# HALAMAN JUDUL

**PREDIKSI HARGA MINYAK MENTAH MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN *BACKPROPAGATION* (JST-BP) DAN *PARTICLE SWARM OPTIMIZATION* (PSO)**

Skripsi

diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi S1 Teknik Informatika

Oleh:

Fitri Amalia Langgundi  
4611417009

**JURUSAN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS NEGERI SEMARANG**

**2021**

# PERNYATAAN

Saya menyatakan bahwa skripsi saya yang berjudul “Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO)” disusun atas dasar penelitian saya dengan arahan dosen pembimbing. Sumber informasi atau kutipan yang berasal dari karya yang diterbitkan telah disebutkan dalam teks dan dicantumkan dalam daftar pustaka di bagian akhir skripsi ini. Saya menyatakan bahwa skripsi ini bebas plagiat dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan perundang-undangan.

Semarang, 01 Juni 2021

Fitri Amalia Langgundi

NIM 4611417009

# PERSETUJUAN PEMBIMBING

|  |  |
| --- | --- |
| Nama | : Fitri Amalia Langgundi |
| NIM | : 4611417009 |
| Program Studi | : Teknik Informatika S1 |
| Judul Skripsi | : Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO) |

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.

Semarang, 01 Juni 2021

Dosen Pembimbing

Aji Purwinarko S.Si., M.Cs.

NIP 198509102015041001

# PENGESAHAN

Skripsi dengan judul:

Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO).

disusun oleh:

Fitri Amalia Langgundi

4611417009

Telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal

Panitia:

|  |  |
| --- | --- |
| Ketua,    Dr. Sugianto, M.Si. NIP 196102191993031001 | Sekretaris,  Dr. Alamsyah, S.Si., M.Kom. NIP 197405172006041001 |
| Penguji 1,  Riza Arifudin S. Pd., M. Cs. NIP 198005252005011001 | Penguji 2,  Budi Prasetiyo S.Si., M.Kom. NIP 198805012014041001 |
| Anggota Penguji/Pembimbing  Aji Purwinarko S.Si., M.Cs.  NIP 198509102015041001 | |

# MOTTO DAN PERSEMBAHAN

**MOTTO**

“*Sesungguhnya Allah tidak merubah keadaan sesuatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri*” (QS. Ar Ra’d : 11)

“*Dan bersabarlah. Sesungguhnya Allah beserta orang-orang yang bersabar*” (Al Anfaal : 46)

“*Jangan cemas, semua ada pada takarannya masing-masing. Tetaplah melangkah walau pelan, tetaplah berusaha meski tertatih, tetaplah mencoba meski berat.*”

**PERSEMBAHAN**

Skripsi ini saya persembahkan kepada:

* Kedua Orang Tua saya Bapak Ahmad Taufik dan Ibu Sri Priyanti yang saya sayangi, kasihi, cintai dan selalu saya doakan, semoga ini bisa menghadirkan senyum di bibir dan hati kalian.
* Kedua kakak, adik dan keponakan saya yang selalu memberi dukungan.
* Teman saya Ikhsan yang telah meluangkan waktunya untuk membantu dan memberikan saya pengarahan untuk skripsi ini.
* Teman-teman saya di jurusan Ilmu Komputer, Fakultas MIPA, Universitas Negeri Semarang serta teman-teman di lingkungan maupun organisasi.
* Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu hingga terselesaikannya penulisan skripsi ini.
* Almamater, Universitas Negeri Semarang.

# PRAKATA

Puji syukur penulis panjatkan kepada *Rabb* seluruh alam semesta Allah Subhanahu *wa ta'ala*, atas berkat rahmat, pertolongan, petunjuk, bimbingan serta kebaikan-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO)**”. Shalawat serta salam tak lupa selalu terpanjatkan kepada junjungan kita makhluk paling mulia di langit dan di bumi, suri tauladan bagi kita semua, manusia yang membawa peradaban kepada zaman yang terang-benderang, *Rasulullah* *Muhammad* *Shallallahu* *alaihi* *wassalam*, keluarganya, sahabatnya dan para pengikutnya.

