3. ANALISIS DAN DESAIN SISTEM

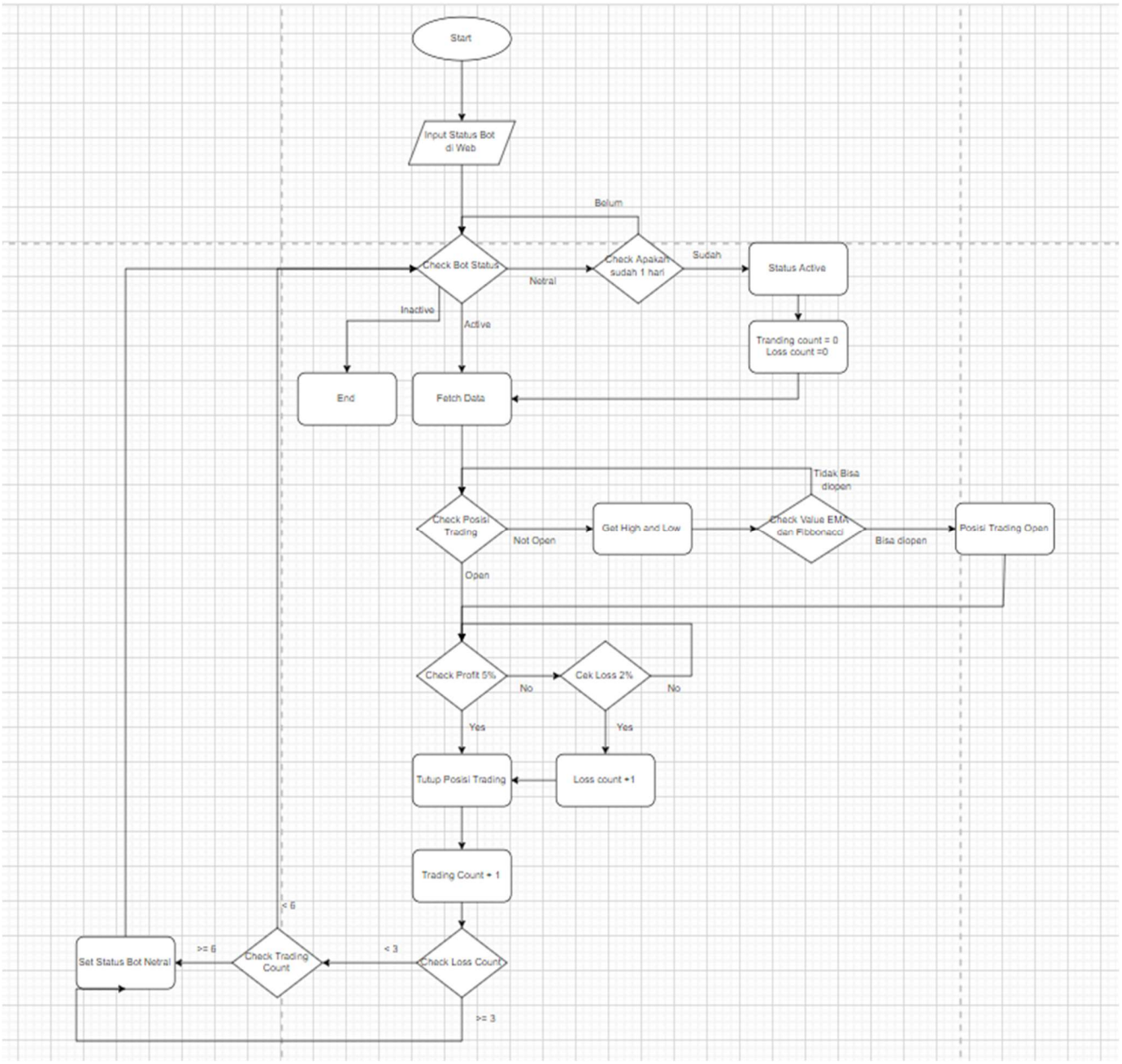
3.1. Analisis Masalah

Dalam dunia *trading*, banyak sekali permasalahan yang dialami oleh para *trader*. Permasalahan tersebut antara lain adalah ketidaktahuan trader terhadap metode yang akan digunakan untuk melakukan trading sehingga trading menjadi kurang optimal. Selain itu, ketidakmampuan *trader* dalam melakukan analisis teknikal juga merupakan suatu hal yang patut dipertimbangkan. Hal tersebut menyebabkan *trader* mengalami kerugian terus menerus dan akan berakibat pada hilangnya modal. Selain itu, banyak robot *trading* yang mengharuskan *user* nya untuk berinteraksi secara terus menerus untuk pembukaan posisi, penutupan posisi ataupun kondisi-kondisi lainnya. Hal ini menyebabkan *user* harus memantau pergerakan harga secara terus menerus dan hal tersebut akan menghabiskan banyak waktu dan tenaga.

