

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Solusi

##### 2.1.1 HATS: A Hierarchical Graph Attention Network for Stock Movement Prediction

Relasi antar perusahaan dapat mempengaruhi pergerakan harga saham perusahaan tersebut dan hal ini dapat dimanfaatkan untuk penggunaan *deep learning* berbasis GNN (Graph Neural Network). Penelitian yang dilakukan oleh Kim et al. (2019), memperkenalkan sebuah model berbasis hierarki yang memanfaatkan informasi hubungan antar perusahaan. Model berbasis hierarki tersebut diberi nama HATS (*Hierarchical Attention Network for Stock Prediction*) dan menggunakan model GAT sebagai dasar model [14].

Penelitian memiliki beberapa tujuan yaitu untuk mencari karakteristik relasi antar simpul yang cocok untuk prediksi harga saham, menguji HATS yang dapat menyatukan informasi pada beberapa relasi yang berbeda. Data yang digunakan dalam penelitian ini yaitu indeks S&P 500. Data indeks tersebut akan menjadi graf dan setiap perusahaan di dalamnya akan menjadi simpul.

Hasil dari penelitian didapatkan bahwa:

- a. Menggunakan data relasi antar saham dalam graf dapat membantu prediksi, namun tidak semua relasi meningkatkan akurasi karena beberapa dapat menimbulkan *noise*.
- b. Pada penelitian ini dapat diketahui bahwa jumlah relasi pada simpul tidak menjamin performa pada model, karena simpul tersebut dapat memiliki *noise*.
- c. Model GNN berbasis *attention* (seperti HATS) unggul dibandingkan GCN dan TGC, karena dapat menekankan relasi yang lebih penting secara adaptif.
- d. Performa HATS dapat lebih bagus, karena HATS dapat mengambil informasi yang berguna pada situasi pasar.

Hasil tersebut menunjukkan bahwa mekanisme attention pada GAT dapat memfilter relasi yang tidak relevan, dan hal ini menjadi alasan pemilihan GAT dalam penelitian ini. Selain itu, hasil penelitian tersebut memperkuat gagasan bahwa struktur relasi (seperti induk dan anak perusahaan) dapat mempengaruhi prediksi harga saham.

### 2.1.2 Stock Price Prediction Using Graph Neural Networks

Penelitian yang dilakukan Meher et al (2023). ini bertujuan untuk memprediksi harga saham menggunakan pendekatan GNN (Graph Neural Network) [11]. Penelitian ini menggunakan data saham S&P 500 selama 5 tahun. Untuk mengubah data tersebut menjadi graf, peneliti terlebih dahulu mengubah data tersebut menjadi MST (*Minimum Spanning Tree*). Proses pembuatan MST dilakukan melalui tahapan: perhitungan log-return, korelasi Pearson antar saham, konversi menjadi metrik jarak, dan konstruksi MST dari relasi terkuat

Setelah graf dibentuk, graf tersebut akan digunakan untuk melatih model GCN. Hasil prediksi yang dikeluarkan oleh model memiliki kemiripan dalam pergerakan harga saham, sehingga model ini dapat menangkap pergerakan harga saham cukup baik. Pada akurasi prediksi terdapat beberapa nilai yang lebih besar dan kebalikannya hal ini disebabkan model tidak terlalu dapat menangkap pergerakan harga yang cukup ekstrim. Hasil dari penelitian didapatkan bahwa:

- a. MST dapat membangun graf dengan relasi antar saham yang paling kuat, sehingga mengurangi noise.
- b. GCN dapat menangkap pergerakan harga saham dengan cukup baik, namun susah menangkap pergerakan yang cukup ekstrim.

Penelitian ini menjadi motivasi penggunaan MST sebagai salah satu graf yang dipake dalam penelitian ini. Selain itu, metode pembuatan graf MST juga didasarkan oleh penelitian ini.

### 2.1.3 Review on Graph Construction and Graph Learning in Stock Price Prediction

Penelitian yang dilakukan oleh Wang et al (2022). ini membahas tentang metode-metode yang dapat digunakan untuk membuat graf saat melakukan prediksi harga menggunakan graph neural network [15]. Dalam pembuatan graf diperlukan sebuah simpul dan atribut di dalam simpul tersebut. Saham dapat digunakan sebagai simpul, yang berisikan atribut seperti data OHLCV (*Opening, High, Close, Low, Volume*), MA (*Moving Average*), RSI (*Relative Strength Index*) dan lainnya.

