BAB I
PENDAHULUAN

1.1 Latar Belakang

Investasi merupakan salah satu pilar utama dalam perekonomian modern. Dalam era globalisasi dengan dinamika perekonomian yang terus berubah, pengelolaan sumber daya keuangan menjadi aspek krusial, baik bagi individu maupun organisasi, untuk mencapai tujuan finansial yang optimal. Namun, keterbatasan sumber daya dan tingginya risiko investasi menuntut pendekatan strategis dalam mengalokasikan dana agar dapat memaksimalkan pengembalian tanpa mengabaikan prinsip kehati-hatian (Kalsum et al., 2023).

Optimalisasi alokasi sumber daya keuangan semakin relevan seiring dengan perubahan paradigma dalam dunia ekonomi dan keuangan. Konsep-konsep seperti efisiensi pasar, diversifikasi, dan manajemen risiko menjadi elemen penting dalam pengambilan keputusan finansial yang cerdas. *Modern Portfolio Theory* (MPT) yang diperkenalkan oleh Harry Markowitz pada tahun 1952 memberikan landasan teoretis untuk memahami hubungan antara risiko dan imbal hasil investasi (Kalsum et al., 2023). Melalui konsep *efficient frontier*, MPT membantu investor menyusun portofolio yang optimal, yaitu portofolio dengan imbal hasil tertinggi pada tingkat risiko tertentu atau risiko terendah untuk imbal hasil tertentu. Dengan demikian, MPT memberikan panduan penting bagi investor dalam membentuk portofolio yang sesuai dengan profil risiko mereka. Dalam kondisi pasar yang seimbang, semua investor cenderung memilih portofolio pasar yang berada pada kurva *efficient frontier* (Adnyana, 2020).

Selain teori MPT, *Capital Asset Pricing Model* (CAPM) juga memberikan kontribusi signifikan dalam analisis portofolio. CAPM menghitung hubungan antara risiko sistematis (*beta*) dengan pengembalian yang diharapkan (*expected return*), sehingga membantu investor memahami alokasi dana yang lebih efektif (Alqiha dan Imani, 2021). Meskipun CAPM dan MPT menawarkan banyak manfaat, model ini memiliki keterbatasan. CAPM mengasumsikan distribusi pengembalian aset yang normal serta pasar yang sempurna, padahal kondisi nyata sering kali jauh lebih kompleks. Selain itu, CAPM tidak mampu menangkap risiko non-sistematis yang juga dapat memengaruhi volatilitas pengembalian aset. Asumsi-asumsi ini dapat menghasilkan hasil yang secara matematis akurat tetapi kurang mencerminkan kondisi pasar nyata (Kuncoro, 2023).

Aspek psikologis yang berasal dari perilaku ekonomi memiliki peran signifikan dalam proses pengambilan keputusan investasi. Bias, emosi, dan asimetri informasi seringkali memengaruhi tindakan investor. Oleh karena itu, strategi investasi yang realistis dan berkelanjutan perlu mempertimbangkan faktor-faktor tersebut. Dalam konteks optimasi portofolio, tidak cukup hanya berfokus pada keuntungan yang diharapkan, tetapi juga penting untuk memahami hubungan antar-aset dan risiko yang melekat. Pemahaman ini memungkinkan investor merancang strategi yang lebih adaptif dan tangguh terhadap dinamika pasar (Kuncoro, 2023).

Di Indonesia, indeks saham LQ45 menjadi salah satu acuan utama bagi investor. Indeks ini terdiri atas 45 emiten dengan likuiditas tertinggi dan fundamental perusahaan yang baik. Sebagai tolok ukur, indeks saham LQ45 membantu investor dalam memantau saham yang aktif diperdagangkan. Indeks ini diperbarui setiap enam bulan sekali berdasarkan kriteria tertentu, seperti kapitalisasi pasar dan nilai transaksi tertinggi di Bursa Efek Indonesia (BEI) (Pasaribu, 2024). Namun, saham-saham dalam LQ45 juga menunjukkan volatilitas yang cukup tinggi, yang mencerminkan risiko inheren dalam pasar saham. Hal ini menunjukkan pentingnya analisis volatilitas dalam membantu investor memahami risiko yang melekat pada aset-aset tersebut (Sumarga et al., 2023).

Seiring perkembangan teknologi, pendekatan berbasis *machine learning* mulai digunakan untuk analisis pasar saham, termasuk memprediksi harga, mengidentifikasi sen*time*n pasar, hingga merancang strategi investasi yang lebih canggih. Salah satu metode yang sering digunakan adalah *Long Short-Term Memory* (LSTM), algoritma dari *Recurrent Neural Network* (RNN) yang memiliki keakuratan tinggi untuk analisis data *time series*. LSTM mampu menangkap ketergantungan temporal dan hubungan *non-linear* dalam data harga saham, sehingga menghasilkan prediksi yang mendekati harga aktual. Pipin et al. (2023) melakukan penelitian tentang efektivitas LSTM dalam memprediksi data harga saham. Penelitian tersebut menemukan bahwa LSTM menghasilkan nilai *Mean Squared Error* (MSE) yang sangat rendah, yaitu sebesar 0,0109012, serta akurasi prediksi yang tinggi dengan *Mean Percentage Error* (MPE) sebesar 1,74%. Selanjutnya, Budiprasetyo et al. (2023) melakukan penelitian yang merujuk pada kajian sebelumnya oleh Siami-Namini et al. (2019), yang membandingkan metode ARIMA dan LSTM dalam memprediksi data *time series*. Dari hasil penelitian tersebut dapat disimpulkan bahwa LSTM memberikan performa yang lebih baik, dengan rata-rata peningkatan akurasi prediksi hingga 85%. Penelitian oleh Kwanda et al. (2024) membandingkan LSTM dan BiLSTM dalam memprediksi data harga saham, dan hasilnya menunjukkan bahwa LSTM unggul meskipun perbedaan performanya tidak signifikan. Anandita dan Wahyuningsih (2024) melakukan studi pada indeks saham syariah JII70. Mereka menyimpulkan bahwa LSTM mampu memberikan akurasi tinggi dengan nilai *Root Mean Squared Error* (RMSE) yang rendah, menegaskan keunggulan algoritma ini dalam memproses berbagai jenis data pasar saham.

Volatilitas, sebagai ukuran fluktuasi harga dalam periode tertentu, digunakan untuk memperkirakan risiko dan mendukung keputusan investasi (Jumiati et al., 2024). Pada indeks LQ45, volatilitas menjadi aspek penting dalam pengelolaan risiko investasi, yang dapat diukur melalui varians saham. Penelitian yang dilakukan oleh Aprilia et al. (2021) menyimpulkan bahwa analisis sifat varians, baik yang bersifat konstan maupun tidak, dapat dilakukan melalui pengujian heteroskedastisitas. Jika data saham menunjukkan sifat homoskedastisitas, maka perhitungan volatilitas dapat dilakukan menggunakan metode sederhana seperti *Value at Risk* (VaR) atau standar deviasi. Namun, apabila data bersifat heteroskedastisitas, pendekatan berbeda seperti metode *Exponentially Weighted Moving Average* (EMWA) diperlukan. Pada penelitian ini, untuk data yang bersifat heteroskedastisitas, akan digunakan model *Generalized Autoregressive Conditional Heteroscedasticity* (GARCH). Model GARCH sering digunakan untuk menganalisis volatilitas saham dengan varians yang tidak konstan, sehingga menjadi alat yang efektif untuk memahami fluktuasi harga yang tajam dan memberikan panduan penting bagi investor dalam mengelola portofolio secara lebih strategis (Ermanely et al., 2023).

Penelitian sebelumnya telah banyak membahas optimasi portofolio menggunakan pendekatan konvensional seperti CAPM dan MPT. Sementara itu, penelitian terkait *machine learning* sebagian besar hanya berfokus pada aspek prediksi, peramalan harga saham, atau evaluasi model secara individu, tanpa mengintegrasikan hasilnya ke dalam analisis portofolio secara komprehensif. Namun, kajian yang secara langsung membandingkan efektivitas pendekatan konvensional seperti CAPM dengan metode berbasis *machine learning*, khususnya LSTM-GARCH, dalam konteks Indeks LQ45 masih sangat terbatas. Oleh karena itu, penelitian ini menawarkan kebaruan dengan mengeksplorasi dan mengevaluasi kinerja kedua pendekatan tersebut secara langsung, khususnya dalam membantu investor mengelola portofolio saham secara lebih efektif dan efisien di pasar modal Indonesia.

Berdasarkan uraian di atas, peneliti tertarik untuk melakukan penelitian berjudul "**Analisis Perbandingan Optimasi Portofolio Menggunakan Pendekatan Konvensional CAPM dan Pendekatan *Machine learning* LSTM-GARCH pada Indeks LQ45**" . Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teori optimasi portofolio dan memberikan wawasan praktis bagi investor dalam merancang strategi investasi yang lebih efektif dan efisien.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana menentukan portofolio saham optimal dengan menggunakan pendekatan berbasis *machine learning* LSTM-GARCH?
2. Bagaimana menentukan portofolio saham optimal dengan menggunakan pendekatan konvensional *Capital Asset Pricing Model* (CAPM)?
3. Bagaimana perbandingan kinerja portofolio yang diperoleh dari pendekatan *machine learning* LSTM-GARCH dan konvensional CAPM?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis pengoptimalan portofolio saham menggunakan pendekatan berbasis *machine learning* LSTM-GARCH.
2. Menganalisis pengoptimalan portofolio saham menggunakan pendekatan konvensional *Capital Asset Pricing Model* (CAPM).
3. Membandingkan kinerja portofolio saham yang diperoleh dari pendekatan *machine learning* LSTM-GARCH dan konvensional CAPM dalam mengoptimalkan portofolio saham.

1.3.2 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis, penelitian ini diharapkan dapat menambah pemahaman mendalam mengenai pendekatan *machine learning* LSTM-GARCH serta pendekatan konvensional CAPM, khususnya dalam konteks optimasi portofolio saham.

2. Bagi Pembaca, penelitian ini dapat memberikan wawasan dan referensi mengenai pendekatan alternatif untuk analisis risiko, *return*, dan optimasi portofolio saham.
3. Bagi Universitas, penelitian ini dapat mendorong pengembangan ilmu di bidang kecerdasan buatan dan keuangan, khususnya terkait implementasi *machine learning* dalam optimasi portofolio.
4. Bagi Investor, hasil penelitian ini dapat menjadi acuan dalam memilih metode analisis yang lebih optimal untuk mendukung pengambilan keputusan investasi, khususnya pada saham-saham yang terdaftar di indeks LQ45.

1.4 Landasan Teori

1.4.1 Pasar Modal

Pasar modal adalah tempat bagi perusahaan atau pihak lain untuk menjual saham dan obligasi guna mendapatkan dana tambahan atau memperkuat keuangan. Di sini, pihak yang memiliki dana berlebih bertemu dengan pihak yang membutuhkan melalui perdagangan sekuritas seperti saham dan obligasi, yang umumnya berjangka waktu lebih dari satu tahun (Fabozzi, 1995). Bursa efek, sebagai pusat pasar modal, memfasilitasi transaksi sekuritas di pasar sekunder setelah sekuritas dijual di pasar perdana. Dengan mendaftarkan sekuritas di bursa efek, perdagangan menjadi lebih terbuka dan terorganisir, menjadikannya inti dari aktivitas pasar modal (Wefi, 2020).

1.4.2 Investasi

Investasi adalah aktivitas menempatkan dana dengan harapan mendapatkan keuntungan atau peningkatan nilai di masa depan. Ini melibatkan alokasi uang saat ini untuk hasil yang lebih besar di kemudian hari. Dalam konteks saham, investasi dilakukan dengan membeli saham di pasar modal, berharap keuntungan dari kenaikan harga saham di masa depan. Secara umum, investasi dibagi menjadi dua jenis aset yaitu aset riil, seperti emas, properti, atau karya seni, dan aset finansial, seperti saham, obligasi, deposito, atau reksadana. Investasi pada aset finansial dapat dilakukan secara langsung dengan membeli aset keuangan dari perusahaan tertentu, atau secara tidak langsung melalui perusahaan investasi yang mengelola portofolio berbagai aset. Kedua metode ini memungkinkan investor untuk memilih sesuai dengan tujuan dan toleransi risiko mereka (Adnyana, 2020).

