

API Documentation

1. Get profit or loss of stock transaction (Abd. Rahman B. Tasrid)

Request

Method	URL
POST	/profit-loss

Request Body

Field	Data type	Description	Sample value	Mandatory
codeStock	String	code of stock	ADMR	yes
buyPrice	number	buy price / share	1000	yes
sellPrice	number	sell price / share	1020	yes
lot	number	quantity of transaction stock (1 lot = 100 shares)	100	yes

Sample Request

```
{
  "codeStock": "ADMR",
  "buyPrice": 1000,
  "sellPrice": 1020,
  "lot": 100
}
```

Response Body

Field	Data type	Description	Sample value
status	String	status of request (success / failed)	success
data			
stock	String	code of stock	ADMR
lot	number	quantity of transaction stock (1 lot = 100 shares)	100

	buyPrice		number	buy price of stock / share	1000
	netBuyAmount				
	buyAmount		number	amount of buy stock ($\text{buyPrice} * \text{lot} * 100$)	10000000
	totalFee		number	Fee of purchased = $0.15\% * \text{buyAmount}$	15000
	netAmount		number	net amount of purchased = $\text{buyAmount} + \text{totalFee}$	10015000
	sellPrice		number	sell price of stock / share	1020
	netSellAmount				
	sellAmount		number	amount of sell stock ($\text{sellPrice} * \text{lot} * 100$)	10200000
	totalFee		number	fee of selling = $0.25\% * \text{sellAmount}$	25500
	netAmount		number	net amount of selling = $\text{sellAmount} - \text{totalFee}$	10174500
	status		String	Status of trading (gain or loss)	Gain
	realized		number	realized of gain or loss ($\text{netSellAmount} - \text{netBuyAmount}$)	159500
	percentRealized		String	percent of realized = $(\text{realized} : \text{netBuyAmount}) * 100\%$	1.59%

Response Success

```
{
  "status": "success",
  "data": {
    "stock": "ADMR",
    "lot": 100,
    "buyPrice": 1000,
    "netBuyAmount": {
      "buyAmount": 10000000,
      "totalFee": 15000,
      "netAmount": 10015000
    },
    "sellPrice": 1020,
    "netSellAmount": {
      "sellAmount": 10200000,
      "totalFee": 25500,
      "netAmount": 10174500
    },
    "status": "Gain",
    "realized": 159500,
    "percentRealized": "1.59%"
  }
}
```

2. Ticketing Bioskop (Erfia Nadia Safari)

Link Documentation Postman :

<https://documenter.getpostman.com/view/38353155/2sAXqp7iRX>

1. Get Films

Request

Method	URL
GET	{{base_url}}/bioskop/films

Response Success

```
[
  {
    "id": 1,
    "title": "Exhuma"
  },
  {
    "id": 2,
    "title": "Kimi no Nawa"
  },
]
```

```
{
  "id": 3,
  "title": "Haikyu!"
}
```

2. Get Days

Request

Method	URL
GET	{{base_url}}/bioskop/days

Response Success

```
[
  "Senin",
  "Selasa",
  "Rabu",
  "Kamis",
  "Jumat",
  "Sabtu",
  "Minggu"
]
```

3. Get Times

Request

Method	URL
GET	{{base_url}}/bioskop/times

Response Success

```
[
  {
    "id": 1,
    "time": "11.00 - 13.15"
  },
  {
    "id": 2,
    "time": "13.30 - 15.45"
  },
  {
    "id": 3,
```

```
        "time": "16.00 - 18.15"
    }
]
```

4. Get Seat Types Request

Method	URL
GET	{{base_url}}/bioskop/seat-types

Response Success

```
[
  {
    "id": 1,
    "type": "Reguler"
  },
  {
    "id": 2,
    "type": "Sweetbox"
  }
]
```

5. Get Auditorium Request

Method	URL
GET	{{base_url}}/bioskop/auditorium

Response Success

```
[
  {
    "id": 1,
    "name": "Audi 1"
  },
  {
    "id": 2,
    "name": "Audi 2"
  },
  {
    "id": 3,
    "name": "Audi 3"
  }
]
```

```
}
]
```

6. Save Book

Request

Method	URL
POST	{{base_url}}/bioskop/book

Request Body

Field	Data type	Description	Sample value	Mandatory
filmId	Int	variant of film	1	yes
dayIndex	Int	0=Senin, 6=Minggu	0	yes
timeId	Int	Id for Schedule of Film	1	yes
audId	Int	Id for Cinema Venue	1	yes
seatTypeId	Int	Id for Seat of Viewer	1	yes

Response Success

```
{
  "message": "Tiket berhasil dipesan!",
  "tiket": {
    "film": "Exhuma",
    "hari": "Senin",
    "waktu": "11.00 - 13.15",
    "kursi": "Reguler"
  }
}
```

Response Error 400

```
{
  "message": "Input tidak valid!"
}
```

Response Error 400

```
{
  "message": "Data tidak valid!"
}
```

3. Cek Tagihan Listrik Prepaid (Alia Jennifer Kim Ritzky)

API ini memiliki fungsi untuk mengecek harga token listrik di lima merchant terkenal, yaitu gopay, grab merchant, jakone mobile, ovo, shoepay dengan menghitung berdasarkan pilihan nominal harga token listrik yang diinginkan dan merchant apa yang dipilih. Alasan API ini dibuat untuk memudahkan mengecek harga token tanpa harus membuka aplikasi atau menginstall banyak aplikasi sekaligus untuk membandingkan harga token yang paling murah dan terjangkau.