Pada pembuatan relasi antara saham, terdapat beberapa metode yang dapat digunakan. Metode pertama yaitu membuat relasi antara saham berdasarkan pengetahuan manusia. Pembangunan relasi ini dilakukan secara manual dan terdapat dua jenis pendekatan. Pendekatan pertama yaitu dengan membangun relasi antara saham berdasarkan kemiripan perusahaan tersebut, contoh perusahaan A dan perusahaan B sama-sama bergerak di bidang yang sama. Pendekatan kedua yaitu melalui koneksi antara perusahaan, seperti hubungan perusahaan induk dan perusahaan anak.

Metode kedua yaitu menggunakan kalkulasi untuk mencari kemiripan antara saham. Cara ini dapat dilakukan dengan menghitung salah satu nilai pada saham untuk mendapatkan korelasi. Setelah nilai korelasi didapatkan, relasi akan dibuat berdasarkan batas nilai tertentu. Untuk mendapatkan nilai korelasi didapatkan dengan berbagai cara, seperti spearman rank-order correlation, pearson correlation, dan lainnya.

Metode ketiga yaitu menggunakan *knowledge graph* untuk mendapatkan relasi. Pembuatan graf ini dilakukan dengan mengekstrak informasi dari suatu entitas, relasi, atribut dan peristiwa. Contoh mengekstrak relasi suatu saham berdasarkan informasi yang ada di berita, atau informasi yang ada pada laporan keuangan perusahaan. Dari penelitian ini diketahui bahwa terdapat banyak cara untuk membuat graf untuk memprediksi harga pergerakan saham.

Hasil dari penelitian didapatkan bahwa:

- a. Penggunaan fitur simpul yang beragam dapat memperkaya informasi yang dikirim antar simpul.
- b. Penggunaan knowledge graph bisa memberikan graf yang akurat tetapi sangat kompleks dalam implementasinya.

Penelitian ini menjadi dasar peneliti untuk mengadopsi dua pendekatan konstruksi graf: MST sebagai graf berbasis kalkulasi kemiripan, dan graf berbasis relasi pengetahuan seperti kepemilikan saham dan sektor. Hal tersebut dipilih mempertimbangkan dengan data yang tersedia dan kompleksitas sistem.

## 2.2 Tinjauan Teori

### 2.2.1 Saham

Saham merupakan tanda seseorang atau entitas lainnya yang ikut memberikan modal kepada suatu perusahaan. Entitas yang menyertakan modal tersebut maka memiliki perusahaan tersebut senilai dengan jumlah saham yang didapatkan. Perusahaan menerbitkan saham untuk mendapatkan pendanaan lebih luas.

Terdapat 2 keuntungan yang bisa didapatkan investor saham. Keuntungan pertama yaitu dividen atau pembagian keuntungan yang diberikan perusahaan berdasarkan hasil keuntungan yang didapatkan perusahaan. Namun untuk mendapatkan dividen, investor harus memiliki saham perusahaan dalam rentang waktu yang ditentukan perusahaan, sehingga tidak semua investor kebagian dividen. Keuntungan kedua didapatkan dari *capital gain* yang merupakan selisih antara nilai harga beli dan jual pada saham [16].

### 2.2.2 MNC Group

MNC Group merupakan sebuah perusahaan investasi dan media di Indonesia. MNC group didirikan pada tahun 1989 dan memiliki 4 sektor bisnis, yaitu media dan hiburan, layanan keuangan, perhotelan/layanan kesehatan, dan energi. Di sektor media dan hiburan, MNC memiliki saluran