1.4.3 Saham

Saham adalah bukti kepemilikan individu atau badan atas suatu perusahaan, yang memberikan hak atas aset dan keuntungan perusahaan. Saham biasanya berbentuk dokumen yang mencantumkan nama perusahaan, nilai nominal, serta hak dan kewajiban pemegangnya. Dengan membeli saham, seseorang secara teknis menjadi salah satu pemilik perusahaan (Todingan et al., 2023). Sebagai instrumen investasi, saham menawarkan potensi keuntungan tinggi tetapi juga memiliki risiko besar karena fluktuasi harga yang cepat. Harga saham dipengaruhi oleh penawaran, permintaan, kondisi ekonomi, performa perusahaan, dan psikologi pasar. Pergerakannya bersifat *non-linear*, *non-stationary*, dan memiliki ketergantungan pada pola masa lalu (*long* *memory dependency*). Meskipun risikonya besar, saham tetap menarik karena dianggap mampu memberikan *return* tertinggi dibandingkan instrumen investasi lainnya. Namun, memahami faktor-faktor yang memengaruhi pergerakan harga saham sangat penting untuk mengelola risiko dan memaksimalkan keuntungan.

1.4.4 *Return*

Return adalah keuntungan atau kerugian yang diperoleh oleh perusahaan, individu, atau lembaga sebagai hasil dari keputusan investasi yang mereka buat. Secara umum, tujuan utama seseorang berinvestasi adalah untuk mendapatkan keuntungan di masa depan yang dapat membantu meningkatkan kualitas hidup mereka. Imbal hasil atau *return* ini dapat diartikan sebagai bentuk penghargaan atau kompensasi yang diterima dari aktivitas investasi yang dilakukan.

Return saham merupakan salah satu daya tarik utama bagi investor dalam melakukan investasi. Oleh karena itu, penting bagi investor untuk memahami dua jenis *return* yang menjadi acuan keuntungan, yaitu *realized return* dan *expected return*. Menurut Hartono (2017), *realized return* adalah keuntungan yang sudah terealisasi, dihitung berdasarkan data historis, serta berfungsi sebagai dasar untuk memperkirakan *expected return* dan risiko di masa depan. Sedangkan *expected return* adalah keuntungan yang diharapkan oleh investor di masa depan. Pendapat serupa dikemukakan oleh Tandelilin (2017), yang menyatakan bahwa *return* dapat dibagi menjadi *expected return* yaitu *return* yang diantisipasi pada masa mendatang, dan *realized return* yaitu *return* yang sudah terjadi di masa lalu.

1.4.4.1 Tingkat Pengembalian Saham Individual (*Return* Individual)

Tingkat pengembalian saham individu merujuk pada total keuntungan atau kerugian yang diperoleh dari investasi saham dalam periode waktu tertentu. Nilai *return* bisa bernilai positif atau negatif, bergantung pada kondisi aktual dari aset yang diinvestasikan. Penghitungan *return* saham dapat dilakukan dengan menggunakan persamaan berikut:

$$R_{i(t)} = \frac{P_{i(t)} - P_{i(t-1)}}{P_{i(t-1)}} \tag{1}$$

Dengan:

$R_{i(t)}$ = *Return* saham *i* periode ke-*t*
$P_{i(t)}$ = Harga penutupan saham *i* periode ke-*t*
$P_{i(t-1)}$ = Harga penutupan saham *i* periode ke-(*t-1*)

Expected return dari saham individu adalah perkiraan keuntungan yang diharapkan atau dijanjikan akan diperoleh investor dari suatu saham. Persamaan untuk menghitung *expected return* saham individual adalah sebagai berikut:

$$E[R_i] = \frac{\sum_{t=1}^{n} R_{i(t)}}{n} \tag{2}$$

Dengan:
$E[R_i]$ = *Expected return* saham *i*
n = Banyaknya periode pengamatan

1.4.4.2 Tingkat Pengembalian Bebas Risiko (*Risk-free rate*)

Dalam menggunakan *Capital Asset Pricing Model* (CAPM) untuk keputusan investasi, penting untuk mengetahui nilai *risk-free rate* (R_f), yang menunjukkan tingkat pengembalian dari investasi tanpa risiko. Hal ini mengindikasikan bahwa investor tidak akan menerima risiko tambahan jika investasi tidak menawarkan pengembalian lebih tinggi dibandingkan dengan aset bebas risiko. Di Indonesia, *risk-free rate* sering dikaitkan dengan tingkat bunga Sertifikat Bank Indonesia (SBI), yang dijamin oleh pemerintah (Ekananda, 2019). Persamaan yang dapat digunakan untuk menghitung tingkat pengembalian bebas risiko adalah sebagai berikut (Silitonga dan Aziz, 2023):

$$R_f = \frac{\sum_{t=1}^{n} R_{f(t)}}{n} \tag{3}$$

Dengan:
R_f = Rata-rata r*isk-fee rate* atau tingkat pengembailan bebas risiko

1.4.4.3 Tingkat Pengembalian Pasar (*Return Market*)

Indeks pasar saham adalah indikator penting yang mencerminkan pergerakan harga saham di pasar modal. Perubahan indeks pasar menunjukkan kondisi umum saham yaitu pelemahan indeks mengindikasikan saham sedang turun, sementara kenaikan indeks menunjukkan saham sedang naik. *Return market* adalah *return* yang diharapkan dari fluktuasi indeks harga saham pasar atau portofolio pasar. Handini dan Astawinetu (2020) berpendapat bahwa *return market* yang mencerminkan kinerja pasar saham sering dirangkum dalam indeks pasar saham, yang menggambarkan pergerakan harga-harga saham. Secara keseluruhan, *return market* (R_m) menunjukkan pengembalian yang berasal dari perkembangan indeks pasar saham dan memberi gambaran kepada investor mengenai kinerja saham-saham yang terdaftar di pasar modal (Martalena dan Malinda, 2019).

Salah satu indeks pasar saham yang ada di Indonesia adalah Indeks LQ45. LQ45 yang merupakan singkatan dari *Liquid* 45, adalah indeks yang terdiri dari 45 saham pilihan di Bursa Efek Indonesia (BEI) yang memiliki likuiditas tinggi dan kapitalisasi pasar besar. Saham-saham yang masuk dalam indeks ini dipilih berdasarkan kriteria tertentu, seperti volume perdagangan dan kapitalisasi pasar, sehingga mencerminkan kinerja pasar saham Indonesia secara umum. LQ45 sering digunakan sebagai acuan bagi investor untuk menganalisis kondisi pasar dan memprediksi *return* saham. Dikarenakan pada penelitian ini dilakukan pada saham yang termasuk ke dalam LQ45, maka *return market* untuk waktu ke-*t* dapat dihitung menggunakan persamaan berikut:

$$R_{m(t)} = \frac{LQ45_{(t)} - LQ45_{(t-1)}}{LQ45_{(t-1)}} \tag{4}$$

Dengan:

$R_{m(t)}$ = *Return market* periode ke-*t*
$LQ45_{(t)}$ = Indeks LQ45 periode ke-*t*
$LQ45_{(t-1)}$ = Indeks LQ45 periode ke-(*t-1*)

Expected return market adalah perkiraan keuntungan yang diharapkan atau diprediksi akan diperoleh investor dari pasar saham secara keseluruhan. Adapun persamaan untuk menghitung *expected return pasar* adalah sebagai berikut:

$$E[R_m] = \frac{\sum_{t=1}^{n} R_{m(t)}}{n} \tag{5}$$

Dengan:

$E[R_m]$ = *Expected return market*

1.4.4.4 *Return* Portofolio

Return portofolio merupakan hasil gabungan dari *return* masing-masing saham dalam portofolio, dihitung sebagai rata-rata tertimbang berdasarkan proporsi dana yang dialokasikan ke setiap saham selama periode tertentu. Misalkan investor berencana untuk berinvestasi pada n aset, maka *return* realisasi portofolio dapat dihitung dengan persamaan berikut:

$$R_p = w_1 R_1 + w_2 R_2 + \cdots + w_n R_n = \sum_{i=1}^{n} w_i R_i \tag{6}$$

Sementara *expected return* dari portofolio dapat dihitung dengan persamaan berikut:

$$E[R_p] = w_1 E[R_1] + w_2 E[R_2] + \cdots + w_n E[R_n] = \sum_{i=1}^{n} w_i E[R_i] \tag{7}$$

Dengan:

w_i = Proporsi atau bobot saham *i*
R_p = *Return* portofolio
$E[R_p]$ = *Expected return* portofolio

1.4.5 Risiko

Risiko yang tidak pasti dapat menghambat tercapainya tujuan, termasuk dalam kegiatan investasi. Selain *return*, risiko juga menjadi faktor yang dipertimbangkan oleh investor, sehingga mereka berusaha untuk menghindari dan meminimalkan risiko tersebut. Menurut Martalena dan Malinda (2019), setiap investor menghadapi ketidakpastian hasil investasi, terutama karena investasi biasanya dilakukan dalam jangka panjang. Ketidakpastian ini disebut sebagai risiko, yang merujuk pada kemungkinan hasil yang berbeda dari yang diharapkan. Dengan demikian, risiko dapat dianggap sebagai ketidakpastian yang menghalangi tercapainya tujuan investasi, yang terlihat dari perbedaan antara *realized return* dan *expected return*. Semakin besar perbedaan tersebut, semakin besar risiko yang dihadapi. Umumnya, investor tidak menyukai risiko, sehingga mereka berusaha untuk menghindarinya. Sikap investor terhadap risiko dapat dibagi menjadi tiga jenis yaitu *risk seeker*, *risk neutrality*, dan *risk averter*.

1.4.5.1 Risiko Saham Individual

Variansi digunakan untuk mengukur sejauh mana hasil investasi menyimpang dari nilai rata-rata, dan semakin tinggi variansi, semakin besar risiko yang terkait dengan investasi tersebut (Adnyana, 2020). Risiko saham dapat dihitung dengan persamaan sebagai berikut:

$$\sigma_i^2 = \frac{\sum_{t=1}^{n} \left(R_{i(t)} - E[R_i]\right)^2}{n-1} \tag{8}$$

$$\sigma_i = \sqrt{\frac{\sum_{t=1}^{n} \left(R_{i(t)} - E[R_i]\right)^2}{n-1}} \tag{9}$$

Dengan:

σ_i^2 = Variansi hasil investasi saham *i*

σ_i = Standar deviasi saham *i*

1.4.5.2 Risiko Pasar

Variansi Pasar adalah ukuran yang digunakan untuk menggambarkan seberapa besar fluktuasi harga pasar secara keseluruhan dari nilai rata-ratanya. Variansi pasar mengukur sejauh mana pergerakan harga sekuritas di pasar secara keseluruhan berfluktuasi dari nilai rata-rata pasar. Semakin tinggi variansi pasar, semakin besar fluktuasi harga di pasar, yang menunjukkan adanya risiko pasar yang lebih tinggi. Secara matematis, risiko pasar dapat dihitung dengan persamaan sebagai berikut:

$$\sigma_m^2 = \frac{\sum_{t=1}^{n} \left(R_{m(t)} - E[R_m]\right)^2}{n-1} \tag{10}$$

$$\sigma_m = \sqrt{\frac{\sum_{t=1}^{n} \left(R_{m(t)} - E[R_m]\right)^2}{n-1}} \tag{11}$$

Dengan:

σ_m^2 = Variansi pasar

σ_m = Standar deviasi pasar

1.4.5.3 Risiko Portofolio

Risiko portofolio adalah risiko yang timbul dari kombinasi saham-saham dalam portofolio, yang dihitung berdasarkan standar deviasi atau variansi dari nilai *return* saham yang tergabung di dalam portofolio. Risiko portofolio dapat dihitung dengan persamaan berikut:

$$\sigma_{ij} = \frac{\sum_{t=1}^{n} \left(R_{i(t)} - E[R_i]\right)\left(R_{j(t)} - E[R_j]\right)}{n-1} \tag{12}$$

$$\sigma_p^2 = \boldsymbol{w^T \Sigma w} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \, \sigma_{ij} \tag{13}$$

$$\sigma_p = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \, \sigma_{ij}} \tag{14}$$

Dengan:

σ_p^2 = Variansi portofolio

σ_p = Standar deviasi portofolio (risiko portofolio)

σ_{ij} = Kovariansi antara *return* saham *i* dengan saham *j*

$\boldsymbol{w}$ = Vektor bobot portofolio

$\boldsymbol{\Sigma}$ = Matriks kovariansi saham

1.4.6 *Beta* (*Systematik Risk*)

Beta mengukur hubungan antara *return* suatu sekuritas dengan *return* pasar. Dalam model CAPM, nilai *beta* mencerminkan sensitivitas sekuritas terhadap pergerakan pasar. Semakin besar nilai *beta* ($\beta_i > 1$), semakin tinggi *return* yang diharapkan dari sekuritas tersebut, tetapi risikonya juga meningkat. Sebaliknya, sekuritas dengan nilai *beta* rendah dianggap memiliki risiko yang lebih kecil. Jika *beta* suatu sekuritas sama dengan *beta* pasar ($\beta_i = 1$), maka *return* yang diharapkan dari sekuritas tersebut setara dengan *return* pasar. *Beta* dapat dihitung menggunakan persamaan berikut (Todingan et al., 2023):

$$\sigma_{im} = \frac{\sum_{t=1}^{n} \left(R_{i(t)} - E[R_i]\right)\left(R_{m(t)} - E[R_m]\right)}{n-1} \tag{15}$$

$$\beta_i = \frac{\sigma_{im}}{\sigma_m^2} \tag{16}$$

Dengan:

β_i = *Beta* saham *i*

σ_{im} = Kovariansi *return* saham *i* dengan *return market*

1.4.7 Model Markowitz

Teori portofolio optimal yang dikembangkan oleh Markowitz menyoroti pentingnya hubungan antara *return* dan risiko dalam membentuk portofolio investasi yang efisien. Teori ini menekankan bahwa portofolio optimal adalah yang mampu meminimalkan risiko untuk tingkat *expected return* tertentu atau memaksimalkan *expected return* untuk tingkat risiko tertentu. Portofolio-portofolio ini membentuk sebuah kurva yang disebut *efficient frontier*, yang merupakan representasi visual dari kombinasi terbaik antara risiko dan pengembalian.

