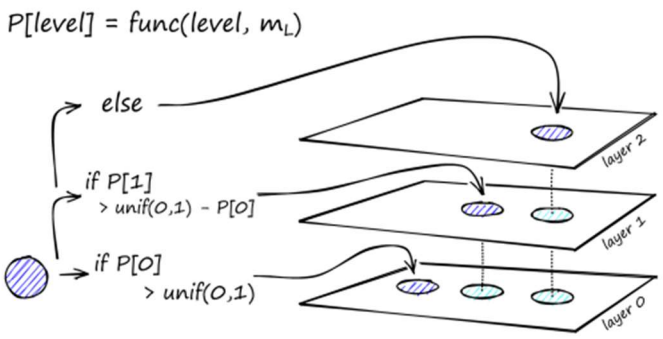
Selamat siang Pa Alfan,

Nama saya Syahrul Apriansyah dengan NPM 2106708311 dari kelas IR izin untuk menyampaikan hasil eksplorasi saya terkait HNSW Graf

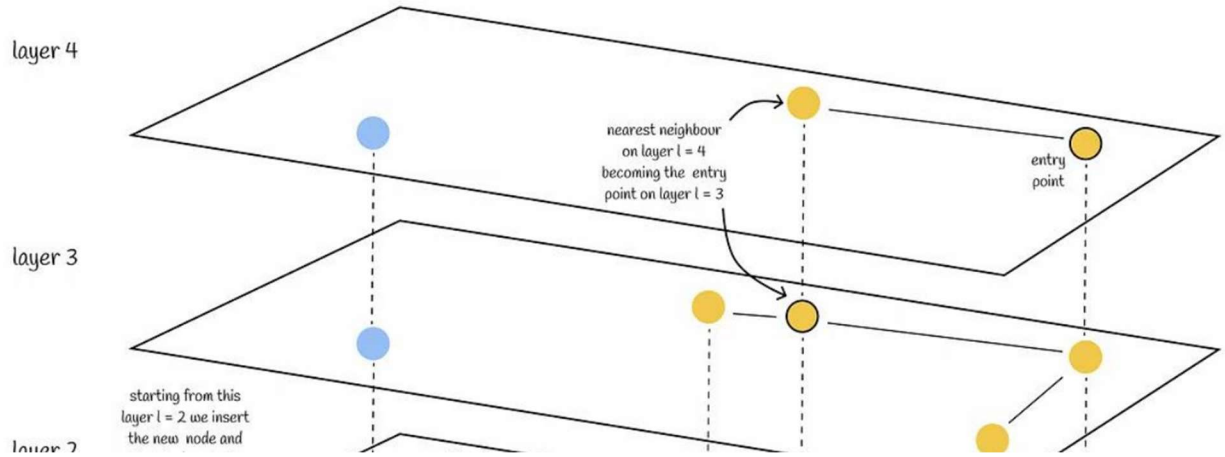
Referensi: https://towardsdatascience.com/similarity-search-part-4-hierarchical-navigable-small-world-hnsw-2aad4fe87d37

Langkah 1: Konstruksi Graf



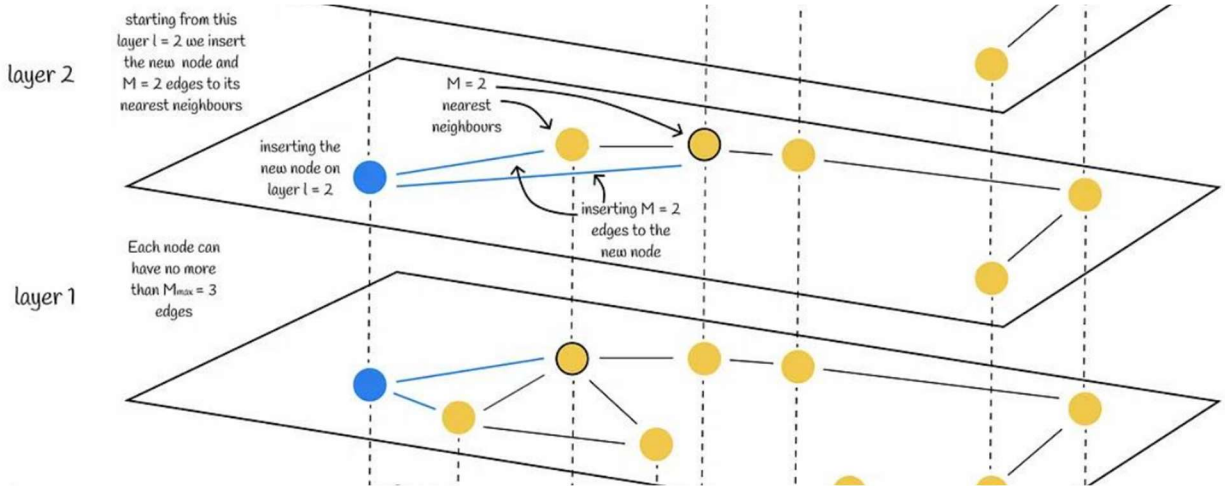
1. Vektor-vektor dimasukkan ke dalam graf secara berurutan, satu per satu.
2. Graf terstruktur dalam beberapa layer, dengan jumlah layer ditentukan oleh parameter **L**.
3. Penyisipan setiap vektor ke layer tertentu diatur oleh fungsi probabilitas yang menurun secara eksponensial, yang dinormalisasi oleh 'pengali level' **m_L**.
4. Fungsi probabilitas ini menentukan apakah sebuah vektor akan menjadi bagian dari suatu layer, dengan layer yang lebih tinggi memiliki lebih sedikit node.
5. Fungsi probabilitas diulang untuk setiap layer, kecuali untuk layer 0, yang mencakup semua vektor.
6. Setiap vektor ditambahkan ke layer sisipan yang ditentukan dan setiap layer di bawahnya.

