, David Rumelhart, Geoffrey Hinton, dan Ronald Williams mempublikasikan penelitian yang mengenalkan propagasi balik alias*backpropagation*[21]. Propagasi balik adalah sebuah algoritma untuk melatih MLP yang masih digunakan hingga sekarang.

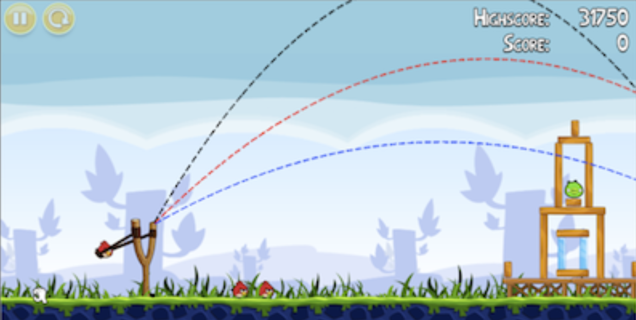
Geron dalam bukunya [4] menyatakan bahwa backpropagation adalah gradient descent dengan teknik yang efisien untuk menghitung gradien secara otomatis hanya dalam dua lintasan (satu maju dan satu mundur). Algoritma ini mampu menghitung gradien kesalahan jaringan dengan memperhatikan setiap parameter model. Ia dapat mengetahui bagaimana setiap bobot koneksi dan bias harus disesuaikan untuk mengurangi kesalahan. Setelah memperoleh gradien yang diinginkan, proses dilanjutkan dengan gradient descent biasa dan seluruh proses diulang sampai jaringan menyatu dengan solusi.

**Propagasi Balik**

Algoritma propagasi balik memungkinkan MLP untuk belajar membuat prediksi menjadi semakin baik dengan suatu teknik yang disebut chain rule. Algoritma ini bekerja dengan cara berikut [4].

* Algoritma ini menangani satu kelompok kecil dalam satu waktu dan melewati keseluruhan set training beberapa kali. Tiap satu lintasan set training kita menyebutnya dengan istilah satu epoch.
* Selanjutnya algoritma menghitung output semua neuron dalam satu layer. Hasilnya diteruskan ke layer berikutnya, dihitung lagi outputnya, dan diteruskan lagi ke layer berikutnya. Demikian seterusnya sampai kita mendapatkan output dari layer terakhir, layer output. Proses ini disebut forward pass, mirip seperti teknik membuat prediksi. Bedanya adalah seluruh hasil pada tahapan menengah disimpan untuk proses backward pass (umpan mundur).
* Keluaran hasil prediksi kemudian diukur tingkat erornya menggunakan loss function dengan cara membandingkan output yang diinginkan dan output dari jaringan.
* Algoritma kemudian melewati setiap lapisan secara terbalik atau mundur hingga mencapai layer input untuk mengukur kontribusi kesalahan dari setiap kesalahan. Proses ini dilakukan secara analitis dengan menerapkan aturan rantai atau chain rule yang membuat langkah ini cepat dan tepat.
* Terakhir, algoritma melakukan langkah penurunan gradien untuk mengubah dan  menyesuaikan bobot koneksi di jaringan. Proses ini bertujuan untuk meminimalisir eror.

Sebuah contoh sederhana dari propagasi balik adalah pada permainan Angry Birds. Kita dapat memenangkan permainan dengan memilih lintasan terbaik untuk mengenai target.

[](https://www.dicoding.com/academies/184/tutorials/8507?from=8502)

Pada awal permainan MLP akan memilih lintasan secara acak. Pada pemilihan lintasan pertama, MLP sangat mungkin menghasilkan eror berupa tak kena target. Eror kemudian diukur dengan loss function di mana eror merupakan jarak antara lintasan yang tidak mengenai target dan target yang dituju. Eror kemudian dikirim dengan propagasi balik dan setiap bobot pada MLP disesuaikan dengan optimizer.

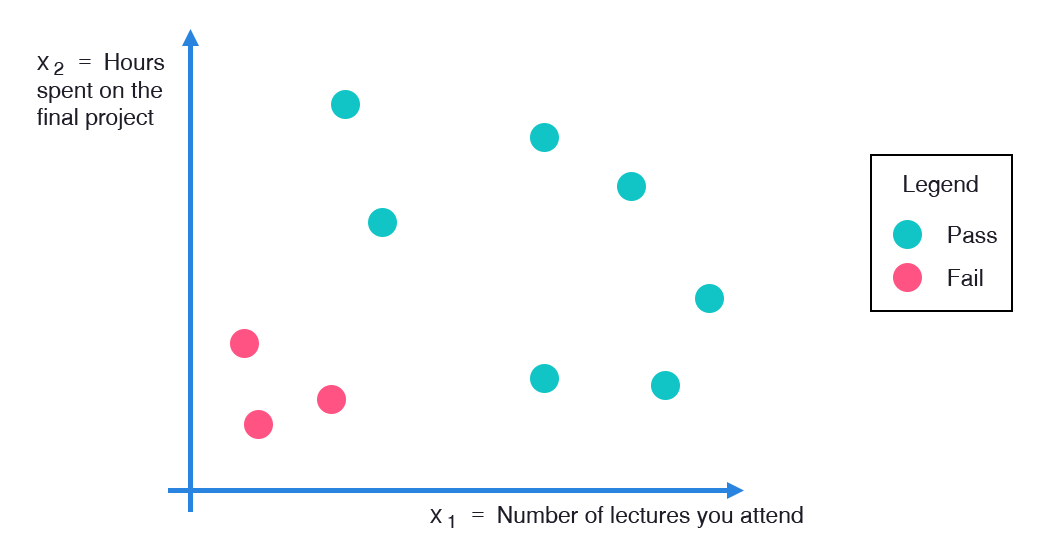
Pada kasus ini bobot yang perlu dipelajari MLP adalah lintasan terbaik yang harus dilalui untuk memenangkan permainan. Kemudian MLP akan terus belajar dengan propagasi balik hingga akhirnya menemukan bobot/lintasan yang berhasil mengenai target. Mudah bukan.