Portofolio optimal berada pada set efisien yang berhubungan dengan *Capital Allocation Line* (CAL), yaitu garis yang menunjukkan kombinasi terbaik antara aset bebas risiko dan portofolio berisiko. Posisi investor pada garis ini bergantung pada preferensi risiko mereka, apakah mereka termasuk dalam kategori *risk taker* yang lebih menyukai risiko demi peluang *return* yang lebih tinggi, atau *risk averse* yang cenderung menghindari risiko. Teori ini memiliki asumsi tertentu, seperti analisis dilakukan untuk satu periode saja, tidak ada biaya transaksi, dan keputusan investasi hanya didasarkan pada hubungan antara *return* dan risiko (Rahmawati, 2023).

Dalam penerapannya, algoritma *efficient frontier* menggunakan pendekatan simulasi untuk mencari portofolio terbaik dengan menggabungkan bobot aset secara acak. Setiap portofolio dihitung tingkat *return* yang diharapkan, risiko (standar deviasi), serta *Sharpe ratio*nya, yaitu perbandingan antara *return* dengan risiko tambahan dibandingkan aset bebas risiko. Simulasi dilakukan berulang kali untuk mendapatkan kombinasi bobot optimal yang membentuk kurva *efficient frontier*. Titik-titik pada kurva ini merepresentasikan portofolio optimal yang dapat dipilih oleh investor, tergantung pada preferensi risikonya.

Selain itu, portofolio optimal dengan *Sharpe ratio* tertinggi memberikan kombinasi *return* terbaik untuk setiap unit risiko, sedangkan portofolio dengan risiko minimum cocok bagi investor yang sangat menghindari risiko. Analisis ini menggabungkan konsep *mean-variance optimization* yang menjadi inti dari teori Markowitz. Melalui pendekatan ini, teori Markowitz memberikan landasan penting dalam pengelolaan portofolio modern, termasuk teori-teori lanjutan seperti *Capital Asset Pricing Model* (CAPM).

1.4.8 *Capital Asset Pricing Model*

Capital Asset Pricing Model (CAPM) adalah sebuah model yang digunakan untuk memahami hubungan antara risiko dan *return* yang diharapkan dari sebuah aset atau portofolio. Model ini membantu menentukan nilai wajar aset berisiko dengan mengaitkan tingkat pengembalian yang diharapkan dengan risiko yang dimiliki aset tersebut. Salah satu fokus CAPM adalah mengevaluasi saham yang memberikan *return* lebih tinggi dari yang diharapkan, serta meninjau aset bebas risiko untuk membantu menentukan tingkat pengembalian investasi.

Dalam kondisi pasar yang seimbang, CAPM menjelaskan bagaimana risiko sistematis suatu aset berisiko dikaitkan dengan pengembalian yang disyaratkan oleh investor. Model ini juga menunjukkan keseimbangan antara tingkat risiko dan keuntungan yang diharapkan dari sekuritas atau portofolio. Inti dari penggunaan CAPM adalah memberikan prediksi yang akurat mengenai hubungan antara risiko dan *return*, serta menentukan harga yang wajar untuk sebuah aset. Oleh karena itu, CAPM sering digunakan untuk memperkirakan potensi keuntungan dari suatu sekuritas, menjadikannya alat yang penting dalam dunia investasi (Todingan et al., 2023).

Pada metode CAPM, perhatian utama diberikan pada saham-saham yang memiliki *actual return* (R_i) lebih besar dibandingkan *expected return* ($E[r_i]$). Saham-saham ini dianggap sebagai *undervalued stocks*, yang berarti harganya masih berada di bawah nilai wajarnya. Sebaliknya, saham yang memiliki *actual return* (R_i) lebih kecil dari *expected return* ($E[r_i]$) disebut sebagai *overvalued stocks*. Saham-saham *overvalued* ini biasanya tidak dimasukkan dalam pembentukan portofolio optimal (Aunillah dan Wahyudi, 2022). Berikut adalah persamaan untuk menghitung *expected return* CAPM:

$$E[r_i] = R_f + \beta_i(E[R_m] - R_f) \tag{17}$$

Dengan:

$E(r_i)$ = *Expected return* CAPM saham *i*

1.4.9 Portofolio Model CAPM

Portofolio adalah kumpulan berbagai jenis aset yang dimiliki oleh seorang investor, baik itu aset riil maupun aset keuangan. Tujuan dari membentuk portofolio adalah untuk mengurangi risiko investasi. Hal ini dilakukan melalui diversifikasi, yaitu dengan menyebarkan dana ke berbagai jenis investasi yang tidak saling berkaitan atau memiliki korelasi negatif, sehingga kerugian dari satu aset bisa diimbangi oleh keuntungan dari aset lainnya (Sriwahyuni, 2023). Terdapat dua jenis portofolio saham, yaitu portofolio efisien dan portofolio optimal.

Portofolio efisien adalah kumpulan aset yang memberikan *return* tertinggi untuk tingkat risiko tertentu atau, sebaliknya, memiliki risiko terendah untuk tingkat *return* tertentu. Sebuah portofolio dapat dikatakan sebagai portofolio yang optimal apabila mampu memaksimumkan *return* dan meminimumkan risiko (Nur dan Zuliana, 2023). Berdasarkan teori *Modern Portfolio Theory* (MPT) oleh Harry Markowitz, portofolio efisien berada di garis batas efisiensi (*efficient frontier*). Saham-saham yang dimasukkan ke dalam portofolio berdasarkan model CAPM adalah saham-saham yang dianggap efisien. Dalam konteks *undervalued* dan *overvalued stocks*, saham *undervalued* dikategorikan sebagai saham efisien, sedangkan saham *overvalued* dikategorikan sebagai saham tidak efisien.

1.4.10 Bobot Saham dalam Portofolio dengan MVEP

Mean Variance Efficient Portfolio (MVEP) adalah salah satu metode untuk membangun portofolio saham dengan meminimalkan variansi dari *return* yang diharapkan. Pendekatan ini setara dengan mengoptimalkan bobot $\boldsymbol{w} = [w_1 \ w_2 \ \dots \ w_N]^T$ berdasarkan *return* yang diharapkan pada tingkat variansi tertentu. Fungsi *Lagrange* dapat digunakan untuk mengatasi permasalahan optimalisasi (Aigidya et al., 2025). Secara matematis, fungsi *Lagrange* dapat dinyatakan dalam persamaan berikut:

$$L = \boldsymbol{w^T \Sigma w} + \lambda_1(ER_p - \boldsymbol{w^T} ER_p) + \lambda_2(1 - \boldsymbol{w^T 1}_p) \quad (18)$$

Dengan:
L = Fungsi *Lagrange*
λ_1, λ_2 = Faktor pengali *Lagrange*

Dari hasil pengoptimasian fungsi *Lagrange*, akan diperoleh bobot untuk MVEP pada persamaan berikut:

$$\boldsymbol{w} = \frac{\boldsymbol{\Sigma^{-1} 1}_p}{\boldsymbol{1}_p^{\boldsymbol{T}} \boldsymbol{\Sigma^{-1} 1}_p} \quad (19)$$

Dengan:
$\boldsymbol{\Sigma^{-1}}$ = Invers matriks kovariansi saham
$\boldsymbol{1}_p$ = Vektor berdimensi $p \times 1$

Pada matriks kovariansi, diperlukan nilai variansi masing-masing saham dan nilai kovariansi antar saham. Dimana untuk nilai variansi saham dapat dilihat pada persamaan (8), dan untuk perhitungan kovariansi antar saham dapat digunakan persamaan (12). Matriks kovariansi dari portofolio yang dibentuk adalah sebagai berikut (Aigidya et al., 2025):

$$\boldsymbol{\Sigma} = \begin{bmatrix} Var(R_A) & Cov(R_A, R_B) & \cdots & Cov(R_A, R_i) \\ Cov(R_B, R_A) & Var(R_B) & \cdots & Cov(R_B, R_i) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(R_i, R_A) & Cov(R_i, R_B) & \cdots & Var(R_i) \end{bmatrix} = \begin{bmatrix} \sigma_A^2 & \sigma_{AB} & \cdots & \sigma_{Ai} \\ \sigma_{BA} & \sigma_B^2 & \cdots & \sigma_{Bi} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{iA} & \sigma_{iB} & \cdots & \sigma_i^2 \end{bmatrix} \quad (20)$$

1.4.11 Evaluasi Kinerja Portofolio

Salah satu indikator yang digunakan untuk menilai kinerja portofolio adalah *Sharpe ratio. Sharpe ratio* merupakan indikator kinerja portofolio yang mengukur hubungan antara *return* ekspektasi portofolio dan risikonya dalam suatu periode waktu. Secara matematis, rasio ini dinyatakan dalam bentuk persamaan berikut (Nasthasya et al., 2023):

$$S_p = \frac{E[R_p] - R_f}{\sigma_p} \quad (21)$$

Dengan:
S_p = *Sharpe ratio*

1.4.12 Prediksi

Prediksi, menurut Kamus Besar Bahasa Indonesia (KBBI, 2017), adalah hasil dari proses meramal. Dalam konteks yang lebih luas, prediksi dapat diartikan sebagai upaya memperkirakan kondisi masa depan, seperti proyeksi penjualan atau kebutuhan produk, sehingga produk dapat disiapkan dalam jumlah yang tepat. Proses ini melibatkan analisis data masa lalu dan menerapkannya ke masa depan melalui model matematis.

Prediksi dapat bersifat kualitatif maupun kuantitatif. Prediksi kualitatif tidak berbentuk angka dan sering kali sulit mencapai tingkat akurasi yang tinggi karena bergantung pada variabel yang sifatnya subjektif. Sebaliknya, prediksi kuantitatif berbasis angka dan menggunakan model matematis serta data historis untuk menghasilkan proyeksi yang lebih terukur dan akurat. Manfaat prediksi sangat signifikan dalam berbagai bidang. Prediksi membantu perencanaan yang lebih efektif dan efisien, menentukan kebutuhan sumber daya di masa depan, serta mendukung pengambilan keputusan yang lebih tepat dan strategis (Ngantung dan Jan, 2019).

1.4.13 *Time Series*

Time series atau runtun waktu mengacu pada kumpulan data yang diurutkan berdasarkan waktu, yang digunakan untuk menganalisis pola hubungan antara variabel tertentu dan waktu. Analisis ini sering kali digunakan untuk peramalan, seperti memprediksi nilai tukar mata uang, penjualan, atau tren ekonomi. Data *time series* dapat memiliki berbagai pola yang penting untuk dikenali sebelum melakukan analisis. Pola horizontal mencerminkan fluktuasi acak yang tidak terduga namun dapat memengaruhi data. Pola tren menunjukkan kecenderungan arah data dalam jangka panjang, baik berupa peningkatan maupun penurunan. Pola musiman menggambarkan fluktuasi yang berulang secara periodik dalam satu tahun, seperti pergerakan triwulanan atau bulanan. Sementara itu, pola siklis merujuk pada fluktuasi yang terjadi dalam periode lebih panjang, seperti siklus ekonomi yang berlangsung selama beberapa tahun.