Langkah 2: Penyisipan Node



1. Ketika sebuah node (vektor) baru disisipkan, ia diberikan nomor layer berdasarkan fungsi probabilitas yang disebutkan sebelumnya.
2. Proses penyisipan dimulai dari layer teratas dan mencari node terdekat secara greedy.
3. Setelah ditemukan, node ini digunakan sebagai titik masuk ke layer berikutnya, dan pencarian terus berlanjut ke bawah.
4. Proses ini diulang sampai layer sisipan **l** tercapai.

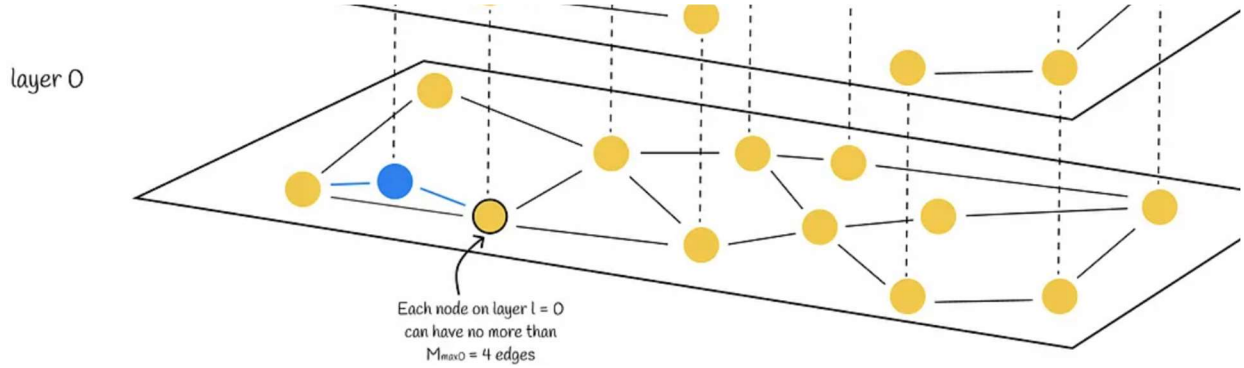
Langkah 3: Pencarian Tetangga dan Pembuatan Edge



1. Mulai dari layer **l**, algoritma menyisipkan node baru di layer saat ini.
2. Kemudian algoritma mencari sejumlah tetangga terdekat, yang ditentukan oleh sebuah hyperparameter (**efConstruction**).
3. Untuk setiap tetangga yang ditemukan, edge dibuat yang menghubungkan node baru dengan tetangga-tetangga ini.
4. Algoritma kemudian turun ke layer berikutnya, menggunakan node yang baru terhubung sebagai titik masuk untuk pencarian tetangga lebih lanjut.

5. Proses ini terus berlanjut sampai layer terbawah (layer 0) tercapai, di mana insertingg dan pembangunan edge terakhir selesai.