1. Get katalog request

Method	URL
GET	{{base_url}}/tagihan-listrik-prabayar

Response Success

```
{
  "message": "Selamat datang di cek tagihan listrik prabayar ^_^",
  "nominal": [
    20000,
    50000,
    100000,
    200000,
    500000,
    1000000,
    5000000
  ],
  "toko": [
    {
      "nama": "gopay"
    },
    {
      "nama": "grab-merchant"
    },
    {
      "nama": "jakone-mobile"
    },
    {
      "nama": "ovo"
    }
  ],
}
```

```

    {
      "nama": "shopeepay"
    }
  ]
}

```

2. Get nominal dan toko request

Method	URL
GET	{{base_url}}/tagihan-listrik-prabayar?nominal=20000&toko=shopeepay

Request body

Key	Tipe Data	Deskripsi	Value
nominal	int	nominal token yang ingin dibeli	20000
toko	string	nama toko untuk topup tagihan	shopeepay

Response Success

```

{
  "nominal": 20000,
  "toko": "shopeepay",
  "totalTagihan": 22750
}

```

Response Error 404 (nama toko salah)

```

{
  "message": "Toko yang diinputkan tidak sesuai. Silahkan ketikkan ulang."
}

```

Response Error 404 (jumlah nominal salah)

```

{
  "message": "Nominal tersebut tidak tercantum pada katalog. Silahkan ketikkan lagi."
}

```


Kalkulator Keuangan (Muhammad Hanif)

Base URLs: <http://localhost:3000>

API Keuangan

GET Calculator List

GET /calculate

Response Examples

```
{
  "calculators": [
    {
      "id": "investment",
      "name": "Kalkulator Investasi"
    },
    {
      "id": "emergencyfund",
      "name": "Kalkulator Dana Darurat"
    }
  ]
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	<u>OK</u>	none	Inline

POST Investment Calculator

POST /calculate/investment

Body Parameters

```
{
  "targetAmount": 1000000000,
  "years": 10,
  "currentAmount": 1000,
  "savingsFrequency": "SetiapBulan",
  "savingsTiming": "AwalBulan",
  "monthlyTarget": 5000,
  "annualReturn": 10
}
```

Params

Name	Location	Type	Required	Title	Description
body	body	object	no		none
» targetAmount	body	integer	yes		none
» years	body	integer	yes		none
» currentAmount	body	integer	yes		none
» savingsFrequency	body	string	yes	Contoh List Data Yang Diperlukan	none
» savingsTiming	body	string	yes	Sesuaikan Dengan Setiap Bulan Atau SetiapTahun Tersedia SetiapTahun SetiapBulan	Sebagai Contoh Jika Setiap Tahun Maka Memilih Awal Tahun atau AkhirTahun
» monthlyTarget	body	integer	yes		none
» annualReturn	body	integer	yes		none

Response Examples

```
{
  "achievedTarget": true,
  "finalAmount": 101116865,
  "periodsNeeded": 74,
  "periodType": "bulan",
  "message": "Selamat! Strategi investasi Anda berhasil mencapai target d
```

```
alam 74 bulan."
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	<u>OK</u>	none	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» achievedTarget	boolean	true	none		none
» finalAmount	integer	true	none		none
» periodsNeeded	integer	true	none		none
» periodType	string	true	none		none
» message	string	true	none		none

POST Emergency Fund

POST /calculate/emergencyfund

Body Parameters

```
{
  "monthlyExpenses": 5000000,
  "emergencyFundMonths": 6,
  "currentSavings": 10000000,
  "monthlySavings": 2000000,
  "returnOnInvestment": 5,
  "targetMonthsToSave": 12
}
```

Params

Name	Location	Type	Required	Title	Description
body	body	object	no		none
» targetAmount	body	integer	yes		none
» years	body	integer	yes		none

» currentAmount	body	integer	yes		none
» savingsFrequency	body	string	yes		none
» savingsTiming	body	string	yes		none
» monthlyTarget	body	integer	yes		none
» annualReturn	body	integer	yes		none

Response Examples

200 Response

```
{
  "totalEmergencyFund": 45000000,
  "remainingEmergencyFund": 35000000,
  "periodsNeeded": 12,
  "achievedTarget": false,
  "investmentResult": 35171653.75781273,
  "totalInvestment": 35171653.75781273,
  "investmentRecommendation": {
    "allocation": {
      "savings": 60,
      "investment": 40
    },
    "investmentOptions": [
      "Deposito",
      "Emas",
      "Reksadana Pasar Uang"
    ]
  },
  "message": "Maaf, Anda belum bisa mencapai target dana darurat sebesar Rp 45.000.000 dalam waktu 12 bulan. Dibutuhkan sekitar 12 bulan untuk mencapai target. Total investasi yang berhasil terkumpul adalah Rp 35.171.653,758."
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	<u>OK</u>	none	Inline