Dalam analisisnya, data *time series* dapat dibagi menjadi univariat (satu variabel) atau multivariat. Data univariat sering dimodelkan menggunakan metode seperti *Autoregressive* (AR), *Moving Average* (MA), atau *Autoregressive Integrated Moving Average* (ARIMA), yang dirancang untuk menangkap pola dan perilaku dari data historis. Pendekatan ini memungkinkan peramalan yang lebih akurat dan membantu dalam pengambilan keputusan yang berbasis data. Misalnya, prediksi nilai tukar dolar terhadap rupiah dapat dilakukan dengan mengamati tren waktu sebelumnya dan menerapkan model *time series* yang relevan (Baidowi dan Buniarto, 2020).

1.4.14 Data *Preprocessing*

Preprocessing data merupakan tahap persiapan data sebelum memasuki proses pelatihan model. Langkah ini penting untuk memastikan data yang digunakan sudah siap dan dapat mendukung proses pelatihan secara optimal (Aprian et al., 2020).

1.4.14.1 Data *Cleaning*

Data *cleaning* adalah proses penting dalam pengolahan data, terutama untuk menangani masalah data yang tidak lengkap atau mengandung *noise*. *Missing value*, yang terjadi ketika ada bagian dataset yang kosong atau tidak terisi, menjadi salah satu tantangan utama dalam analisis data. Untuk mengatasi masalah ini, salah satu pendekatan adalah menghapus baris data yang memiliki nilai hilang, memastikan dataset yang tersisa adalah lengkap dan konsisten untuk analisis lebih lanjut. Alternatif lain adalah menggunakan metode imputasi, di mana nilai yang hilang digantikan dengan nilai tertentu, seperti rata-rata, median, modus, atau metode imputasi lainnya yang sesuai dengan konteks data. Namun, dalam penelitian ini, pendekatan yang dipilih adalah menghapus seluruh baris data yang memiliki *Missing value* untuk memastikan hasil analisis lebih akurat dan tidak bias.

1.4.14.2 Normalisasi Data

Normalisasi data merupakan langkah penting yang bertujuan untuk memastikan kualitas data sebelum digunakan dalam model pembelajaran. Proses ini dilakukan untuk menyamakan skala data, sehingga nilai dari satu fitur tidak terlalu jauh berbeda dengan nilai fitur lainnya. Dengan normalisasi, risiko bias terhadap satu fitur dapat diminimalkan, sehingga model dapat belajar dengan lebih efektif dan menghasilkan prediksi yang akurat (Fitrinanda dan Djunaidy, 2022). Salah satu metode yang umum digunakan adalah *min-max scaling*, yang mengubah data menjadi rentang tertentu, biasanya antara 0 dan 1. Metode ini dirumuskan dalam persamaan berikut (Puteri, 2023):

$$x_{norm,i(t)} = \frac{x_{i(t)} - x_{i(min)}}{x_{i(max)} - x_{i(min)}} \tag{22}$$

Dengan:

$x_{norm,i(t)}$	= Harga saham *i* periode ke-*t* setelah dinormalisasi
$x_{i(t)}$	= Harga saham aktual saham *i* periode ke-*t*
$x_{i(max)}$	= Harga tertinggi saham *i*
$x_{i(min)}$	= Harga terendah saham *i*

1.4.14.3 Pembagian Data *Training* dan *Testing*

Pembagian data adalah proses membagi dataset menjadi dua bagian utama yaitu data pelatihan (*training*) dan data pengujian (*testing*). Data pelatihan digunakan untuk melatih model, membantu algoritma memahami pola dan hubungan dalam data. Sementara itu, data pengujian digunakan untuk mengevaluasi keakuratan dan performa model yang telah dilatih. Tujuan utama dari pembagian data ini adalah memastikan model yang dihasilkan memiliki akurasi dan kinerja yang optimal dalam memprediksi data baru (Cahyadi et al., 2020).

1.4.14.4 Denormalisasi Data

Setelah pengujian dilakukan pada data uji, langkah selanjutnya adalah mengembalikan data ke skala aslinya melalui proses denormalisasi. Proses ini bertujuan untuk mengonversi *output* yang masih berada dalam rentang 0 hingga 1 agar sesuai dengan nilai aktual data. Denormalisasi dapat dihitung menggunakan persamaan berikut (Puteri, 2023):

$$d_{i(t)} = \hat{y}_{i(t)} \times \left(x_{i(max)} - x_{i(min)}\right) + x_{i(min)} \quad (23)$$

Dengan:

$\hat{y}_{i(t)}$ = Hasil prediksi saham *i* periode ke-*t* sebelum denormalisasi

$d_{i(t)}$ = Hasil prediksi saham *i* periode ke-*t* setelah denormalisasi

1.4.15 *Neural Network*

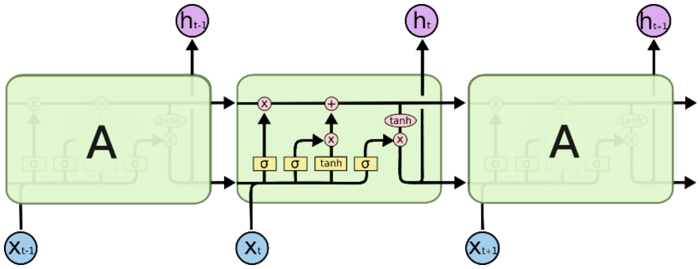
Jaringan Saraf Tiruan (*Neural Network*) adalah teknologi yang meniru cara kerja saraf manusia, bagian penting dari otak. *Neural Netrowk* (NN) terdiri dari dua lapisan utama yaitu lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*). Setiap lapisan memiliki sejumlah *neuron* (*node*) dengan fungsi aktivasi yang menentukan *output* dari *neuron* tersebut. Untuk meningkatkan kemampuan NN, dapat ditambahkan lapisan tersembunyi (*hidden layer*). NN dilatih menggunakan data pelatihan, dan semakin banyak data yang digunakan, kinerja model biasanya semakin baik. Namun, model NN dasar ini hanya efektif untuk menangani tugas regresi dan klasifikasi. Salah satu kekurangannya adalah keterbatasan dalam menghubungkan informasi baru dengan informasi sebelumnya (Karno, 2020).

Untuk mengolah data *time series* yang memiliki keterkaitan antar data, digunakan pendekatan *Recurrent Neural Network* (RNN). Meski RNN mampu menangani data *time series*, kemampuannya untuk menyimpan informasi dalam waktu lama terbatas, sehingga sering terjadi hilangnya informasi (*vanishing gradient*). Oleh karena itu, RNN lebih cocok untuk data dengan ketergantungan jangka pendek. Sebagai solusi, dikembangkan model *Long Short-Term Memory* (LSTM) yang dilengkapi dengan gerbang kompleks dan kemampuan memori yang lebih baik, memungkinkan pengolahan data dengan ketergantungan jangka panjang (Karno, 2020).

1.4.16 *Long Short Term Memory*

Long Short-Term Memory (LSTM) adalah salah satu bentuk *Recurrent Neural Network* (RNN) yang pertama kali diperkenalkan oleh Sepp Hochreiter dan Jürgen Schmidhuber pada tahun 1997. LSTM dirancang untuk mengatasi kelemahan utama RNN, yaitu masalah *vanishing gradient*, yang membuat RNN sulit mempelajari ketergantungan jangka panjang pada data. RNN sendiri memiliki koneksi berulang, di mana keluaran pada *time step* sebelumnya menjadi masukan untuk *time step* berikutnya. Mekanisme ini memungkinkan RNN memodelkan data sekuensial dengan baik. Namun, masalah *vanishing gradient* sering menghambat kemampuan RNN dalam mempelajari pola pada data *time series* yang memerlukan pemrosesan informasi jangka panjang (Fitrinanda dan Djunaidy, 2022).

Untuk mengatasi permasalahan tersebut, LSTM menggantikan *neuron* pada lapisan tersembunyi RNN dengan sel memori LSTM yang dilengkapi dengan tiga jenis gerbang, yaitu *input gate*, *forget gate*, dan *output gate*. Mekanisme ini membuat LSTM sangat efektif dalam memproses dan memprediksi data *time series* yang memiliki ketergantungan jangka panjang (Ramadhan, 2024). Struktur LSTM dirancang untuk mengelola memori pada setiap *input* secara lebih efisien, sehingga mampu mengatasi kelemahan RNN dalam mempertahankan informasi jangka panjang. Struktur ini dijelaskan lebih rinci oleh Christopher (2015) dan dapat dilihat pada Gambar 1.



Gambar 1. Struktur LSTM

Sumber: *Christoper, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

LSTM memiliki kemampuan untuk menyaring informasi melalui struktur gerbang yang dirancang untuk mempertahankan dan memperbarui keadaan dalam sel memori. Selain memiliki tiga jenis gerbang utama, setiap sel memori juga dilengkapi dengan tiga lapisan sigmoid dan satu lapisan tanh.

Lapisan sigmoid menghasilkan nilai antara 0 dan 1, yang menunjukkan seberapa banyak informasi yang perlu dipertahankan atau dibuang. Nilai 0 berarti "tidak ada informasi yang diperbolehkan lewat", sementara nilai 1 berarti "biarkan semua informasi lewat". Fungsi aktivasi sigmoid akan menghasilkan nilai mendekati 0 jika berada pada rentang $0 \leq \sigma \leq 0{,}5$ dan mendekati 1 jika berada pada rentang $0{,}5 < \sigma \leq 1$ (Ramadhan, 2024). Fungsi aktivasi sigmoid dapat dinyatakan pada persamaan berikut:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{24}$$

Lapisan tanh mengatur nilai *input* agar berada dalam rentang -1 hingga 1, sehingga data tetap stabil dan terkontrol selama proses pembelajaran. Pada lapisan ini, terdapat fungsi aktivasi tanh yang membantu menyeimbangkan informasi baru yang masuk ke sel memori dan memastikan data yang disimpan relevan. Bersama fungsi aktivasi sigmoid, tanh memungkinkan LSTM menangani ketergantungan jangka panjang dengan lebih efektif. Fungsi aktivasi tanh didefinisikan dalam persamaan sebagai berikut:

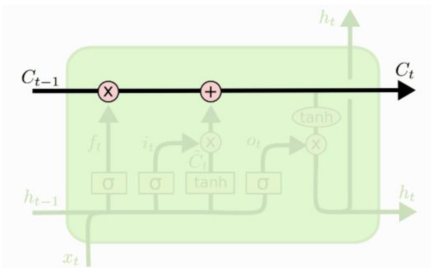
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{25}$$

Jika fungsi tanh dihubungkan dengan fungsi sigmoid pada persamaan (24), maka hubungan tersebut dapat dinyatakan melalui persamaan berikut:

$$\tanh(x) = 2\sigma(2x) - 1 \tag{26}$$

Dengan:
$\sigma(x)$ = Fungsi aktivasi sigmoid
x = Data *input*

Kunci utama LSTM adalah *cell state*, jalur horizontal yang menghubungkan *cell state* lama (C_{t-1}) dengan *cell state* baru (C_t). Jalur ini memungkinkan informasi penting mengalir dari satu langkah ke langkah berikutnya dengan sedikit modifikasi, menjaga informasi tetap utuh selama proses pengolahan data (Aulia, 2020). Sebagai tempat penyimpanan jangka panjang, *cell state* memungkinkan LSTM mempertahankan atau memperbarui informasi sesuai kebutuhan tanpa kehilangan data penting. Hal ini membuat LSTM mampu mengenali pola data yang bergantung pada hubungan jangka panjang, seperti pada teks atau *time series*. Mekanisme ini adalah fondasi utama yang membuat LSTM efektif dalam mengolah data sekuensial dengan kompleksitas tinggi. *Cell state* berdasarkan (Christoper, 2015) dapat dilihat pada Gambar 2.

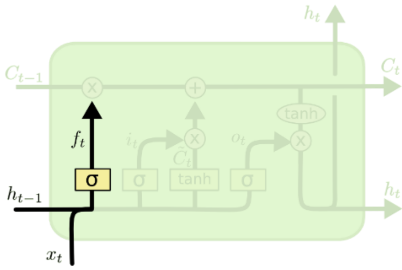


Gambar 2. *Cell state*

Sumber: Christoper, 2015. *https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

Langkah pertama dalam proses LSTM adalah menentukan informasi mana yang harus dibuang dari *cell state*. Keputusan ini dibuat melalui lapisan sigmoid yang dikenal sebagai *forget gate layer*. *Forget gate* menerima dua jenis *input*, yaitu *output* dari *timestep* sebelumnya (h_{t-1}) dan informasi eksternal pada *timestep* saat ini $(\boldsymbol{x_t})$. Kedua *input* ini digabungkan menjadi satu vektor melalui operasi konkatenasi $[h_{t-1}, \boldsymbol{x_t}]$. Hasilnya kemudian diteruskan ke fungsi sigmoid (σ), yang menghasilkan nilai antara 0 dan 1 untuk setiap elemen pada *cell state* sebelumnya (C_{t-1}).