Penulis menyadari bahwa penulisan skripsi ini tidak akan selesai tanpa adanya dukungan serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Dr. Sugianto, M.Si., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Dr. Alamsyah, S.Si., M.Kom., Ketua Jurusan Ilmu Komputer FMIPA Universitas Negeri Semarang yang telah meluangkan waktu, membantu, membimbing, mengarahkan dan memberikan saran sehingga penulis dapat menyelesaikan skripsi ini.
4. Aji Purwinarko S.Si., M.Cs., Dosen Pembimbing saya, guru saya yang dengan penuh kesabaran telah meluangkan waktu, membantu, membimbing, mengarahkan, memberikan saran, dan membagi ilmunya sehingga penulis dapat menyelesaikan skripsi ini.
5. Riza Arifudin S. Pd., M.Cs., selaku Dosen Penguji I yang telah memberikan banyak masukan serta arahan dalam penyelesaian skripsi ini.
6. Budi Prasetiyo S.Si., M.Kom., selaku Dosen Penguji II yang telah memberikan banyak masukan serta arahan dalam penyelesaian skripsi ini.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan bekal kepada penulis dalam penyusunan skripsi ini.
8. Kedua Orang Tua saya Ibu Sri Priyanti dan Bapak Ahmad Taufik yang selalu tanpa lelah mencurahkan kasih sayang, doa, restu dan dukungannnya.
9. Anak kecil Gibran yang selalu memberikan tawanya disela-sela lelah dan kejenuhan.
10. Teman saya Ikhsan yang telah meluangkan waktunya untuk membantu dan memberikan saya pengarahan untuk skripsi ini, terima kasih.
11. Teman-teman saya di jurusan Ilmu Komputer, terutama teman-teman ilkom angkatan 2017, dan teman-teman terdekat saya yang telah memberikan bantuan, semangat dan dukungannya.
12. Semua pihak yang telah membantu terselesaikannya skripsi ini, terima kasih atas bantuannya.

Semoga skripsi ini dapat memberikan manfaat bagi pembaca di masa yang akan datang.

Semarang, 04 Mei 2021

Fitri Amalia Langgundi

NIM 4611417009

# ABSTRAK

**Langgundi, Fitri Amalia**. 2021. Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO). Skripsi. Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Aji Purwinarko S.Si., M.Cs.

Kata kunci: Prediksi, JST-BP, PSO, Minyak Mentah

Minyak mentah dunia menjadi salah satu produk minyak bumi alami paling strategis yang diperdagangkan di pasar dunia. Oleh karena itu, harga minyak mentah sangat berpengaruh pada fluktuasi harga pasar komoditas. Sebagai salah satu negara penghasil minyak dunia, Indonesia perlu memperhatikan dengan cermat lonjakan atau turunnya harga minyak demi pertumbuhan ekonomi negara. Harga minyak mentah yang akan datang dapat diprediksi dengan menggunakan data rekaman harga minyak mentah sebelumnya. Pada penelitian ini, dilakukan prediksi harga minyak mentah dunia berdasarkan *dataset Crude Oil* WTI (CL=F) yang diperoleh dari situs finance.yahoo.com. Sebanyak 1058 data yang digunakan memiliki nilai harga yang sangat bervariasi karena ketidakstabilan harga minyak mentah. Untuk menyamakan data dalam *range* tertentu dilakukannya normalisasi data dengan penskalaan [0,1]. Setelah *dataset* dinormalisasi, kemudian dilakukan pengujian parameter untuk JST-BP dan PSO. Pengujian parameter bertujuan untuk menentukan nilai parameter terbaik untuk digunakan dalam metode JST-BP – PSO sehingga metode dapat berjalan dengan optimal. Nilai parameter terbaik dari setiap pengujian digunakan untuk penentuan model jaringan JST-BP dan nilai PSO melalui proses *training* dengan menggunakan 70% *dataset*. Kemudian, 30% *dataset* digunakan untuk prediksi melalui proses *testing* dengan nilai parameter yang memiliki tingkat *error* rendah. Hasil tingkat akurasi yang diperoleh menggunakan metode JST-BP – PSO untuk prediksi harga minyak mentah yaitu sebesar 94,97993%. Sedangkan akurasi yang diperoleh pada penggunaan metode JST-BP yaitu sebesar 93,71677% yang berarti pada metode JST-BP – PSO mengalami peningkatan akurasi sebesar 1,26316%.