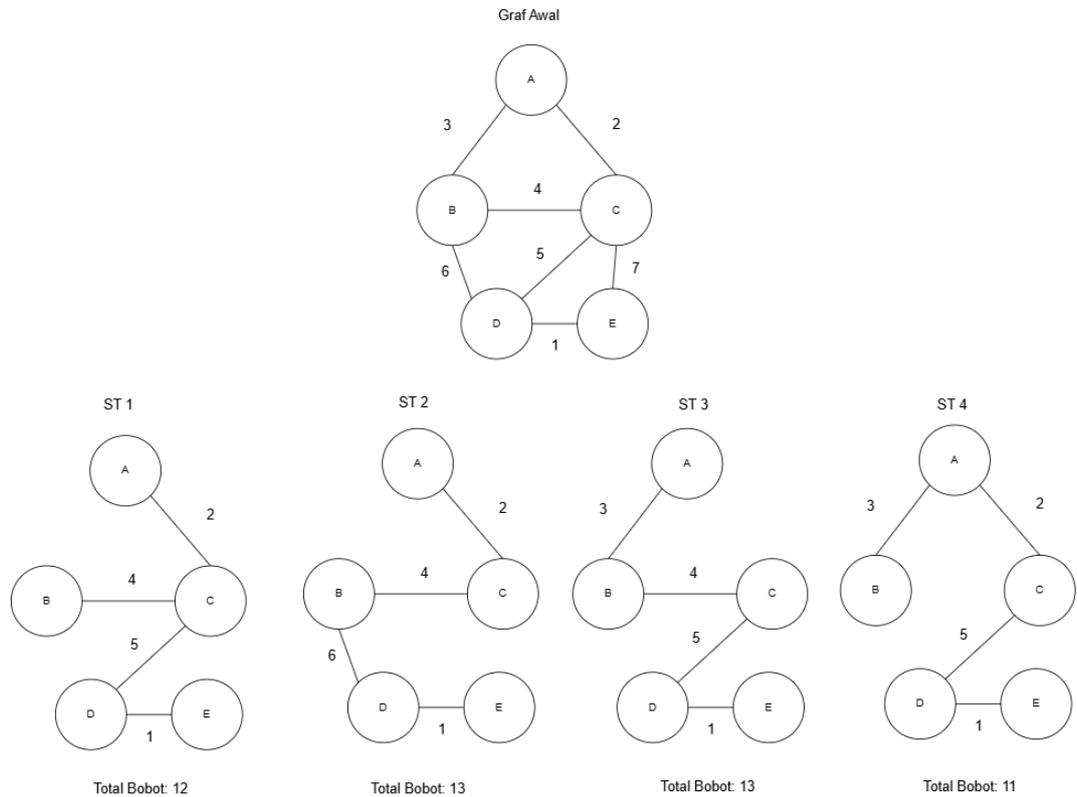
TV nasional yaitu RCTI, MNC TV, dan GTV. Di sektor keuangan, melalui PT MNC Kapital Indonesia Tbk, terdapat layanan perbankan bernama MNC Bank. Di sektor perhotelan/layanan kesehatan MNC mengelola beberapa properti hotel mewah seperti Park Hyatt Jakarta dan The Westin Resort Nusa Dua Bali. Pada sektor energi, MNC Group memiliki MNC Energy Investments yang bergerak di bidang pertambangan batubara di Sumatera Selatan [17].

### 2.2.3 *Minimum Spanning Tree*

ST (*Spanning Tree*) merupakan sebuah subgraf berbentuk seperti pohon dari sebuah graf terhubung yang tidak memiliki arah dan mencakup semua simpul dari graf tersebut. Terdapat kriteria untuk membentuk ST sebagai berikut:

- a. Jumlah simpul pada graf sama dengan ST (*Spanning Tree*).
- b. Jumlah sisi pada ST harus kurang dari 1 jumlah simpul.
- c. Setiap simpul harus terhubung ke salah satu simpul.
- d. Tidak ada sisi yang membentuk siklus antara simpul.
- e. Total bobot ST didapatkan dengan menghitung total setiap bobot sisi.
- f. ST memiliki banyak kemungkinan bentuk.





Gambar 2.1 Contoh Graf *Spanning Tree*

- Graf Awal: Graf yang terdiri dari 5 simpul yang setiap simpulnya memiliki koneksi. Graf ini bukan ST karena memiliki siklus (A-B-C-A), dan jumlah sisi lebih dari jumlah simpul – 1.
- ST 1 - 3: ST pertama yang terbentuk dari graf awal, jumlah sisi adalah  $4(n-1)$ , dan tidak ada siklus.
- ST 4: ST dengan bobot terendah sehingga ST 4 merupakan MST.

MST (*Minimum Spanning Tree*) merupakan ST yang memiliki jumlah bobot paling kecil dibandingkan ST lainnya [18].

#### 2.2.4 Graf

Graf adalah bagian dari matematika yang mempelajari bagaimana jaringan dapat direpresentasikan dan dianalisis berdasarkan struktur dan informasi yang ada didalam struktur tersebut. Graf terdiri dari simpul (*node*) yang mewakili suatu entitas, dan sisi (*edge*) yang merepresentasikan hubungan antara simpul.

Terdapat berbagai jenis graf, salah satunya adalah graf sederhana, dimana setiap simpul hanya satu sisi saja antar simpulnya seperti pada jalan tunggal. Kemudian terdapat multigraf yang memiliki lebih dari satu sisi antar simpulnya. Hubungan antar graf sangat dipengaruhi oleh sisi yang menghubungkan simpul. Graf juga dapat dikategorikan berdasarkan orientasi arah pada sisi, yaitu graf berarah dan graf tidak berarah, dan arah tersebut dapat memiliki bobot maupun tidak memiliki bobot [19].