Nilai yang dihasilkan menunjukkan seberapa besar informasi yang perlu dipertahankan atau dibuang. Angka mendekati 0 berarti informasi tersebut akan dibuang sepenuhnya, sementara angka mendekati 1 berarti informasi akan dipertahankan sepenuhnya. Dengan mekanisme ini, *forget gate* membantu LSTM secara selektif menyaring informasi yang tidak relevan, sehingga hanya data penting yang dilanjutkan ke langkah berikutnya (Ramadhan, 2024).



Gambar 3. Struktur LSTM *forget gate*

Sumber: Christoper, 2015. *https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

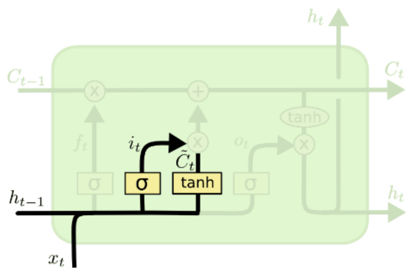
Dimana, f_t dapat dirumuskan sebagai berikut:

$$f_t = \sigma(\boldsymbol{W_f}[h_{t-1}, \boldsymbol{x_t}] + b_f) \tag{27}$$

Dengan:

f_t	= *Forget gate* ke-*t*
$\boldsymbol{W_f}$	= Vektor *weight* untuk *forget gate*
h_{t-1}	= Nilai *output* orde ke-(*t-1*)
$\boldsymbol{x_t}$	= Vektor *input* pada orde ke-*t*
b_f	= Nilai bias pada *forget gate*

Langkah selanjutnya dalam mekanisme kerja LSTM adalah memutuskan informasi mana yang akan disimpan dalam *cell state*. Proses ini dimulai dengan dua langkah utama. Pertama, sebuah lapisan sigmoid yang dikenal sebagai *input gate* (i_t) menentukan nilai-nilai yang perlu diperbarui dalam *cell state*. Kedua, lapisan tanh menciptakan vektor kandidat baru $(\tilde{C}_t)$ yang siap ditambahkan ke *cell state*. Kombinasi kedua langkah ini menghasilkan pembaruan pada *cell state*. *Input gate* memiliki peran krusial dalam menentukan seberapa banyak informasi dari *input* pada waktu saat ini $(\boldsymbol{x_t})$ yang akan disimpan dalam *cell state* (C_t). Mekanisme ini membantu mencegah informasi yang tidak relevan atau tidak signifikan memasuki memori.



Gambar 4. Struktur LSTM *input gate*

Sumber: Christoper, 2015. *https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

Dalam proses ini, *input gate* bekerja melalui persamaan berikut:

$$i_t = \sigma(\boldsymbol{W_i}[h_{t-1}, \boldsymbol{x_t}] + b_i) \quad (28)$$

Dengan:

i_t = *Input gate* ke-*t*
$\boldsymbol{W_i}$ = Vektor *weight* untuk *input gate*
b_i = Nilai bias pada *input gate*

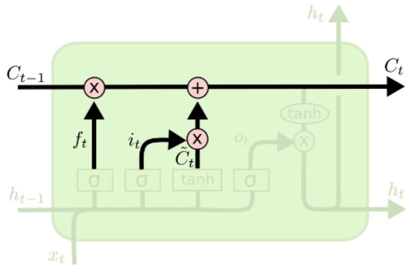
Setelah nilai *input gate* dihitung, lapisan tanh membuat vektor kandidat baru untuk *cell state* ($\tilde{C}t$) melalui persamaan berikut:

$$\tilde{C}_t = \tanh(\boldsymbol{W_c}[h_{t-1}, \boldsymbol{x_t}] + b_c) \quad (29)$$

Dengan:

$\tilde{C}_t$ = Nilai baru yang dapat ditambahkan ke *cell state*
$\boldsymbol{W_c}$ = Nilai *weight* untuk *cell state*
b_c = Nilai bias pada *cell state*

Pembaruan pada *cell state* yang lama (C_{t-1}) dilakukan dengan mengalikan nilai lama dengan *forget gate* (f_t), sehingga informasi yang tidak relevan dihapus. Selanjutnya, nilai i_t dikalikan dengan $\tilde{C}_t$, dan hasilnya ditambahkan ke *cell state* untuk menyimpan informasi baru yang relevan. Proses ini memastikan bahwa *cell state* selalu terbarui dengan informasi yang penting dan terkini.



Gambar 5. Struktur LSTM pembaruan *cell state*

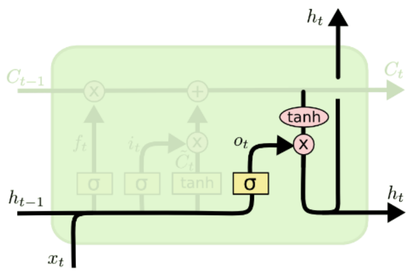
Sumber: Christoper, 2015. *https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

Dapat dirumuskan secara matematis sebagai berikut:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{30}$$

Dengan:
C_t = *Cell state* orde ke-*t*
C_{t-1} = *Cell state* sebelum orde ke-*t*

Langkah terakhir dalam proses kerja LSTM adalah menentukan nilai keluaran (*output*). Nilai *output* ini diperoleh berdasarkan *cell state* yang telah difilter melalui serangkaian langkah. Pertama, lapisan sigmoid digunakan untuk menentukan bagian dari *cell state* mana yang akan menjadi *output*. Selanjutnya, *cell state* diproses menggunakan fungsi aktivasi tanh untuk membatasi nilai *output* dalam rentang -1 hingga 1. Hasil dari tanh ini kemudian dikalikan dengan *output* dari lapisan sigmoid untuk menghasilkan keluaran akhir.



Gambar 6. Struktur LSTM output gate

Sumber: Christoper, 2015. *https://colah.github.io/posts/2015-08-Understanding-LSTMs/*

Pada mekanisme ini, *output gate* memiliki peran penting dalam mengontrol informasi yang akan diteruskan sebagai *output*. *Output gate* pertama-tama menghitung nilai sigmoid menggunakan persamaan berikut:

$$O_t = \sigma(\boldsymbol{W_o}[h_{t-1}, \boldsymbol{x_t}] + b_o) \quad (31)$$

Dengan:

O_t = Nilai *output gate*
$\boldsymbol{W_o}$ = Vektor *weight* untuk *output gate*
b_o = Nilai bias pada *output gate*

Selanjutnya, nilai akhir *cell state* (C_t) yang telah diproses melalui tanh dikalikan dengan O_t untuk menghasilkan nilai *output* akhir sel:

$$h_t = O_t * \tanh(C_t) \quad (32)$$

Dengan demikian, *output gate* memadukan informasi dari *cell state* dan *output* sebelumnya untuk menentukan keluaran yang relevan di setiap waktu (Ramadhan, 2024).

Setelah mekanisme *Long Short-Term Memory* (LSTM) memproses informasi dalam *hidden state* (h_t), tahap akhir dalam proses ini adalah menghasilkan nilai prediksi ($\hat{y}_{i(t)}$). *Hidden state* terakhir yang diperoleh dari LSTM masih berada dalam ruang vektor dengan dimensi tertentu dan belum dapat langsung diinterpretasikan sebagai hasil prediksi. Oleh karena itu, diperlukan lapisan tambahan berupa *dense layer* dengan fungsi aktivasi *linear* untuk mengubah *hidden state* menjadi *output* yang sesuai dengan skala target yang diharapkan.

Lapisan *dense* ini bertindak sebagai lapisan *linear* yang mengubah *hidden state* menjadi nilai prediksi dengan menggunakan transformasi *linear* sederhana. Secara matematis, transformasi ini dinyatakan sebagai:

$$\hat{y}_{i(t)} = W_y . h_t + b_y \quad (33)$$

Dengan:

W_y = Bobot dari *dense layer*
b_y = Bias yang digunakan untuk menyesuaikan *output*

Karena fungsi aktivasi yang digunakan dalam lapisan ini adalah *linear*, nilai keluaran yang dihasilkan hanya bergantung pada bobot dan bias tanpa mengalami transformasi *non-linear* tambahan. Dalam konteks implementasi menggunakan *TensorFlow/Keras*, fungsi aktivasi *linear* pada lapisan *dense* memungkinkan model untuk menghasilkan *output* dalam skala yang sama dengan target yang ingin diprediksi. Hal ini menjadikan pendekatan ini sangat efektif dalam tugas-tugas regresi, seperti prediksi harga saham karena model dapat langsung menghasilkan nilai kontinu tanpa memerlukan normalisasi tambahan. Dengan demikian, *hidden state* yang telah diproses oleh LSTM dapat langsung dimanfaatkan untuk menghasilkan prediksi yang relevan sesuai dengan tujuan pemodelan.

1.4.17 *Adaptive Moment Estimation*

Optimasi berbasis gradien stokastik memainkan peran penting dalam berbagai bidang sains dan teknik. Banyak permasalahan di bidang ini dapat dirumuskan sebagai upaya untuk mengoptimalkan fungsi tujuan dengan memaksimalkan atau meminimalkan nilai parameter. Salah satu algoritma optimasi yang banyak digunakan adalah *Adaptive Moment Estimation* (Adam), yang dirancang untuk menggabungkan kelebihan dua metode sebelumnya, yaitu *Adaptive Gradient* (AdaGrad) yang unggul dalam menangani gradien jarang dan *Root Mean Square Propagation* (RMSProp) yang efektif dalam mengatasi perubahan gradien.

Keunggulan utama Adam terletak pada kemampuannya untuk tidak hanya menyesuaikan tingkat pembelajaran berdasarkan rata-rata pertama (*mean*) seperti RMSProp, tetapi juga memanfaatkan rata-rata kedua (varians tak terpusat) dari gradien. Algoritma ini secara adaptif menghitung rata-rata bergerak eksponensial dari gradien dan gradien kuadratnya, dengan parameter β_1 dan β_2 yang mengatur tingkat peluruhan, masing-masing biasanya bernilai 0,9 dan 0,999, serta epsilon (ε) sebesar 10^{-8}. Fitur ini membuat Adam unggul dalam menangani pembaruan parameter yang konsisten, tidak dipengaruhi oleh penskalaan gradien, serta memiliki ukuran pembaruan yang dibatasi oleh *hyperparameter stepsize*. Selain itu, Adam juga mampu bekerja secara efektif pada data non-stasioner (Ramadhan, 2024). Penelitian Riyantoko et al., (2020) menemukan bahwa penggunaan Adam pada algoritma LSTM dengan variasi *epoch* menghasilkan penurunan nilai RMSE yang konsisten. Hal ini mengindikasikan bahwa Adam mampu meningkatkan akurasi prediksi harga penutupan saham, menjadikannya algoritma yang andal dalam konteks pengolahan data finansial.

1.4.18 Evaluasi Model

Evaluasi model adalah langkah untuk mengukur tingkat akurasi model dalam memprediksi data berdasarkan pengujian sebelumnya. Tujuan evaluasi ini adalah menilai seberapa baik model yang telah dilatih dalam memproses data dan menghasilkan prediksi. Menurut Wiranda dan Sadikin (2020), *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE) adalah metrik yang umum digunakan. RMSE mengukur kesalahan prediksi dalam satuan asli data, sedangkan MAPE mengukur kesalahan dalam bentuk persentase. Keduanya cocok untuk evaluasi harga penutupan saham karena dapat menilai deviasi absolut dan persentase kesalahan prediksi secara efektif.

1.4.18.1 *Root Mean Square Error*

Root Mean Square Error (RMSE) adalah metrik yang digunakan untuk mengevaluasi akurasi suatu model dengan menghitung seberapa besar tingkat kesalahannya. RMSE dihitung dengan cara mengurangi data aktual dengan data hasil prediksi, mengkuadratkan selisih tersebut, lalu mengambil akar kuadrat dari rata-rata hasil tersebut. Proses ini membantu mengukur kesalahan model secara lebih mendalam.