# DAFTAR ISI

[**HALAMAN JUDUL** i](#_Toc73457084)

[**PERNYATAAN** ii](#_Toc73457085)

[**PERSETUJUAN PEMBIMBING** iii](#_Toc73457086)

[**PENGESAHAN** iv](#_Toc73457087)

[**MOTTO DAN PERSEMBAHAN** v](#_Toc73457088)

[**PRAKATA** vi](#_Toc73457089)

[**ABSTRAK** viii](#_Toc73457090)

[**DAFTAR ISI** ix](#_Toc73457091)

[**DAFTAR TABEL** xi](#_Toc73457092)

[**DAFTAR GAMBAR** xii](#_Toc73457093)

[**DAFTAR LAMPIRAN** xiii](#_Toc73457094)

[**BAB 1 PENDAHULUAN** 1](#_Toc73457095)

[1.1. Latar Belakang 1](#_Toc73457096)

[1.2. Rumusan Masalah 3](#_Toc73457097)

[1.3. Batasan Masalah 3](#_Toc73457098)

[1.4. Tujuan Penelitian 4](#_Toc73457099)

[1.5. Manfaat Penelitian 4](#_Toc73457100)

[1.6. Sistematika Penulisan 4](#_Toc73457101)

[1.6.1. Bagian Awal Skripsi 4](#_Toc73457102)

[1.6.2. Bagian Isi Skripsi 4](#_Toc73457103)

[1.6.3. Bagian Akhir Skripsi 5](#_Toc73457104)

[**BAB 2 TINJAUAN PUSTAKA** 6](#_Toc73457105)

[2.1. Prediksi 6](#_Toc73457107)

[2.2. Jaringan Syaraf Tiruan (JST) 7](#_Toc73457108)

[2.2.1. Pengertian Jaringan Syaraf Tiruan 7](#_Toc73457109)

[2.2.2. Komponen Jaringan Syaraf Tiruan 8](#_Toc73457110)

[2.2.3. Arsitektur Jaringan Syaraf Tiruan 9](#_Toc73457111)

[2.2.4. Fungsi Aktivasi Jaringan Syaraf Tiruan 12](#_Toc73457112)

[2.3. JST-Backpropagation 15](#_Toc73457113)

[2.3.1. Backpropagation 15](#_Toc73457114)

[2.3.2. Algoritma Backpropagation 16](#_Toc73457115)

[2.3.3. Arsitektur Backpropagation 19](#_Toc73457116)

[2.4. *Particle Swarm Optimization* (PSO) 19](#_Toc73457117)

[2.5. Harga Minyak Mentah (*Crude Oil Price*) 22](#_Toc73457118)

[2.6. Penelitian Terkait 24](#_Toc73457119)

[**BAB 3 METODE PENELITIAN** 28](#_Toc73457120)

[3.1. Studi Pendahuluan 28](#_Toc73457122)

[3.2. Pengambilan Data 29](#_Toc73457123)

[3.3. Normalisasi Data 30](#_Toc73457124)

[3.4. Pengujian Parameter 31](#_Toc73457125)

[3.5. Penentuan Model 31](#_Toc73457126)

[3.6. Prediksi 36](#_Toc73457127)

[3.7. Perancangan Sistem 37](#_Toc73457128)

[3.8. Penarikan Kesimpulan 38](#_Toc73457129)