Untuk memahami cara kerja propagasi balik secara detail maupun rumus matematika dibaliknya, silakan kunjungi tautan-tautan berikut yah: [tautan 1](https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd)dan [tautan 2](https://brilliant.org/wiki/backpropagation/).

**Klasifikasi dengan MLP**

Setelah kita memahami perceptron, multi layer perceptron, dan propagasi balik, kita dapat melihat bagaimana MLP bekerja. MLP adalah model machine learning kategori supervised sehingga MLP dapat dipakai dalam kasus klasifikasi dan regresi.

Kita akan melihat bagaimana MLP bekerja dalam kasus klasifikasi. Pada gambar di bawah terdapat sebuah kasus klasifikasi untuk menentukan apakah seorang murid akan lulus atau tidak dari sebuah kelas. Pada data terdapat 2 atribut yaitu jumlah presensi dan lama waktu yang dihabiskan pada tugas akhir. Dan label pada data ada 2 yaitu lulus dan gagal.

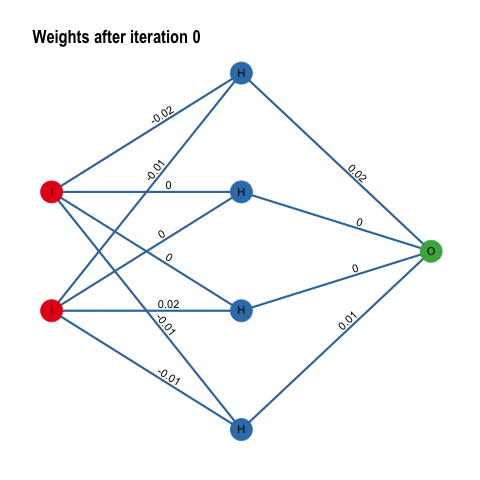


Untuk menyelesaikan masalah di atas kita dapat menggunakan MLP dengan satu hidden layer yang memiliki 4 perceptron. Input pada MLP merupakan 2 atribut pada data dan output memiliki 1 perceptron yang akan mengeluarkan 2 prediksi kelas.

Pada masalah klasifikasi 2 kelas (biner) kita dapat menggunakan 2 perceptron untuk output di mana satu perceptron mewakili sebuah kelas. Namun khusus untuk klasifikasi biner kita dapat menggunakan hanya 1 perceptron pada output layer yang mengeluarkan probabilitas untuk masing-masing kelas (probabilitas>0.5 memprediksi lulus dan probabilitas <0.5 memprediksi tidak lulus). Untuk masalah klasifikasi 3 kelas ke atas, jumlah perceptron pada layer output sebaiknya menyesuaikan jumlah kelas pada data.

Animasi di bawah menunjukan sebuah MLP untuk kasus klasifikasi kelulusan siswa yang kita bahas sebelumnya. Pada MLP di bawah terdapat 2 input dan sebuah perceptron output yang mengeluarkan probabilitas untuk setiap kelas. Untuk hidden layer, MLP di bawah menggunakan 4 buah perceptron.

Tak ada aturan baku tentang pemilihan jumlah hidden layer dan banyak perceptronnya. Jumlah hidden layer dapat disesuaikan dengan kerumitan masalah yang akan diselesaikan.

[](https://www.dicoding.com/academies/184/tutorials/8507?from=8502)

Pada animasi di atas kita dapat melihat bobot pada setiap garis yang menghubungkan perceptron. Bobot tersebut adalah parameter yang dipelajari oleh MLP agar menghasilkan prediksi yang benar. Proses ini sama dengan contoh Angry Birds sebelumnya di mana bobot pada setiap iterasi akan disesuaikan dengan propagasi balik.

Ketika telah melalui beberapa iterasi kita dapat melihat beberapa garis menjadi lebih tebal dari lainnya. Garis yang lebih tebal menunjukkan bahwa bobotnya lebih tinggi. Dari seluruh bobot yang dipelajari, prediksi MLP pada iterasi terakhir akan menjadi lebih tepat.

Selamat! Anda telah memahami bagaimana MLP bekerja. Pada modul selanjutnya kita akan menulis program MLP dengan TensorFlow.