Dalam penelitian ini, *algorithmic trading* yang dibuat bertujuan untuk mengatasi permasalahan yang telah dibahas sebelumnya. *User* tidak perlu berinteraksi secara terus menerus dengan *market* karena algoritma akan mengatur pembukaan dan penutupan posisi secara otomatis. Algoritma juga didesain dengan dasar analisis teknikal yang valid sehingga *user* tidak perlu mengetahui secara mendalam mengenai analisis teknikal untuk dapat melakukan *trading*. Selain itu, ada juga mekanisme *back testing* yang berguna untuk mengukur profitabilitas trading sekaligus menjadi pembanding terhadap metode-metode apa saja yang cocok untuk digunakan dalam gaya trading dan timeframe trading tertentu.

3.2. Desain Sistem

3.2.1 Desain *Bot Trading*



Gambar 3.1**:** *Flowchart Algorithmic Trading*

Gambar 3.1 merupakan flowchart dari cara kerja algorithmic trading mulai dari awal sampai dengan akhir. User akan menginputkan terlebih dahulu jumlah target profit dan loss per trade sesuai dengan profil risiko lalu user akan mengaktifkan robot trading. Setelah robot trading aktif, maka sistem akan melakukan pengambilan data dari market secara realtime untuk melakukan pengecekkan apakah

posisi trading sudah bisa dibuka ataupun belum. Setelah pengambilan data, maka algoritma akan melakukan perhitungan dari indikator *Fibonacci Retracement* dan *Exponential Moving Average* dengan rumus sebagai berikut:

a. *Exponential Moving Average:*

$$EMAPeriod=(ValuePeriod*(1+PSmoothing))+EMAPrev*(1-(1+PSmoothing)) \quad (2)$$

Di mana:

EMAperiod=Exponential Moving Average pada periode tertentu

EMAprev=Exponential Moving Average pada periode sebelumnya

ValuePeriod=Harga penutupan pada periode tersebut

PSmoothing=Konstanta penghalusan data pada periode tersebut

b**.** *Fibonacci Retracement:*

$$Retracement = Price\_High-((Price\ High - Price\ Low)* Fib) \quad (3)$$

Di mana:

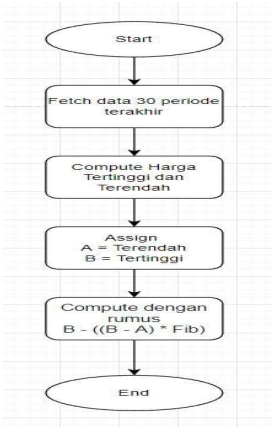
Retracement = Nilai *Retracement* dari *Fibonacci*

Price_High = Harga tertinggi selama 50 periode terakhir

Price_Low = Harga terendah selama 50 periode terakhir

Fib = Rasio *Fibonacci* yang digunakan(0.618 dan 0.5)

Untuk flowchart dari kedua rumus yaitu *Fibonacci Retracement* dan *Exponential Moving Average* dapat dilihat pada gambar 3.2.



Gambar 3.2: *Flowchart* perhitungan *Fibonacci Retracement*

Setelah perhitungan dilakukan, maka algoritma akan mengecek apakah pergerakan harga yang sekarang sudah cocok dengan syarat dari pembukaan posisi *buy* maupun *sell* seperti pada tabel 3.1.

Tabel 3.1: Syarat penentuan posisi *buy* dan/atau *sell*

POSISI	*RULE*
BUY	EMA_CLOSE 21> EMA_CLOSE 34, EMA_CLOSE21>EMA_CLOSE 50, EMA_CLOSE34>EMA_CLOSE 50, PRICE_CLOSE>EMA_CLOSE 50, PRICE_LOW<=FIB618, PRICE_CLOSE>=FIB618
SELL	EMA_CLOSE 50> EMA_CLOSE 34, EMA_CLOSE50>EMA_CLOSE 21, EMA_CLOSE34>EMA_CLOSE 21, PRICE_CLOSE<EMA_CLOSE 50, PRICE_HIGH>=FIB50, PRICE_CLOSE<=FIB50

Di mana:

EMA_Close = Nilai *EMA* dari harga penutupan selama *n* periode.