RMSE merupakan akar dari rata-rata nilai kesalahan yang dikuadratkan. Metode ini digunakan untuk mengevaluasi model dengan mengidentifikasi nilai RMSE terkecil, yang menunjukkan tingkat akurasi yang optimal. Dengan demikian, RMSE sangat efektif dalam mengukur performa model dan memberikan wawasan yang mendalam tentang tingkat kesalahan prediksi (Fitrinanda dan Djunaidy, 2022). Persamaan untuk menghitung RMSE adalah sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{34}$$

Dengan:

y_t = Nilai aktual pada waktu ke-*t*

$\hat{y}_t$ = Nilai prediksi pada waktu ke-*t*

1.4.18.2 *Mean Absolute Percentage Error*

Mean Absolute Percentage Error (MAPE) adalah metrik yang mengukur persentase kesalahan prediksi berdasarkan perbedaan antara nilai aktual dan hasil prediksi. Persamaan untuk menghitung MAPE adalah sebagai berikut (Puteri, 2023):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{(y_t - \hat{y}_t)}{y_t} \times 100\right| \tag{35}$$

Pada model LSTM, semakin rendah nilai MAPE, semakin akurat prediksi terhadap nilai sebenarnya. Model dengan MAPE terkecil menunjukkan kinerja terbaik dalam melakukan prediksi. Rentang nilai MAPE yang digunakan untuk mengukur hasil prediksi model LSTM dapat dilihat pada Tabel 1 berikut:

Tabel 1. Kriteria nilai MAPE

Skala MAPE	Interpretasi
$x < 10\%$	Sangat Akurat
$10\% \le x < 20\%$	Akurat
$20\% \le x < 50\%$	Cukup
$x \ge 50\%$	Tidak Akurat

1.4.19 Uji Stasioner

Stasionaritas merupakan konsep penting dalam analisis deret waktu, karena data yang bersifat stasioner memiliki rata-rata, varians, dan kovariansi yang tetap sepanjang waktu. Sayangnya, banyak data deret waktu yang ditemukan di dunia nyata tidak memenuhi sifat ini. Untuk mengatasi masalah tersebut, sering dilakukan proses diferensiasi, yaitu teknik menghitung selisih antar data secara berurutan guna menghilangkan tren atau pola yang mengganggu (Adiningsih, 2021). Dalam konteks pengujian stasionaritas,

Dickey dan Fuller (1979) memperkenalkan uji akar unit (*unit root*), sebuah metode populer untuk menguji stasioneritas data. Salah satu bentuk uji akar unit yang paling umum digunakan adalah *Augmented Dickey-Fuller* (ADF). Uji ADF digunakan untuk menentukan apakah data bersifat stasioner atau tidak.
Pengujian ini dilakukan dengan menyusun hipotesis sebagai berikut:

H_0 : Data tidak stasioner

H_1 : Data stasioner

Keputusan diambil dengan membandingkan nilai *p-value* terhadap taraf signifikansi α sebesar 0,05. Jika *p-value* lebih kecil dari 0,05, maka hipotesis nol tidak diterima, yang berarti data bersifat stasioner. Sebaliknya, jika *p-value* lebih besar dari 0,05, maka hipotesis nol diterima, menunjukkan bahwa data tidak stasioner (Alhafiz, 2023).

1.4.20 Homoskedastisitas dan Heteroskedastisitas

Homoskedastisitas dan heteroskedastisitas adalah konsep penting dalam analisis regresi yang berkaitan dengan pola varians dari *error* atau residual dalam model. Pemahaman terhadap kedua konsep ini menjadi krusial, karena keberadaannya dapat memengaruhi validitas estimasi parameter dan kesimpulan statistik. Dengan memahami keduanya, analisis model dapat dilakukan dengan lebih akurat, terutama dalam konteks data keuangan yang sering kali menunjukkan pola varians yang tidak konstan.

Homoskedastisitas merujuk pada situasi di mana varians dari residu atau kesalahan tetap konstan sepanjang waktu atau antar pengamatan. Ini berarti bahwa varians tidak bergantung pada nilai-nilai variabel independen (X_t), sehingga distribusi data memiliki varian yang seragam tanpa fluktuasi yang signifikan. Secara matematis, ini dapat dinyatakan dengan persamaan berikut:

$$Var[\varepsilon_t | X_t = x] = \sigma^2 \tag{36}$$

Dengan:

ε_t = Residual pada periode ke-*t*

X_t = Variabel independen pada periode ke-*t*

σ^2 = Varians tetap untuk semua periode

Heteroskedastisitas terjadi ketika varians dari residu tidak konstan dan bervariasi tergantung pada nilai-nilai yang dianalisis. Dalam hal ini, varians bergantung pada variabel independen (X_t) dan dapat berubah berdasarkan nilai-nilainya. Secara matematis, hal ini dinyatakan sebagai:

$$Var[\varepsilon_t | X_t = x] = \sigma_t^2 \tag{37}$$

Dengan:

σ_t^2 = Variansi pada periode ke-*t*

1.4.21 Model *Time series*

Model-model *time series* seperti *Autoregressive* (AR), *Moving Average* (MA), *Autoregressive Moving Average* (ARMA), dan *Autoregressive Integrated Moving Average* (ARIMA) telah menjadi alat yang sangat efektif dalam menganalisis dan meramalkan pola dalam data *time series*. Setiap model ini dirancang untuk menangkap berbagai aspek dari dinamika data, baik itu hubungan antara nilai saat ini dan masa lalu, pengaruh kesalahan sebelumnya, atau integrasi proses *differencing* untuk mengatasi ketidakstasioneran.

1.4.21.1 *Autoregressive* (AR)

Model *autoregressive* (AR) adalah proses regresi pada diri sendiri, yang berarti nilai pada waktu tertentu bergantung pada nilai-nilai sebelumnya (lag). Model AR dengan orde p dinotasikan AR(p) atau ARIMA(p,0,0). Bentuk umum dari model ini adalah sebagai berikut (Alhafiz, 2023):

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad (38)$$

Atau, dapat dinyatakan dalam persamaan berikut:

$$\left(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p\right) y_t = C + \varepsilon_t \quad (39)$$
$$\phi_p(B) y_t = C + \varepsilon_t \quad (40)$$

Dengan:
p = Orde model *autoregressive*
C = Konstanta
ϕ = Parameter model *autoregressive*
$\phi_p(B)$ = Polinomial lag untuk komponen AR

1.4.21.2 *Moving Average* (MA)

Model *Moving Average* (MA) adalah proses yang menggunakan galat sebelumnya untuk meramalkan nilai di masa depan. Model MA dengan orde q dinotasikan MA(q) atau ARIMA(0,0,q). Bentuk umum dari model ini adalah sebagai berikut:

$$y_t = C + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (41)$$

Atau, dapat dinyatakan dalam persamaan berikut:

$$y_t = C + \left(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q\right) \varepsilon_t \quad (42)$$
$$y_t = C + \theta_q(B) \varepsilon_t \quad (43)$$

Dengan:

q = Orde model *moving average*

θ = Parameter model *moving average*

$\theta_q(B)$ = Polinomial lag untuk komponen MA

1.4.21.3 *Autoregressive Moving Average* (ARMA)

Model *Autoregressive Moving Average* (ARMA) adalah kombinasi dari model AR dan MA. Model ini mengasumsikan bahwa nilai pada waktu tertentu dipengaruhi oleh nilai pada periode sebelumnya serta kesalahan atau galat dari periode sebelumnya. Model ARMA dengan orde p dan q dinotasikan ARMA(p,q) atau ARIMA(p,0,q). Secara umum, model ini dapat dinyatakan sebagai berikut (Pratami et al., 2023):

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{44}$$

Atau, dapat dinyatakan dalam persamaan berikut:

$$\left(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p\right) y_t = C + \left(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q\right) \varepsilon_t \tag{45}$$

$$\phi_p(B) y_t = C + \theta_q(B) \varepsilon_t \tag{46}$$

Dalam penerapan nyata, deret waktu yang bersifat stasioner umumnya dapat direpresentasikan dengan baik menggunakan model *autoregressive*, *moving average*, atau kombinasi keduanya. Biasanya, nilai orde AR(p) dan MA(q) tidak lebih besar dari 2, dan sering kali nilainya bahkan lebih kecil (Alhafiz, 2023).

1.4.21.4 *Autoregressive Integrated Moving Average* (ARIMA)

Kalengkongan et al. (2020) menjelaskan bahwa model deret waktu biasanya diasumsikan stasioner. Namun, banyak data ekonomi bersifat tidak stasioner, sehingga diperlukan *differencing* untuk mengubahnya menjadi stasioner. Model ARMA cocok untuk data stasioner, sementara ARIMA digunakan untuk data tidak stasioner dengan mengaplikasikan *differencing* yang tepat (Alhafiz, 2023). Model ARIMA adalah kombinasi dari tiga komponen utama, yaitu *autoregressive* (p), *integrated* (d), dan *moving average* (q). Oleh karena itu, bentuk umum model ARIMA (p,d,q) dapat dijelaskan sebagai berikut (Pratami et al., 2023):

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} + \Delta^d y_t \tag{47}$$

$$\Delta^d y_t = (1 - B)^d y_t \tag{48}$$

Dengan:

d = Orde *differencing*

$\Delta^d y_t$ = Nilai *differencing* ke-*d*

Atau, dapat dinyatakan dalam persamaan berikut:

$$\left(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p\right)(1-B)^d y_t=C+\left(1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q\right)\varepsilon_t \quad (49)$$

$$\phi_p(B)(1-B)^d y_t=C+\theta_q(B)\varepsilon_t \quad (50)$$

Model ARIMA sering kali menggunakan *differencing* pertama (d = 1) atau terkadang *differencing* kedua (d = 2) untuk membuat data stasioner sebelum menggabungkan model *autoregressive* dan *moving average*. Fungsi Autokorelasi (ACF) dan Autokorelasi Parsial (PACF) digunakan untuk menentukan model ARIMA yang paling cocok. Selanjutnya, model terbaik dipilih berdasarkan parameter signifikan dan nilai AIC tertinggi. Setelah model ARIMA terbaik ditetapkan, tahap berikutnya adalah menguji autokorelasi residual serta mengidentifikasi adanya heteroskedastisitas menggunakan uji ARCH-LM. Jika ditemukan heteroskedastisitas, langkah berikutnya adalah melanjutkan pemodelan menggunakan GARCH.

1.4.22 ARCH dan GARCH

Secara umum, pemodelan data deret waktu harus memenuhi asumsi bahwa variansnya tetap konstan (homoskedastisitas). Namun, data deret waktu di sektor keuangan cenderung sangat fluktuatif. Hal ini menunjukkan adanya heteroskedastisitas, yaitu kondisi di mana variansnya berubah-ubah dan tidak konstan seiring waktu (Alhafiz, 2023). Model *Autoregressive Heteroscedasticity* (ARCH) diperkenalkan oleh Robert Engle pada tahun 1982 untuk menangani heteroskedastisitas dalam data keuangan. Model ARCH mengasumsikan bahwa varians dari residual bergantung pada nilai-nilai sebelumnya, yang berarti varians tidak tetap tetapi berubah seiring waktu. Secara matematis, model ARCH orde m dapat dinyatakan sebagai (Garini dan Anbiya, 2022):

$$\sigma_t^2=\omega+\alpha_1\varepsilon_{t-1}^2+\alpha_2\varepsilon_{t-2}^2+\cdots+\alpha_m\varepsilon_{t-m}^2=\omega+\sum_{i=1}^{m}\alpha_i\varepsilon_{t-i}^2 \quad (51)$$

Dengan:

σ_t^2	= Varians residu pada waktu ke-*t*
ω	= Konstanta positif
α	= Parameter ARCH
m	= Orde ARCH

Dalam model ARCH, terdapat beberapa syarat penting yang harus dipenuhi agar model dapat digunakan dengan valid. Salah satu syarat utama adalah bahwa nilai intersep varians (ω) harus lebih besar dari nol ($\omega > 0$). Hal ini memastikan bahwa varians bersyarat (σ_t^2) memiliki komponen dasar yang positif, sehingga varians tidak pernah bernilai nol atau negatif.Selain itu, koefisien ARCH (α_m) untuk setiap $m > 0$ harus bernilai nol atau lebih ($\alpha_m \geq 0$). Syarat ini memastikan bahwa kontribusi dari kejutan di masa lalu terhadap varians saat ini tetap positif atau tidak memengaruhi varians sama sekali. Dengan kata lain, syarat ini menjaga agar varians bersyarat (σ_t^2) tetap konsisten dengan konsep varians dalam statistik, yang secara definisi tidak boleh negatif. Model ini membentuk proses berdasarkan persamaan berikut:

$$\varepsilon_t = \sigma_t v_t \tag{52}$$

Dengan:

σ_t = Akar positif dari σ_t^2

v_t = *White noise* periode ke-*t*

v_t merupakan variabel acak yang mengikuti sifat *Independent and Identically Distributed* (IID). v_t sering diasumsikan sebagai *white noise* dengan rata-rata nol dan variansi satu. Biasanya, diasumsikan mengikuti distribusi normal standar N(0,1) atau distribusi Student-t. Selain itu, v_t dianggap independen dengan ε_t, $s < t$. Dengan adanya asumsi ini, mengakibatkan distribusi bersyarat dari ε_t akan berdistribusi normal N(0, σ^2). Hal ini menunjukkan bagaimana model ARCH mampu menangkap karakteristik volatilitas dalam data yang sering kali berubah seiring waktu.