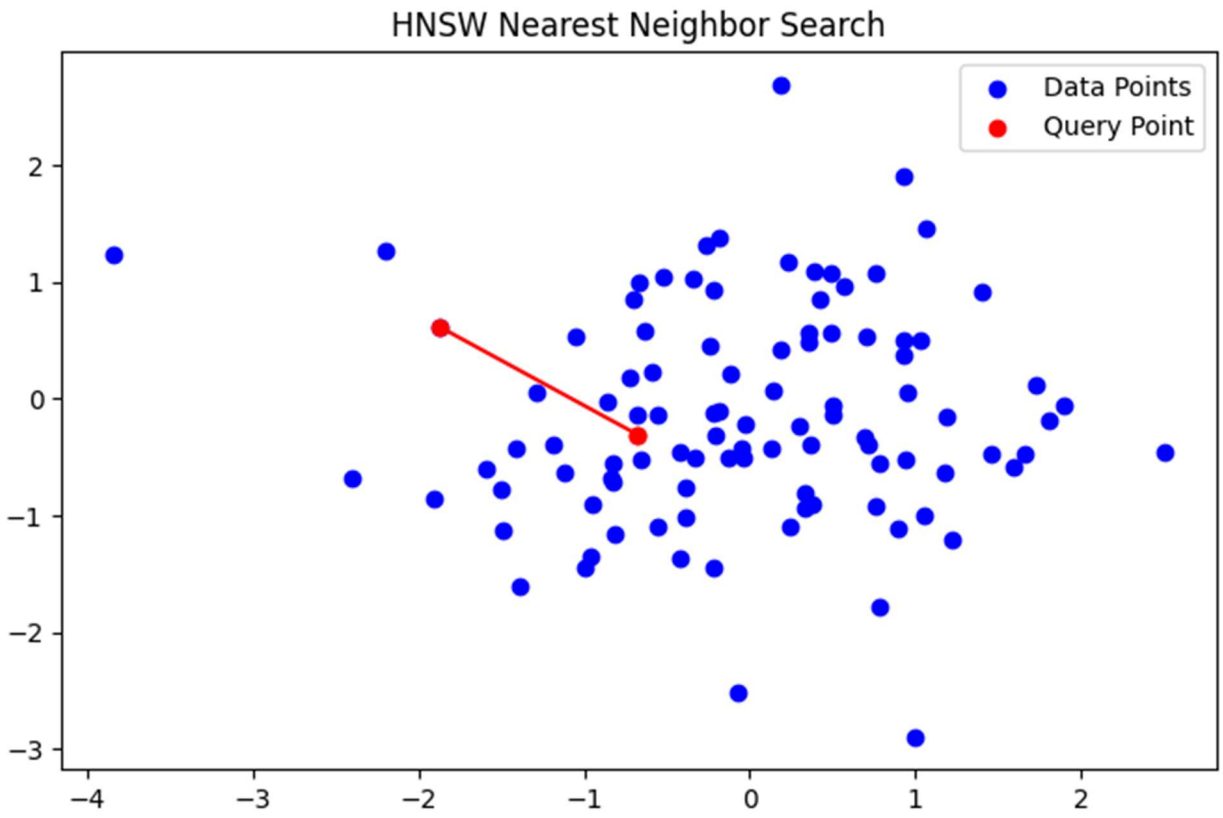
Menyelesaikan Graf HNSW



- Setiap node di layer 0 memiliki jumlah maksimum edge yang tetap, ditentukan oleh parameter (mungkin **M** atau **Max**).
- Algoritma memastikan bahwa setiap node yang di-insert terhubung dengan tetangga terdekatnya sesuai dengan parameter yang ditetapkan untuk graf tersebut.
- Proses konstruksi memastikan bahwa node di layer yang lebih tinggi berfungsi sebagai jalan pintas untuk mencapai node di layer yang lebih rendah dengan lebih cepat selama kueri pencarian.

Berikut implementasi pembuatan HNSW Graf saya:

https://colab.research.google.com/drive/1zjJrAe_rCOFjXJxha-QNGFWILezNjz1T#scrollTo=STR__EMLBqdS



Berikut ini adalah hasil visualisasi akhirnya, terlihat bahwa pencarian tetangga terdekat dengan menggunakan algoritma HNSW telah menghasilkan suatu titik yang secara geografis dekat dengan titik kueri pada ruang dua dimensi. Titik kueri, yang diwakili oleh warna merah, dihubungkan dengan garis merah ke titik data terdekat dalam himpunan data yang ditandai dengan warna biru. Namun, karena HNSW merupakan algoritma yang berorientasi pada efisiensi pencarian dalam ruang berdimensi tinggi dan oleh karena itu mengutamakan kecepatan dan efisiensi di atas solusi yang global optima. Oleh karena itu, solusi yang ditemukan oleh HNSW cenderung menjadi solusi aproksimasi dari tetangga terdekat dan tidak selalu merupakan yang terdekat secara global. Jadi hasil akhir pada visualisasi di atas belum tentu merupakan solusi yang global optima