Price_Low = Harga terendah selama *n* periode.

Price_Close = Harga penutupan pada *n* periode.

Price_High = Harga tertinggi selama n periode.

Fib618 = Nilai Rasio *Fibonacci* 0.618(61,8%)

Fib50 = Nilai Rasio *Fibonacci* 0.5(50%)

EMA 21 dan 34 digunakan karena merupakan Fibonacci Based EMA, di mana periode nya merupakan nilai dari deret Fibonacci, sedangkan EMA 50 merupakan salah satu EMA periode menengah yang dapat digunakan untuk menentukan tren harga dalam jangka waktu menengah(Quiroga, 2019). Untuk rasio Fibonacci 0.618 digunakan karena rasio tersebut diyakini merupakan level support yang kuat(turning point) di mana pada saat harga mengalami koreksi dan memantul dari level tersebut, harga

diyakini masih mengikuti tren naik(Tika, 2023). Harga penutupan yang lebih besar dair EMA 50 juga mengindikasikan bahwa harga belum mengalami breakdown dari support dinamisnya, maka dari itu tren bullish masih valid. Sedangkan pada saat downtrend, harga harus berada di bawah EMA 50 untuk menandakan bahwa trend bearish valid(Quiroga, 2019).

Setelah posisi trading sudah terbuka, maka algoritma akan melakukan pengecekkan apakah posisi trading sudah memenuhi syarat take profit ataupun stop loss di tabel 3.2:

Tabel 3.2: Implementasi algoritma *take profit* dan *stop loss*

STATUS	RULE
TAKE PROFIT	-IF POSITION_OPEN = (POSITION_OPEN + (X/100)) Then Position = Close -Number of Trade++
STOP LOSS	-IF POSITION_OPEN = (POSITION_OPEN - (X/100)) Then Position = Close -Stoploss Count++ -Number of Trade++
FLOATING LOSS/PROFIT	(Tidak melakukan apa-apa)

Di mana:

POSITION_OPEN = Harga pada saat pembukaan posisi(Entry price)

X = Konstanta *persentase* kenaikan/penuruan harga yang diinputkan oleh user.

Stoploss Count: Jumlah total *stoploss*

Number of Trade: Jumlah total *trade*

Pada saat harga mengalami kenaikan sebesar *x%* pada saat posisi *trading* sudah terbuka, maka posisi *trading* akan ditutup sebagai kondisi trading yang profit. Jika harga mengalami penurunan sebesar *x%*(ditentukan oleh *user* di *interface*), maka posisi *trading* akan ditutup sebagai kondisi merugi. Setelah posisi *trading* sudah ditutup, maka jumlah *trading* akan ditambahkan sebanyak 1 *trade* dan jika posisi *trading* yang ditutup adalah rugi, maka jumlah trading akan ditambahkan sebanyak 1 *trade* dan jumlah *stop loss* akan ditambahkan sebanyak 1 *loss*.

Limit loss dan limit profit ditentukan langsung oleh user karena setiap user memiliki preferensi yang berbeda mengenai rasio keuntungan dibandingkan dengan kerugian pada saat melakukan trading. Namun pada umumnya, rasio keuntungan dibanding kerugian yang wajar pada saat melakukan trading adalah 3:1, di mana minimal keuntungan adalah 3x dari nilai kerugian yang telah ditentukan(Hayes, 2023).