### 2.2.5 Deep learning

*Deep learning* adalah proses komputasi yang bekerja berdasarkan data input dan dapat beradaptasi dengan arsitektur dari sistem agar memberikan hasil yang diinginkan. Proses yang paling penting dalam *deep learning* yaitu pelatihan model. Pelatihan ini merupakan proses adaptasi yang dilakukan dengan memberikan sampel data input dan hasil yang diinginkan. Dengan adanya pelatihan, algoritma secara optimal akan mengkonfigurasi dirinya sendiri agar dapat memberikan hasil yang lebih baik dan dapat menggeneralisasi untuk memberikan hasil dari data berdasarkan input yang tidak pernah dilatih.

Secara arsitektur *deep learning* memiliki tiga lapisan, yaitu lapisan input, lapisan tersembunyi dan lapisan output. Didalam setiap lapisan tersebut terdapat sebuah *node* yang tersambung antar diantara ketiga lapisan. Terdapat 3 jenis *deep learning* yaitu, *supervised learning*, *unsupervised learning*, dan *semi-supervised learning*.

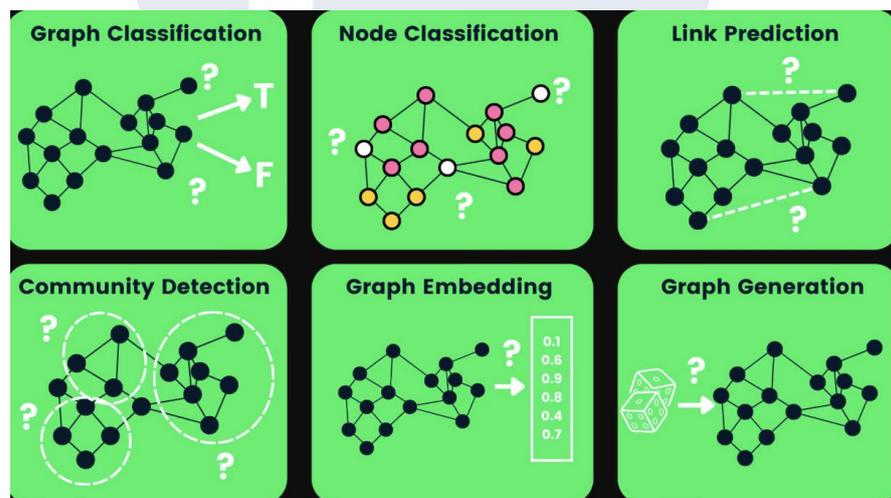
*Supervised learning* adalah proses pembelajaran yang digunakan untuk memperkirakan output yang tidak diketahui labelnya, berdasarkan sampel pelatihan yang memiliki label. *Unsupervised learning* adalah proses pembelajaran yang input dan output tidak diketahui labelnya. *Semi-supervised learning* adalah proses pembelajaran yang tidak semua datanya diberi label [20].

### 2.2.6 Graph Neural Network

Graph Neural Network (GNN) merupakan arsitektur dalam *deep learning* yang dirancang untuk menganalisis data dalam bentuk graf. GNN

berfungsi dengan mengambil informasi dari simpul-simpul tetangga untuk setiap simpul target. Proses ini dilakukan secara iteratif, dimana setiap iterasi mengambil fitur simpul berdasarkan lingkungan sekitarnya dan memperbarui fitur simpul tersebut. Dengan adanya proses tersebut, GNN dapat menangkap pola spasial dan hubungan yang lebih kompleks antara data.

GNN memiliki beberapa tipe seperti GCN (Graph Convolutional Networks) yang menggunakan operasi konvolusi untuk mengagregasi informasi dari tetangga, GAT (Graph Attention Networks) menambahkan mekanisme *attention* untuk mementingkan pentingnya hubungan antar simpul, GRN (Graph Recurrent Networks) Memanfaatkan arsitektur *recurrent* untuk menangkap hubungan temporal dalam graf dinamis, dan masih banyak lagi. GNN dapat diaplikasikan ke beberapa hal seperti network, prediksi, clustering, dan visualisasi [21].



Gambar 2.2 Ilustrasi Pengaplikasian GNN [22]

### 2.2.7 Graph Attention Network

GAT (Graph Attention Network) adalah arsitektur berbasis graf neural network yang dirancang untuk memiliki fleksibilitas dan energi yang kecil dalam memproses data graf. GAT diperkenalkan oleh Velickovic et al. untuk mengatasi permasalahan pada model berbasis graf sebelumnya. Masalah yang diselesaikan yaitu komputasi pada GCN (Graph

Convolutional Network), yang sangat memerlukan operasi matriks yang cukup berat, salah satunya penggunaan dekomposisi eigen.