[**BAB 4 HASIL DAN PEMBAHASAN** 39](#_Toc73457130)

[4.1. Hasil Penelitian 39](#_Toc73457131)

[4.1.1. Tahap Pengambilan Data 39](#_Toc73457132)

[4.1.2. Tahap Normalisasi Data 41](#_Toc73457133)

[4.1.3. Tahap Pengujian Parameter 41](#_Toc73457134)

[4.1.4. Tahap Penentuan Model 45](#_Toc73457135)

[4.1.5. Tahap Prediksi 47](#_Toc73457136)

[4.2. Implementasi Sistem 48](#_Toc73457137)

[4.2.1. Tahap Perancangan User Interface 48](#_Toc73457138)

[4.2.2. Tahap Implementasi Algoritma 54](#_Toc73457139)

[4.3. Pembahasan 58](#_Toc73457140)

[**BAB 5 PENUTUP** 61](#_Toc73457141)

[5.1. Kesimpulan 61](#_Toc73457143)

[5.2. Saran 61](#_Toc73457144)

[**DAFTAR PUSTAKA** 62](#_Toc73457145)

[**LAMPIRAN** 67](#_Toc73457146)

# DAFTAR TABEL

[Tabel 3.1 Data Harga Minyak Mentah Dunia 29](#_Toc71017159)

[Tabel 4.1 Harga Minyak Mentah Dunia 39](#_Toc71017252)

[Tabel 4.2 Harga *Close* Minyak Mentah Dunia 40](#_Toc71017253)

[Tabel 4.3Hasil Pengujian *Input*, *Hidden Layer* dan Iterasi BP 42](#_Toc71017254)

[Tabel 4.4 Hasil Pengujian *Learning Rate* BP 43](#_Toc71017255)

[Tabel 4.5 Hasil Pengujian Jumlah Iterasi PSO 43](#_Toc71017256)

[Tabel 4.6 Iterasi Terbaik 44](#_Toc71017257)

[Tabel 4.7Hasil Nilai r1 dan r2 Terbaik 45](#_Toc71017258)

[Tabel 4.8 Parameter PSO dan JST-BP 46](#_Toc71017259)

[Tabel 4.9 Hasil MSE dan MAPE Proses *Training* 46](#_Toc71017260)

[Tabel 4.10 Hasil Prediksi 47](#_Toc71017261)

[Tabel 4.11 Kode Menampilkan *Dataset* 49](#_Toc71017262)

[Tabel 4.12 Kode Normalisasi *Dataset* 50](#_Toc71017263)

[Tabel 4.13 Kode Fungsi untuk mengambil nilai parameter 51](#_Toc71017264)

[Tabel 4.14 Kode fungsi AJAX utk mengirim, memproses dan menampilkan data 51](#_Toc71017265)

[Tabel 4.15 *Class Neuron* 55](#_Toc71017266)

[Tabel 4.16 Potongan Kode *Class Backpropagation* 55](#_Toc71017267)

[Tabel 4.17 Potongan Kode *Class BackpropagationParticle* 56](#_Toc71017268)

[Tabel 4.18 Potongan Kode *Class PSOxBackpro* 57](#_Toc71017269)

[Tabel 4.19 Hasil Tiap Proses Perhitungan 59](#_Toc71017270)

# DAFTAR GAMBAR

[Gambar 2.1 Struktur *neuron* jaringan (Sudarsono, 2016) 9](#_Toc71014107)

[Gambar 2.2 Jaringan Syaraf Tiruan Lapisan Tunggal (Revi *et al*., 2018) 10](#_Toc71014108)

[Gambar 2.3 Jaringan Syaraf Tiruan Banyak Lapisan (Revi *et al*., 2018) 11](#_Toc71014109)

[Gambar 2.4 Jaringan Syaraf Tiruan Lapisan Kompetitif (Revi *et al*., 2018) 11](#_Toc71014110)