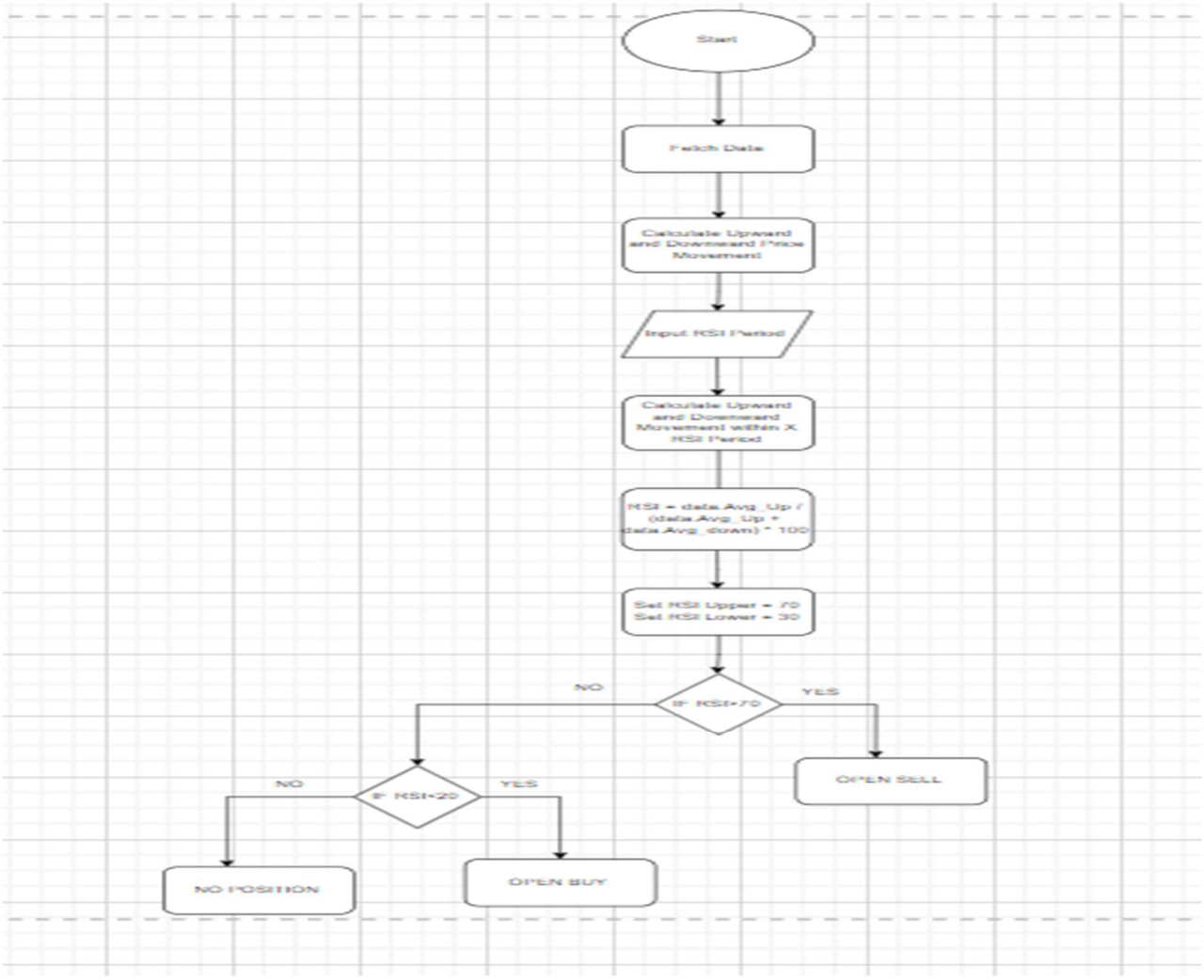
Adapun *rule* lain-lain dapat dilihat di tabel 3.3:

Tabel 3.3: Implementasi *rule* tambahan

RULE TAMBAHAN	1) Jika jumlah *trading* pada hari yang sama telah mencapai total 6x *trading*, maka algoritma akan berhenti melakukan trading dan *trading* akan dilanjutkan pada hari setelahnya dengan perhitungan jumlah *trading* yang dimulai dari 0. 2) Mengacu pada algoritma *stop loss*, jika posisi trading terkena *stop loss* sebanyak 3x, maka algoritma akan berhenti melakukan *trading* dan *trading* akan dilanjutkan pada hari setelahnya dengan perhitungan jumlah *trading* yang dimulai dari 0. 3) *Money management berupa pengaturan stop loss dan take profit* dapat ditentukan pada interface yang telah disediakan 4) Selain daripada kondisi *entry*, *take profit* dan *stop loss* yang telah ditetapkan, maka posisi *trading* adalah netral(*no position*).

Rule tambahan di tabel 3.3 berfungsi agar melindungi trader dari jumlah trading yang berlebihan sekaligus melindungi trader dari kerugian trading yang terus menerus. Pada saat trader tidak memiliki mekanisme pengaman yang kuat saat melakukan trading, seperti money management yang salah maupun rasio keuntungan/kerugian yang tidak sesuai, maka trader akan dengan mudah sekali kehilangan modalnya terutama di market yang volatilitasnya tinggi(Chen, 2022).