Untuk mengatasi masalah tersebut digunakan mekanisme *attention*, yang memberikan bobot berbeda pada tetangga simpul (*node*) dalam graf berdasarkan relevansi mereka terhadap simpul target. Dengan sistem tersebut, memungkinkan model untuk meningkatkan hubungan dari setiap graf lebih dinamis. Graf juga memiliki keunggulan untuk bekerja dengan graf yang tidak terstruktur dan tidak homogen, di mana hubungan antar simpul dapat memiliki tingkat kepentingan yang bervariasi [11].

#### 2.7.1.1. Arsitektur Graph Neural Network

Lapisan *graph attention layer* berfungsi sebagai inti dari arsitektur GAT. Lapisan ini akan menerima input berupa simpul yang berisikan fitur awal simpul  $h_i \in \mathbb{R}^F$ , di mana  $F$  berisi dimensi fitur pada setiap simpul. Setiap fitur simpul  $h_i$  kemudian ditransformasikan secara linear dengan matrix bobot  $W \in \mathbb{R}^{F' \times F}$  menghasilkan fitur baru  $Wh_i$ . Tujuan dari transformasi ini untuk memproyeksikan fitur ke dimensi ruang dimensi baru  $F'$ .

Untuk menggunakan mekanisme *self-attention*, dibutuhkan simpul  $i$  dan simpul tetangganya  $j$ . Tahap awal yang dilakukan yaitu menghitung skor perhatian mentah ( $e_{ij}$ ) menggunakan fungsi berikut (1).

$$e_{ij} = a \left( W \begin{matrix} \rightarrow \\ h_i \end{matrix}, W \begin{matrix} \rightarrow \\ h_j \end{matrix} \right) \quad (1)$$

$e_{ij}$  : Attention Mechanism

$a$  : Attention

$W$  : Weight

$h$  : Node Attributes

Dimana  $e_{ij}$  adalah *attention mechanism*,  $a$  adalah *attention*,  $W$  adalah *weight*, dan  $h$  adalah *node attributes*.

Perhitungan  $e_{ij}$  (*attention coefficient*) berfungsi untuk mengetahui seberapa penting simpul  $j$  terhadap simpul  $i$ . Fungsi  $a$  (*attention*) adalah jaringan saraf dengan satu lapisan yang parameternya dinyatakan sebagai  $\vec{a}$  dan menerima konkatenasi  $[Wh_i || Wh_j]$ . sehingga skor *attention* dihitung sebagai berikut (2).

$$e_{ij} = \text{LeakyReLU}_a \left( \vec{a}^T \left[ W \begin{matrix} \rightarrow \\ h_i \end{matrix} \parallel \left[ W \begin{matrix} \rightarrow \\ h_j \end{matrix} \right] \right] \right) \quad (2)$$

Fungsi aktivasi LeakyRelu digunakan dengan slope negatif sebesar 0.2 untuk memberikan non-linearitas, hal tersebut agar gradien tetap mengalir saat nilai negatif.

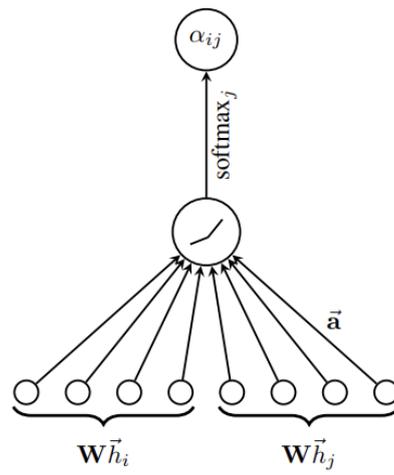
Perhitungan ini perlu dinormalisasi terlebih dahulu. Normalisasi diperlukan karena setiap sisi dapat menghasilkan nilai yang berbeda-beda, dan normalisasi ini dilakukan agar hasilnya dapat berada dalam besaran yang proporsional. Fungsi yang digunakan yaitu softmax dengan perhitungan sebagai berikut (3).

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \quad (3)$$

Dimana  $\alpha_{ij}$  merupakan bobot *attention final* dari simpul  $j$  ke  $i$ ,  $e_{ij}$  merupakan skor *attention* mentah dari hasil LeakyReLU, dan  $N_i$  merupakan tetangga dari simpul  $i$

Berikut ilustrasi dari arsitektur Graph Attention Network pada gambar 2.3.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.3 Ilustrasi Arsitektur GAT [11]