Model *Generalized Autoregressive Heteroscedasticity* (GARCH) merupakan penyempurnaan dari model ARCH. Model ini dirancang untuk mengatasi masalah orde yang terlalu tinggi pada ARCH(m). Dalam GARCH, variansi bersyarat dipengaruhi tidak hanya oleh *error* dari periode sebelumnya tetapi juga oleh variansi bersyarat itu sendiri. Variansi bersyarat dalam model GARCH terdiri dari dua elemen utama yaitu *error* kuadrat dari periode sebelumnya dan variansi bersyarat pada periode sebelumnya. Secara umum, model GARCH(m,s) dapat dirumuskan sebagai berikut (Badu, 2021):

$$\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \cdots + \alpha_m\varepsilon_{t-m}^2 + \beta_1\sigma_{t-1}^2 + \beta_2\sigma_{t-2}^2 + \cdots + \beta_s\sigma_{t-s}^2 = \omega + \sum_{i=1}^{m}\alpha_i\varepsilon_{t-i}^2 + \sum_{j=1}^{s}\beta_j\sigma_{t-j}^2 \tag{53}$$

Dengan:

β = Parameter GARCH

s = Orde GARCH

Dalam model GARCH, terdapat beberapa syarat yang harus dipenuhi agar model valid dan dapat digunakan dengan baik. Pertama, parameter-parameter dalam model harus bernilai positif. Ini berarti bahwa nilai intersep varians (ω) harus lebih besar dari nol ($\omega > 0$), sedangkan koefisien ARCH (α_m) dan koefisien GARCH (β_s) harus bernilai nol atau lebih ($\alpha_m \geq 0$ dan $\beta_s \geq 0$). Kondisi ini memastikan bahwa varians bersyarat (σ_t^2) selalu bernilai positif, sehingga hasil perhitungan model menjadi realistis. Selain itu, agar model bersifat stasioner, jumlah koefisien α_m dan GARCH β_s harus memenuhi syarat $\sum_{m=1}^{M}\alpha_m + \sum_{s=1}^{S}\beta_s < 1$. Kondisi ini penting untuk mencegah divergensi varians bersyarat seiring waktu, yang berarti model tetap terkendali dan dapat memberikan estimasi yang stabil. Ketika s = 0, model tersebut akan tereduksi menjadi model ARCH(m) (W Alhafiz, 2023).

1.4.23 Uji Asumsi

Uji asumsi merupakan langkah penting dalam penelitian untuk memastikan model yang digunakan memenuhi syarat-syarat statistik tertentu sehingga hasil analisis dapat diandalkan. Pada uji asumsi mencakup beberapa jenis pengujian, yaitu uji signifikansi parameter, uji normalitas, uji autokorelasi, dan uji heteroskedastisitas.

1.4.23.1 Uji Signifikansi Parameter

Uji signifikansi parameter merupakan langkah penting dalam pemodelan ARIMA dan GARCH untuk memastikan bahwa parameter yang diestimasi memberikan kontribusi signifikan terhadap model. Uji ini dilakukan menggunakan metode statistik seperti uji t-statistik. Pada model ARIMA, uji signifikansi parameter digunakan untuk menentukan apakah parameter *autoregressive* (ϕ) dan *moving average* (θ) memengaruhi perilaku data secara signifikan. Sementara itu, pada model GARCH, uji ini diterapkan pada parameter varians, yaitu intersep koefisien ARCH (α), dan koefisien GARCH (β). Statistik uji t dapat dihitung menggunakan persamaan berikut (Badu, 2021):

$$t = \frac{\hat{\theta}}{SE(\hat{\theta})} \tag{54}$$

Dengan:

$\hat{\theta}$ = Nilai estimasi parameter

$SE(\hat{\theta})$ = Standar *error* dari parameter

Uji t dilakukan dengan menyusun hipotesis sebagai berikut:

H_0: Parameter tidak signifikan

H_1: Parameter signifikan

Tingkat signifikansi yang digunakan dalam uji adalah 0,05. Keputusan diambil berdasarkan nilai *p-value* yang diperoleh. Jika *p-value* lebih besar dari 0,05, maka hipotesis nol diterima, yang berarti parameter tidak signifikan dalam model. Sebaliknya, jika *p-value* kurang dari atau sama dengan 0,05, maka hipotesis nol tidak diterima, yang menunjukkan bahwa parameter signifikan dalam model dan memberikan kontribusi dalam menjelaskan pola data.

1.4.23.2 Uji Normalitas

Uji *Kolmogorov-Smirnov* (K-S) digunakan untuk menguji apakah data berdistribusi normal dengan membandingkan fungsi distribusi kumulatif data dengan distribusi normal teoretis. Hipotesis awal menyatakan bahwa data mengikuti distribusi normal. Statistik uji *Kolmogorov-Smirnov* dapat diamati pada persamaan berikut:

$$D = \sup_{x} |S(x) - F(x)| \tag{55}$$

Dengan:

D = Statistik uji Kolmogorov-Smirnov

$S(x)$ = Fungsi distribusi kumulatif empiris data

$F(x)$ = Fungsi distribusi kumulatif distribusi normal teoretis

Uji *Kolmogorov-Smirnov* dilakukan dengan menyusun hipotesis sebagai berikut:

H_0: Data berdistribusi normal

H_1: Data tidak berdistribusi normal

Tingkat signifikansi yang digunakan dalam uji adalah 0,05. Keputusan diambil berdasarkan nilai *p-value* yang diperoleh. Jika *p-value* lebih besar dari 0,05, maka hipotesis nol diterima, yang berarti data berdistribusi normal. Sebaliknya, jika *p-value* kurang dari atau sama dengan 0,05, maka hipotesis nol tidak diterima, yang menunjukkan bahwa data tidak berdistribusi normal.

1.4.23.3 Uji Autokorelasi

Uji *Ljung-Box* adalah metode statistik yang dirancang untuk mendeteksi keberadaan autokorelasi pada residual hingga lag tertentu. Metode ini diperkenalkan oleh Ljung dan Box pada tahun 1978 dan sering digunakan untuk mengevaluasi kualitas model statistik seperti ARIMA dan GARCH. Pada model ARIMA, uji *Ljung-Box* membantu memastikan bahwa residual dari model telah bebas dari autokorelasi, yang menjadi indikasi bahwa model sudah mampu menangkap pola dalam data secara memadai. Jika residual masih menunjukkan adanya autokorelasi, maka model perlu disesuaikan, misalnya dengan menambahkan atau mengubah orde AR atau MA.

Dalam konteks model GARCH, fokus utama dari uji *Ljung-Box* adalah untuk memeriksa standar residual yang dihasilkan. Standar residual digunakan untuk memastikan bahwa dinamika volatilitas bersyarat telah dijelaskan dengan baik oleh model, tanpa adanya pola autokorelasi yang tersisa. Dengan demikian, model GARCH yang baik harus menghasilkan standar residual yang bersifat acak dan tidak memiliki pola autokorelasi yang signifikan. Statistik uji dihitung menggunakan persamaan berikut (Brilliantya, 2022):

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k} \tag{56}$$

Dengan:

Q = Statistik uji *Ljung-Box*

n = Banyaknya data

$\hat{\rho}_k$ = Koefisien autokorelasi pada lag ke-*k*

h = Jumlah lag yang akan diuji

Uji *Ljung-Box* dilakukan dengan menyusun hipotesis sebagai berikut:

H_0: Tidak ada autokorelasi pada residual hingga lag tertentu

H_1: Terdapat autokorelasi pada residual hingga lag tertentu

Tingkat signifikansi yang digunakan dalam uji adalah 0,05. Keputusan diambil berdasarkan nilai *p-value* yang diperoleh. Jika *p-value* lebih besar dari 0,05, maka hipotesis nol diterima, yang berarti residual bebas dari autokorelasi, menunjukkan bahwa model telah mampu menjelaskan dinamika data dengan baik. Sebaliknya, jika *p-value* kurang dari atau sama dengan 0,05, hipotesis nol tidak diterima, yang menunjukkan masih adanya autokorelasi pada residual. Oleh karena itu, model perlu disesuaikan lebih lanjut untuk mengurangi autokorelasi tersebut.

1.4.23.4 Uji Heteroskedastisitas

Uji heteroskedastisitas adalah metode statistik yang digunakan untuk memeriksa apakah varians residual atau galat pada suatu model regresi bersifat konstan (homoskedastisitas) atau bervariasi (heteroskedastisitas). Dalam model regresi yang baik, asumsi homoskedastisitas sangat penting karena varians residual yang konstan memastikan hasil estimasi parameter menjadi efisien dan valid. Jika varians residual tidak konstan, atau terjadi heteroskedastisitas, maka estimasi parameter model bisa menjadi bias dan tidak efisien, yang dapat memengaruhi akurasi prediksi model.

Heteroskedastisitas sering muncul pada data ekonomi atau keuangan, di mana variabilitas nilai residual cenderung berubah seiring dengan tingkat variabel independen atau waktu. Untuk mendeteksi keberadaan heteroskedastisitas, beberapa metode uji dapat digunakan, salah satunya adalah uji *Lagrange Multiplier* (LM) yang sering digunakan dalam model *time series*. Menurut Engle dalam Gujarati (2003), uji LM melibatkan perhitungan statistik yang mengikuti distribusi *Chi Kuadrat* dengan derajat kebebasan (d_f) yang sesuai dengan jumlah parameter yang ditambahkan. Hal ini dapat dinyatakan dalam persamaan berikut (Brilliantya, 2022):

$$LM = nR^2 \sim \chi^2 \tag{57}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{58}$$

Dengan:

LM = Statistik uji LM

R^2 = Besarnya kontribusi keragaman yang dapat dijelaskan oleh model

Uji heteroskedastisitas dilakukan dengan menyusun hipotesis sebagai berikut:

H_0: Tidak ada heteroskedastisitas dalam model (homoskedastisitas)

H_1: Terdapat heteroskedastisitas dalam model (heteroskedastisitas)

Tingkat signifikansi yang digunakan dalam uji adalah 0,05. Keputusan diambil berdasarkan nilai *p-value* yang diperoleh. Jika *p-value* lebih besar dari 0,05, maka hipotesis nol diterima, yang berarti tidak terdapat heteroskedastisitas yang signifikan, menunjukkan bahwa model memenuhi asumsi homoskedastisitas. Sebaliknya, jika *p-value* kurang dari atau sama dengan 0,05, maka hipotesis nol tidak diterima, yang menunjukkan adanya heteroskedastisitas dalam model. Oleh karena itu, model perlu disesuaikan lebih lanjut, dengan menggunakan model GARCH.

1.4.24 *Akaike Information Criterion*

Akaike Information Criterion (AIC) adalah suatu metode yang digunakan untuk memilih model terbaik dengan mempertimbangkan keseimbangan antara kualitas kesesuaian model terhadap data dan jumlah parameter yang digunakan. Dalam membandingkan 2 atau lebih model ARIMA atau GARCH, maka model dengan AIC yang paling kecil merupakan model yang lebih baik. AIC dihitung dengan menggunakan persamaan berikut (Garini dan Anbiya, 2022):

$$AIC = n \log\left(\frac{RSS}{n}\right) + 2k \tag{59}$$

$$RSS = \sum_{i=1}^{n} \varepsilon_i^2 \tag{60}$$

Dengan:

RSS = *Residual Sum of Squares* (jumlah kuadrat residu)

k = Jumlah parameter dalam model

BAB II
METODE PENELITIAN

2.1 Jenis dan Sumber Data

Pendekatan yang digunakan dalam penelitian ini adalah kuantitatif deskriptif dengan metode penelitian eksperimental bersifat prediktif. Pendekatan kuantitatif deskriptif dipilih karena penelitian ini berfokus pada pengolahan data numerik berupa harga penutupan harian saham, yang dianalisis menggunakan teknik statistik dan algoritma *machine learning* untuk menggambarkan pola serta karakteristik data secara matematis. Penelitian ini juga bersifat eksperimental prediktif, karena melibatkan pengujian model prediksi LSTM-GARCH, untuk memperkirakan harga saham masa depan berdasarkan data historis. Selain itu, penelitian bertujuan menentukan portofolio optimal dengan mempertimbangkan *return* dan risiko, yang hasilnya dibandingkan dengan pendekatan konvensional seperti CAPM.