3.2.2 Desain algoritma *Backtester dan Forwardtester RSI*



Gambar 3.3: *Flowchart backtester* metode *Relative Strength Index*

Gambar 3.3 merupakan *flowchart* dari *backtester maupun forwardtester trading* dengan menggunakan metode *RSI*. Hal pertama yang dilakukan adalah *fetch data* berupa harga(*open, high, low* dan *close*) masa lampau *Bitcoin* dari *website* resmi *Binance*. Setelah pengambilan data, maka algoritma

akan menghitung terlebih dahulu periode kenaikan maupun penurunan harga selama periode yang telah ditentukan. Setelah itu, data dan pergerakan harga yang telah diperoleh akan dihitung dengan menggunakan rumus *RSI* sehingga akan dihasilkan output berupa pembukaan dan/atau penutupan pada posisi *trading* dapat dilihat pada tabel 3.4 berikut ini:

Tabel 3.4: Implementasi algoritma *Relative Strength Index*

POSISI	RULE
BUY	RSI < 20
SELL	RSI > 70
NEUTRAL	RSI > 20 , RSI < 70

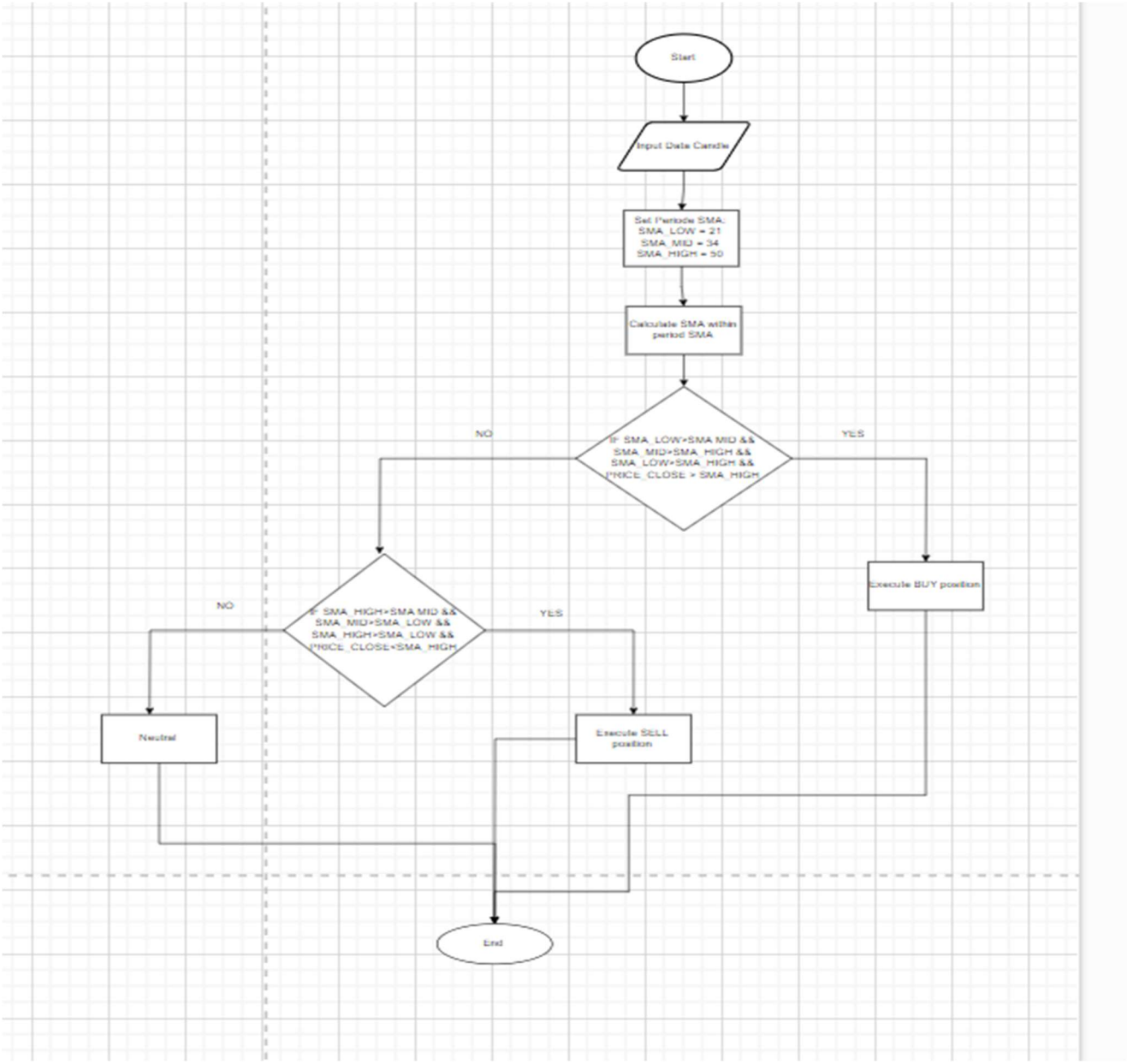
Di mana:

RSI = Nilai *Relative Strength Index*

RSI sendiri merupakan oscilator di mana nilai nya bergerak di antara 0-100, berbeda dengan indikator trend(leading) seperti Moving Average ataupun chart pattern. Namun, pada penelitian ini, RSI dipilih untuk membandingkan apakah instrumen BTC/USDT Perpetual lebih cocok untuk diperdagangkan menggunakan indikator trend seperti Moving Average atau menggunakan oscilator(leading indicator) seperti RSI. Periode RSI yang digunakan oleh para trader adalah periode 14 dengan batas overbought 20 dan batas oversold 70(Fernando, 2023).

Adapun untuk melakukan *forward testing* dari metode *Relative Strength Index*, dapat digunakan fitur yang telah disediakan oleh *Binance* yakni *Binance Testnet* untuk melakukan *trading* secara *virtual*.

3.2.2 Desain algoritma *Backtester dan Forwardtester SMA crossover*



Gambar 3.4: *Flowchart backtester* metode *SMA Crossover*

Gambar 3.4 merupakan flowchart dari *backtester* trading dengan menggunakan metode *SMA crossover*. *SMA crossover* merupakan strategi di mana pembukaan posisi dilakukan pada saat sedang terjadi posisi golden cross/death cross. Golden cross merupakan kondisi di mana nilai *SMA* dengan periode yang lebih rendah berada di atas nilai *SMA* dengan periode yang lebih tinggi. Sedangkan untuk *death cross*, nilai *SMA* dengan periode lebih rendah berada di bawah nilai *SMA* dengan periode lebih

tinggi. Hal pertama yang dilakukan adalah *fetch data* berupa harga(*open*, *high*, *low* dan *close*) masa lampau *Bitcoin* dari *website* resmi *Binance*. Setelah pengambilan data, maka algoritma akan menghitung terlebih dahulu nilai *SMA_CLOSE*, *SMA_MID* dan *SMA_LONG* dengan menggunakan rumus *SMA*. Untuk syarat dari pembukaan, penutupan dan posisi netral adalah seperti pada tabel 3.5:

Tabel 3.5: Implementasi algoritma *SMA crossover*

POSISI	RULE
BUY	IF SMA_SHORT>SMA_MID AND SMA_MID>SMA_LONG AND SMA_SHORT>SMA_LONG AND IF PRICE_CLOSE > SMA 50
SELL	IF SMA_LONG>SMA_MID AND SMA_MID>SMA_SHORT AND SMA_LONG >SMA_SHORT AND IF PRICE_CLOSE < SMA 50
NETRAL	-

Di mana:

SMA_SHORT = *SMA* dengan periode terkecil

SMA_MID = *SMA* dengan periode menengah(nilai kedua yang terkecil)

