PSPEC PSEUDOCODE

Nama Modul: Backend Flask – RaidAnalyst

Fungsi Utama: Menyediakan API penerimaan dan penyajian data sinyal trading dari EA ke website.

VARIABLES:

- db_connection: objek koneksi ke database MySQL
- jwt_secret: string, kunci rahasia JWT
- signal_data: dictionary
    - pair: string
    - timeframe: string
    - type: string (BUY/SELL)
    - buy_stop: float
    - stop_loss: float
    - take_profit: float
    - timestamp: datetime
- api_token: string (JWT yang dikirim EA)
- app: instance Flask

INITIALIZATION:

- Import library Flask, JWT, dan konektor MySQL
- Inisialisasi aplikasi Flask:
  app = Flask(__name__)
- Inisialisasi koneksi database
- Tentukan kunci JWT:
  jwt_secret = "supersecretkey"
- Daftarkan route utama API:
    - /api/mt5/signal (POST)
    - /api/signal/list (GET)

FUNGSI UTAMA:

1. AUTHENTICATION CHECK

- Terima header Authorization dari request.
- Ekstrak token JWT.
- Verifikasi menggunakan jwt.decode(token, jwt_secret).
- Jika token tidak valid → kembalikan response { "error": "Unauthorized" }.

2. RECEIVE SIGNAL (POST /api/mt5/signal)

Input (dikirim dari EA dalam format JSON):

```
{
 "pair": "XAUUSD",
 "timeframe": "M15",
 "type": "BUY",
 "buy_stop": 2345.50,
 "stop_loss": 2330.00,
 "take_profit": 2370.00,
 "timestamp": "2025-10-08 12:30:00"
}
```

Proses:

- Baca dan parsing JSON input.
- Validasi semua field terisi.
- Simpan ke tabel signals dalam database.
- Return status sukses.

Pseudocode:

```
ROUTE POST /api/mt5/signal:
   token = ambil_header("Authorization")
   VERIFY_TOKEN(token)
   data = ambil_json_body()
   VALIDATE(data)
   db.insert("signals", data)
   return { "status": "success", "message": "Signal saved" }
```

3. RETRIEVE SIGNALS (GET /api/signal/list)

Proses:

- Terima permintaan dari frontend.
- Query data terbaru dari database:
  SELECT * FROM signals ORDER BY timestamp DESC LIMIT 50;
- Kembalikan data dalam format JSON.

Pseudocode:

```
ROUTE GET /api/signal/list:
    results = db.query("SELECT * FROM signals ORDER BY timestamp DESC LIMIT 50")
    return jsonify(results)
```

4. DELETE OLD SIGNALS (opsional)

Membersihkan data sinyal lama untuk efisiensi.

```
ROUTE DELETE /api/signal/cleanup:
    db.query("DELETE FROM signals WHERE timestamp < NOW() - INTERVAL 30 DAY")
    return { "status": "cleaned" }
```

HANDLING ERROR

```
TRY:
    proses_request()
EXCEPT DatabaseError:
    return { "error": "Database connection failed" }
EXCEPT JWTError:
    return { "error": "Invalid token" }
EXCEPT Exception:
    return { "error": "Unexpected server error" }
```

OUTPUT

- **POST /api/mt5/signal** → { "status": "success", "message": "Signal saved" }
- **GET /api/signal/list** → daftar sinyal terbaru (JSON)
- **DELETE /api/signal/cleanup** → { "status": "cleaned" }

FLOW DIAGRAM (deskripsi teks)

1. EA (Expert Advisor) menghasilkan sinyal → kirim ke endpoint Flask /api/mt5/signal dengan JWT.
2. Flask Backend menerima request, memverifikasi token, menyimpan data ke MySQL.
3. Website frontend memanggil endpoint /api/signal/list untuk menampilkan sinyal terbaru.
4. Admin dapat membersihkan sinyal lama dengan endpoint /api/signal/cleanup.