Data yang digunakan dalam penelitian ini adalah data kuantitatif berupa data sekunder harga penutupan saham harian (*close*) selama dua tahun terakhir, yaitu dari November 2022 hingga November 2024. Proses pengolahan dan analisis data dilakukan menggunakan perangkat lunak *Google Colaboratory* untuk mendukung pemodelan, analisis, serta optimasi portofolio berbasis *machine learning* (LSTM-GARCH). Sementara itu, *Microsoft Excel* digunakan untuk menghitung portofolio optimal berdasarkan pendekatan konvensional CAPM. Kombinasi metode dan alat ini diharapkan dapat memberikan hasil yang komprehensif dalam membandingkan kedua pendekatan tersebut.

2.2 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Big Data, Program Studi Ilmu Aktuaria, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin. Pengambilan data berupa harga penutupan harian 13 saham yang terpilih dari indeks LQ45, melalui situs *https://id.investing.com/* dan *https://finance.yahoo.com/*. Penelitian ini berlangsung mulai dari bulan Oktober hingga Desember 2024.

2.3 Objek Penelitian

Objek penelitian ini adalah harga penutupan saham (*close*) dari 13 perusahaan yang terdaftar di indeks LQ45. Pemilihan ke-13 saham tersebut didasarkan pada tingkat pengembalian tahunan tertinggi selama periode pengamatan dua tahun terakhir. Dari total 45 saham yang terdapat dalam indeks LQ45, 13 saham dengan performa terbaik tersebut dipilih sebagai fokus penelitian. Adapun saham-saham yang dimaksud adalah BRIS, ADRO, ESSA, ARTO, PGAS, SIDO, ICBP, INDF, PTBA, BBCA, GOTO, INTP, dan UNTR.

2.4 Metode Pengumpulan Data

Metode pengumpulan data dilakukan melalui studi literatur dan pengunduhan data harga penutupan saham harian dari situs *https://id.investing.com/.* Selain itu, analisis dan pengembangan model didukung oleh berbagai referensi, termasuk buku, jurnal, skripsi, serta artikel yang membahas model LSTM, ARCH/GARCH, CAPM, dan optimasi portofolio dengan pendekatan *Mean-Variance*.

2.5 Metode Analisis Data

Penelitian ini membandingkan dua pendekatan analisis, yaitu pendekatan berbasis *machine learning* (LSTM- GARCH) dan pendekatan konvensional menggunakan *Capital Asset Pricing Model*.

2.5.1 Metode *Machine learning* (LSTM-GARCH)

Berikut ini adalah langkah-langkah analisis yang diterapkan pada model LSTM:

1. Membaca dataset saham, fokus penelitian ini adalah pada kolom "*Close*".
2. Melakukan *exploratory data analysis* dan data *preprocessing*.
3. Melakukan normalisasi harga saham menggunakan *min-max scaling* ke rentang [0, 1].
4. Membagi data menjadi data latih (*training*) dengan proporsi 80% dan data uji (*testing*) dengan proporsi 20%.
5. Menentukan jumlah *neuron* dan *epoch* yang dilakukan dengan pendekatan *trial and error*. Proses ini menggunakan metrik evaluasi MAPE dan RMSE sebagai acuan untuk memilih model terbaik.
6. Melatih model LSTM pada data latih (*training*) dan melakukan prediksi terhadap data uji (*testing*).
7. Melakukan denormalisasi (mengembalikan hasil prediksi ke skala awal).
8. Menyimpan hasil prediksi harga saham untuk analisis volatilitas.

Selanjutnya model GARCH digunakan untuk memodelkan volatilitas saham berdasarkan hasil prediksi yang diperoleh dari model LSTM:

1. Menghitung *return* harian untuk setiap saham.
2. Melakukan analisis deskriptif untuk data *return* harian.
3. Menguji kestasioneran data *return* dengan menggunakan *Augmented Dickey-Fuller Test* (ADF *Test*). Jika data tidak stasioner, dilakukan *differencing* hingga data menjadi stasioner.
4. Mengidentifikasi model *Box-Jenkins* (ARIMA) berdasarkan visualisasi plot ACF dan PACF.
5. Mengestimasi parameter untuk berbagai model ARIMA dan menemukan parameter yang signifikan. Memilih model ARIMA terbaik berdasarkan nilai *Akaike Information Criterion* (AIC).
6. Melakukan *diagnostic checking* untuk mendeteksi adanya heteroskedastisitas. Jika tidak ada heteroskedastisitas, analisis berhenti pada model ARIMA dan volatilitas

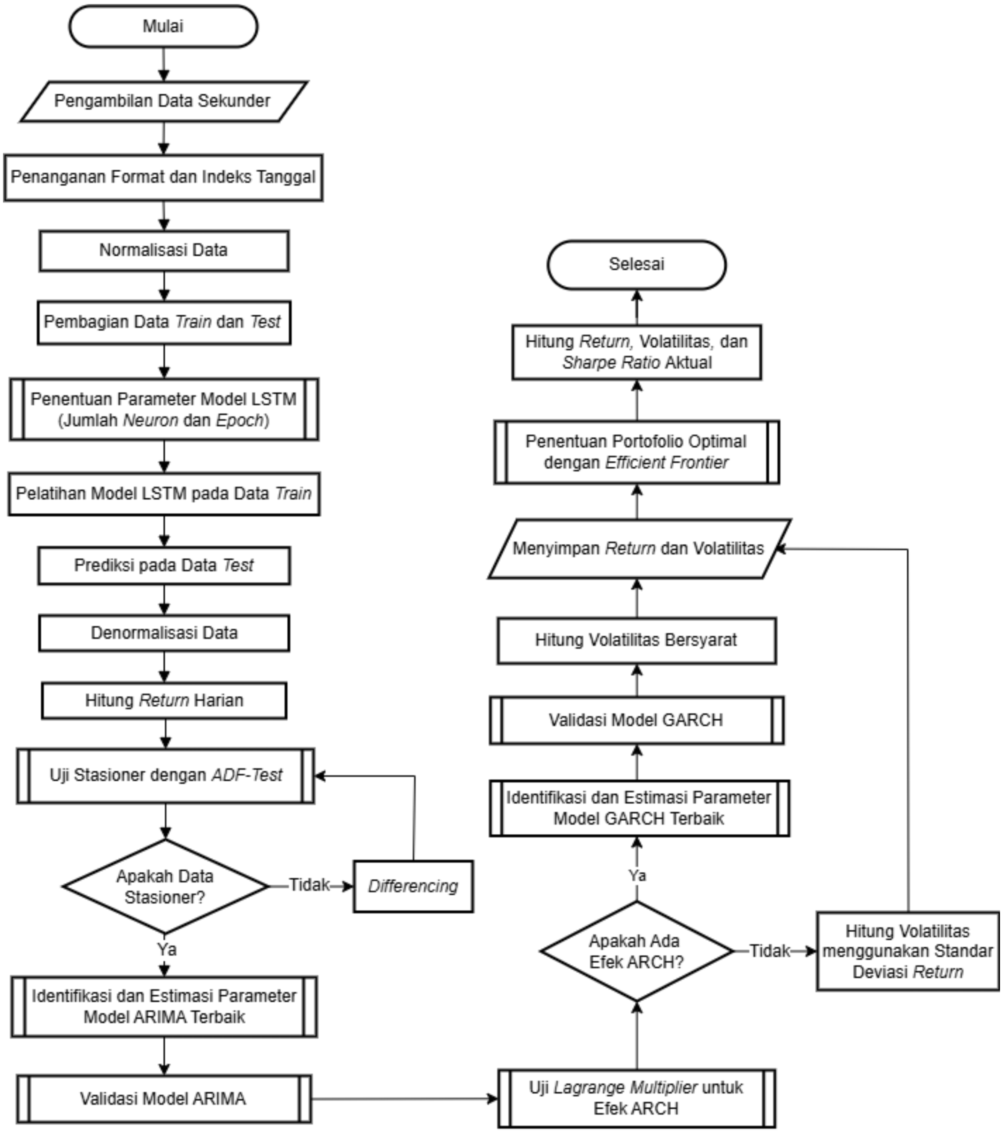
diukur menggunakan standar deviasi *return*. Jika terdeteksi ada heteroskedastisitas, analisis dilanjutkan ke pemodelan GARCH.
7. Mengidentifikasi model GARCH awal menggunakan visualisasi ACF dan PACF dari residual kuadrat model ARIMA. Memilih model GARCH dengan parameter signifikan dan nilai AIC terkecil.
8. Melakukan validitas model GARCH berdasarkan beberapa asumsi, yaitu residual standar harus bebas dari heteroskedastisitas, yang dapat diuji menggunakan uji LM, serta tidak adanya autokorelasi pada residual, yang dapat diuji menggunakan uji *Ljung-Box*.
9. Menggabungkan *return* dan volatilitas saham untuk optimasi portofolio menggunakan plot *Efficient Frontier* di *Google Colaboratory* dengan pendekatan *Mean-Variance*.
10. Menghitung besarnya *return* tahunan, volatilitas tahunan, dan *Sharpe ratio* dari portofolio yang terbentuk menggunakan data saham aktualnya yang diambil dari situs *https://finance.yahoo.com/*.

2.5.2 Metode Konvensional (*Capital Asset Pricing Model*)

Tahapan analisis CAPM meliputi:
1. Menggunakan data aktual saham dari situs *https://id.investing.com/* sesuai dengan panjangnya data hasil prediksi pendekatan *machine learning.*
2. Menghitung *return* harian berdasarkan data historis harga penutupan masing-masing saham, dan pasar.
3. Menghitung *expected return* dari masing-masing saham dan pasar.
4. Menghitung *risk free rate*.
5. Menghitung *beta* masing-masing saham.
6. Menghitung *expected return* CAPM.
7. Menentukan saham yang efisien membentuk portofolio.
8. Menghitung proporsi masing-masing saham untuk portofolio menggunakan MVEP.
9. Menghitung *return* portofolio tahunan, volatilitas portofolio tahunan, dan *Sharpe ratio.*

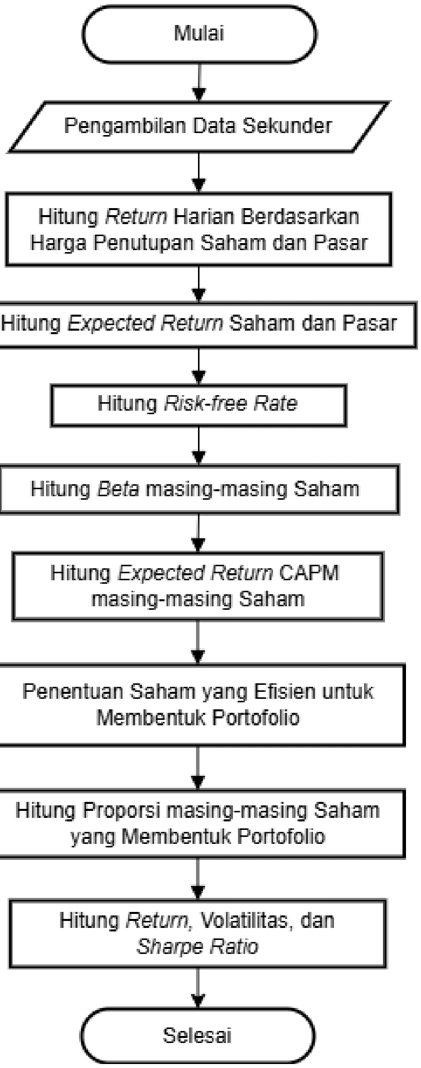
2.6 Alur Kerja

2.6.1 Pendekatan *Machine learning* (LSTM-GARCH)



Gambar 7. Alur kerja pendekatan *machine learning* LSTM-GARCH

2.6.2 Pendekatan Konvensional CAPM



Gambar 8. Alur kerja pendekatan CAPM