SMA_LONG = *SMA* dengan periode terbesar

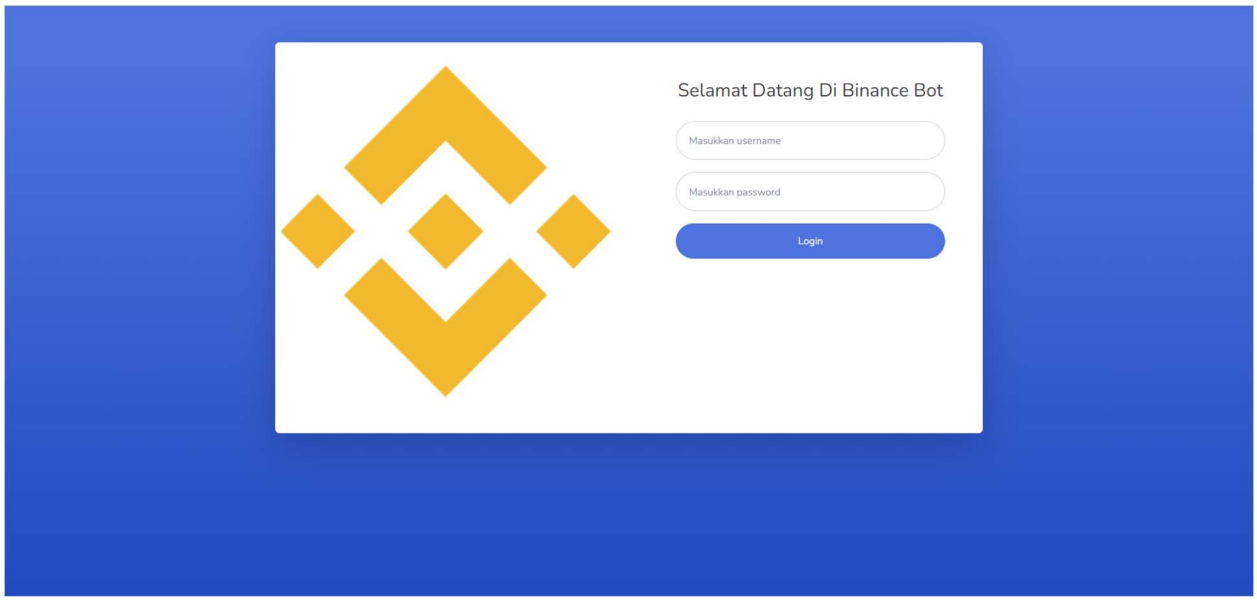
Indikator SMA merupakan indikator trend yang memiliki konsep yang kurang lebih sama dengan EMA, hanya saja SMA mengambil rata-rata harga secara rata sedangkan EMA mengambil rata-rata data berdasarkan data-data yang terkini. Namun terkadang terjadi beberapa kali false signal pada indikator EMA. Maka dari itu, SMA dapat mengatasi masalah tersebut karena reaksinya lebih lambat daripada EMA sehingga tidak bereaksi terlalu cepat terhadap false signal.

Periode SMA yang digunakan juga merupakan nilai dari deret fibonacci yaitu 21 dan 34. Selain itu, SMA 50 yang digunakan juga merupakan Moving Average dengan periode menengah untuk mendeteksi pergerakan harga dalam jangka menengah.

Adapun untuk melakukan *forward testing* dari metode *Simple Moving Average crossover*, dapat digunakan fitur yang telah disediakan oleh *Binance* yakni *Binance Testnet* untuk melakukan *trading* secara *virtual*.

3.3. Desain *Interface*

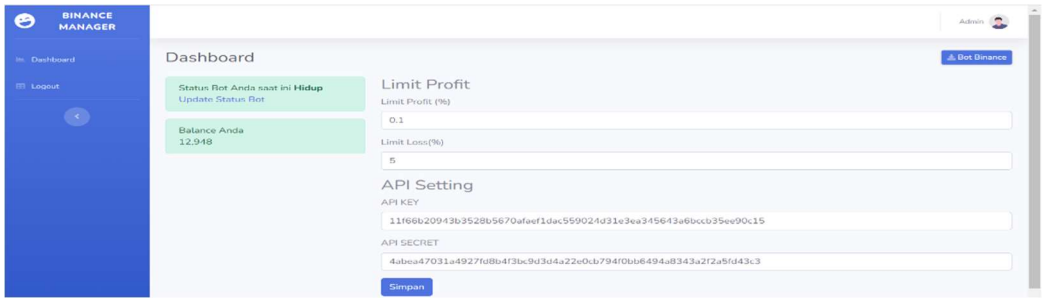
Desain *interface* dari *algorithmic trading* yang dibuat ditunjukkan pada gambar 3. Gambar 3.3 merupakan desain halaman *login dari algorithmic trading* yang akan digunakan. *User* dapat memasukkan username dan password untuk mulai melakukan *trading* dan kemudian *user* akan diarahkan pada halaman berikutnya.



Gambar 3.5: Tampilan halaman *login website algorithmic trading*

Lalu setelah masuk ke halaman *berikutnya*, *user* akan diberikan pilihan untuk melakukan pengaturan terhadap *stoploss* dan take profit yang nantinya akan digunakan dalam *trading* seperti pada gambar 3.6. Hal ini bertujuan agar *trading* dilakukan sesuai dengan profil risiko dari setiap *user*. Pada saat status *bot* belum aktif, maka *user* diminta untuk mengaktifkan terlebih dahulu dengan menekan button

