FishCastAI API Documentation

Base URL

`http://localhost:8000/api/`

Authentication

Saat ini API menggunakan `AllowAny` permission, jadi tidak memerlukan authentication untuk development.

Endpoints

1. Health Check

GET `/api/health/`

Check status API.

Response:

```
{
    "status": "healthy",
    "message": "FishCastAI API is running"
}
```

2. Dataset Management

List All Datasets **GET** `/api/datasets/`

Response:

```
[
    {
        "id": 1,
        "name": "Sample Dataset",
        "file": "/media/datasets/sample.csv",
        "uploaded_at": "2025-08-01T04:21:59Z",
        "processed_data": {
            "columns": ["stok_ikan", "bulan_normalized"],
            "shape": [100, 2],
            "sample_data": [...]
        },
        "description": "Sample fish data"
    }
]
```

Upload Dataset **POST** `/api/datasets/`

Form Data: - `name` (string): Nama dataset - `file` (file): File CSV - `description` (string, optional): Deskripsi dataset

Response:

```
{
    "id": 1,
    "name": "Sample Dataset",
    "file": "/media/datasets/sample.csv",
    "uploaded_at": "2025-08-01T04:21:59Z",
    "processed_data": {...},
    "description": "Sample fish data"
}
```

Get Specific Dataset **GET** `/api/datasets/{id}/`

Response: Same as upload response

Delete Dataset **DELETE** `/api/datasets/{id}/`

Response: 204 No Content

3. Prediction

Run Predictions **POST** `/api/predict/`

Request Body:

```
{
    "dataset_id": 1,
    "models": ["Linear", "GRU", "LSTM", "BiLSTM"]
}
```

Response:

```
{
    "message": "Predictions completed successfully",
    "results": [
        {
            "id": 1,
            "dataset": 1,
            "dataset_name": "Sample Dataset",
            "model_type": "Linear",
            "predictions": [1.0, 2.0, 3.0, 4.0, 5.0],
            "actual_values": [1.1, 2.1, 3.1, 4.1, 5.1],
            "mse": 0.01,
            "mae": 0.1,
            "created_at": "2025-08-01T04:21:59Z"
        }
```

```
    ]
}
```

List All Predictions **GET** `/api/predictions/`

Response:

```
[
    {
        "id": 1,
        "dataset": 1,
        "dataset_name": "Sample Dataset",
        "model_type": "Linear",
        "predictions": [...],
        "actual_values": [...],
        "mse": 0.01,
        "mae": 0.1,
        "created_at": "2025-08-01T04:21:59Z"
    }
]
```

4. Optimization

Run NSGA-III Optimization **POST** `/api/optimize/`

Request Body:

```
{
    "dataset_id": 1,
    "population_size": 40,
    "generations": 100
}
```

Response:

```
{
    "message": "Optimization completed successfully",
    "result": {
        "id": 1,
        "dataset": 1,
        "dataset_name": "Sample Dataset",
        "solutions": [[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]],
        "best_solution": [0.1, 0.2, 0.3],
        "best_total_stok": 10.0,
        "best_mse": 0.01,
        "population_size": 40,
        "generations": 100,
        "created_at": "2025-08-01T04:21:59Z"
```

```
    }
}
```

List Optimization Results **GET** `/api/optimization-results/`

Response:


```
[
    {
        "id": 1,
        "dataset": 1,
        "dataset_name": "Sample Dataset",
        "solutions": [...],
        "best_solution": [...],
        "best_total_stok": 10.0,
        "best_mse": 0.01,
        "population_size": 40,
        "generations": 100,
        "created_at": "2025-08-01T04:21:59Z"
    }
]
```


5. Correlation Analysis

Generate Correlation Analysis **POST** `/api/correlation/`

Request Body:

```
{
    "dataset_id": 1
}
```

Response:


```
{
    "message": "Correlation analysis completed successfully",
    "result": {
        "id": 1,
        "dataset": 1,
        "dataset_name": "Sample Dataset",
        "correlation_matrix": {
            "stok_ikan": {
                "stok_ikan": 1.0,
                "bulan_normalized": 0.5
            },
            "bulan_normalized": {
                "stok_ikan": 0.5,
                "bulan_normalized": 1.0
            }
```

```
        },
        "plot_base64": null,
        "created_at": "2025-08-01T04:21:59Z"
    }
}
```

List Correlation Results **GET** `/api/correlation-results/`

Response:

```
[
    {
        "id": 1,
        "dataset": 1,
        "dataset_name": "Sample Dataset",
        "correlation_matrix": {...},
        "created_at": "2025-08-01T04:21:59Z"
    }
]
```

6. Export

Export Prediction Results **GET** `/api/export/{prediction_id}/`

Download prediction results as CSV file.

Response: CSV file download

Error Responses

400 Bad Request

```
{
    "error": "Error message here"
}
```

404 Not Found

```
{
    "detail": "Not found."
}
```

Contoh Penggunaan dengan cURL

1. Health Check

```
curl http://localhost:8000/api/health/
```

2. Upload Dataset

```
curl -X POST http://localhost:8000/api/datasets/ \
  -F "name=my_dataset" \
  -F "file=@data.csv" \
  -F "description=Sample fish data"
```

3. Run Predictions

```
curl -X POST http://localhost:8000/api/predict/ \
  -H "Content-Type: application/json" \
  -d '{
    "dataset_id": 1,
    "models": ["Linear"]
  }'
```

4. Run Optimization

```
curl -X POST http://localhost:8000/api/optimize/ \
  -H "Content-Type: application/json" \
  -d '{
    "dataset_id": 1,
    "population_size": 40,
    "generations": 100
  }'
```

5. Get Correlation

```
curl -X POST http://localhost:8000/api/correlation/ \
  -H "Content-Type: application/json" \
  -d '{
    "dataset_id": 1
  }'
```

6. Export Results

```
curl -O http://localhost:8000/api/export/1/
```

Status Codes

- `200 OK`: Request berhasil
- `201 Created`: Resource berhasil dibuat
- `204 No Content`: Request berhasil, tidak ada content
- `400 Bad Request`: Request tidak valid
- `404 Not Found`: Resource tidak ditemukan
- `500 Internal Server Error`: Server error

Notes

1. **Mock Data**: Saat ini API menggunakan mock data karena ML dependencies belum terinstall
2. **File Upload**: Pastikan file CSV memiliki format yang sesuai
3. **CORS**: API sudah dikonfigurasi untuk CORS, bisa digunakan dengan frontend
4. **Development**: API berjalan di development mode dengan `DEBUG=True`