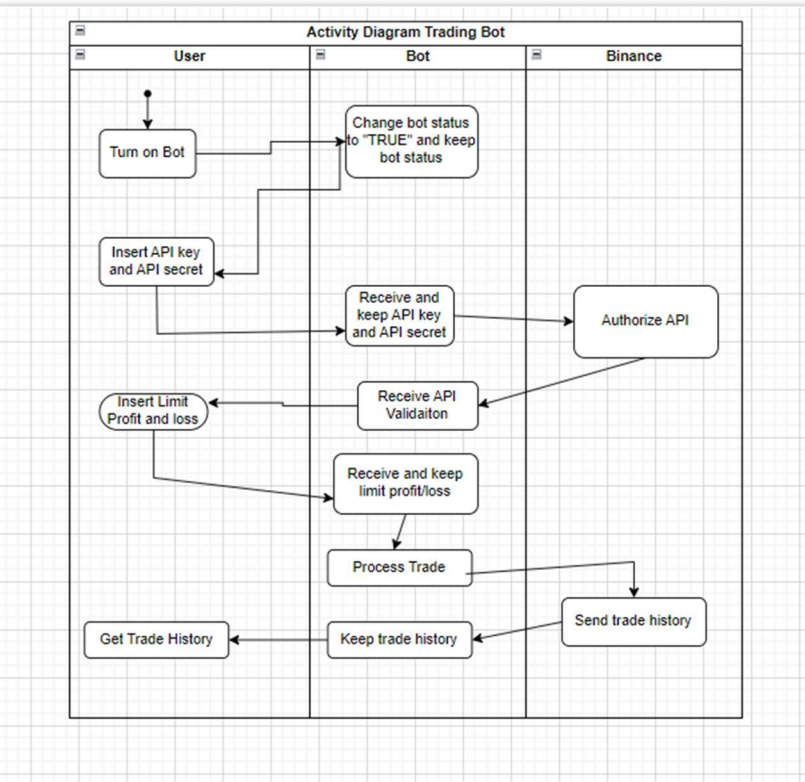
"*Update Status Bot*" dan setelah itu, algoritma akan berjalan sesuai dengan seperangkat peraturan yang telah ditentukan pada awal pembuatan algoritma. Selain itu, user juga diminta untuk menginputkan API Key dan API secret yang akan digunakan untuk melakukan *algorithmic trading* di Binance Futures.



Gambar 3.6: Tampilan halaman utama dari *website algorithmic trading*

3.4 Activity Diagram

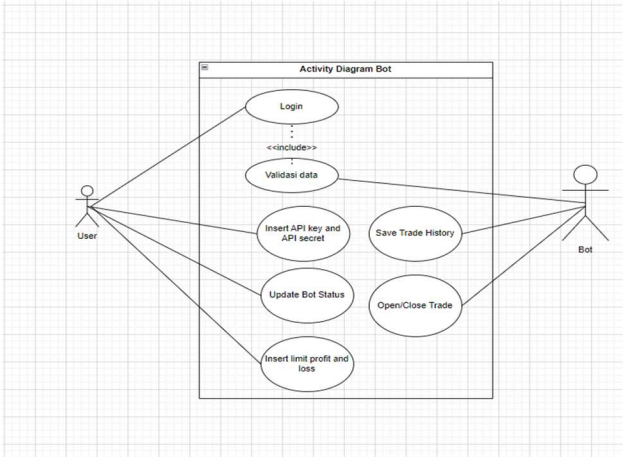
Gambar 3.7 merupakan activity diagram dari trading bot yang akan digunakan. Pertama-tama, user akan menghidupkan bot terlebih dahulu dan kemudian memasukkan API key dan API secret yang diekstrak dari website Binance Futures Testnet. Setelah memasukkan API, Binance akan melakukan validasi terlebih dahulu apakah API yang dimasukkan valid atau tidak. Jika valid, maka Binance akan mengirim pesan kepada bot bahwa API valid dan proses dapat dilanjutkan. Setelah memasukkan API, maka user akan diminta memasukkan limit loss dan profit yang kemudian akan diproses oleh database dari bot trading. Setelah seluruh informasi telah diperoleh, maka bot akan mulai memproses trading sesuai dengan ketentuan-ketentuan yang berlaku. Pada saat trade sudah selesai, maka Binance akan mengirim hasil trade ke database bot yang dapat dilihat oleh user.



Gambar 3.7: Activity Diagram dari trading bot

3.5 Use Case Diagram

Gambar 3.8 menunjukkan use case diagram dari trading bot yang akan dibuat. Fitur-fitur yang dapat diperoleh user antara lain adalah memasukkan API key dan API Secret, menghidupkan/mematikan bot dan menentukan limit profit dan loss per trade nya. Sedangkan bot akan memproses(membuka/menutup dan mencari posisi) trade dan menyimpan history trade.



Gambar 3.8: *Use Case Diagram trading bot*