[Gambar 2.5 Fungsi Aktivasi Undak Biner Hard Limit (Siregar, 2019) 12](#_Toc71014111)

[Gambar 2.6 Fungsi Aktivasi Undak Biner *Threshold* (Siregar, 2019) 12](#_Toc71014112)

[Gambar 2.7 Fungsi Bipolar *Symetric Hard Limit* (Siregar, 2019) 13](#_Toc71014113)

[Gambar 2.8 Fungsi Bipolar *Threshold* (Siregar, 2019) 13](#_Toc71014114)

[Gambar 2.9 Fungsi Identitas (Siregar, 2019) 13](#_Toc71014115)

[Gambar 2.10 Struktur Jaringan *Backpropagation* (Revi *et al*., 2018) 16](#_Toc71014116)

[Gambar 2.11 Arsitektur JST-*Backpropagation* (Julpan *et al*., 2015) 19](#_Toc71014117)

[Gambar 2.12 Simulasi burung yang sedang mencari makanan (Xu *et al*., 2018) 21](#_Toc71014118)

[Gambar 3.1 Tahapan Penelitian 28](#_Toc71014119)

[Gambar 3.2 *Flowchart* Penentuan Model JST-BP dan PSO 34](#_Toc71014120)

[Gambar 3.3 *Flowchart* Pelatihan JST-BP 35](#_Toc71014121)

[Gambar 3.4 *Flowchart* Pengujian JST-BP 37](#_Toc71014122)

[Gambar 4.1 Tampilan Halaman Beranda 48](#_Toc71014147)

[Gambar 4.2 Tampilan Sub Menu Dataset Asli 49](#_Toc71014148)

[Gambar 4.3 Tampilan Sub Menu Hasil Normalisasi 49](#_Toc71014149)

[Gambar 4.4 Tampilan Menu Proses JST-BP 50](#_Toc71014150)

[Gambar 4.5 Tampilan Menu Hasil *Training* JST-BP 52](#_Toc71014151)

[Gambar 4.6 Tampilan Menu Hasil *Testing* JST-BP 52](#_Toc71014152)

[Gambar 4.7 Tampilan Proses JST-BP – PSO 53](#_Toc71014153)

[Gambar 4.8 Tampilan Hasil Proses *Training* JST-BP – PSO 54](#_Toc71014154)

[Gambar 4.9 Tampilan Hasil Proses *Testing* JST-BP – PSO 54](#_Toc71014155)

[Gambar 4.10 Tampilan Halaman Tentang 54](#_Toc71014156)

# DAFTAR LAMPIRAN

[Lampiran 1. Kode Program Sistem 67](#_Toc71031270)

[Lampiran 2. Kode Program User Interface 73](#_Toc71031271)

[Lampiran 3. Kode Program Implementasi Algoritma 98](#_Toc71031272)

[Lampiran 4. Crude Oil WTI (CL=F) Dataset 109](#_Toc71031273)

# BAB 1 PENDAHULUAN

## Latar Belakang

Minyak mentah merupakan salah satu produk minyak bumi alami yang terdiri dari endapan hidrokarbon dan bahan organik lainnya yang tidak dimurnikan (Chen, 2020). Minyak mentah termasuk komoditas paling strategis diperdagangkan di pasar dunia karena sangat berperan penting dalam masyarakat, termasuk bidang ekonomi, politik, dan teknologi (Li *et al*., 2019). Minyak mentah telah diperdagangkan secara internasional oleh beberapa Negara penghasil minyak, perusahaan minyak, perorangan hingga Negara pengimpor minyak. Harga minyak mentah dipengaruhi oleh beberapa faktor diantaranya yaitu tingkat stock, pertumbuhan ekonomi, aspek politik, ketidakstabilan politik, keputusan OPEC (*Organization of the Petroleum Exporting Countries*) dan psikologis lebih lanjut dari pedagang (Bildirici & Ersin, 2015).

Dibeberapa waktu harga minyak mentah mengalami perubahan yang sangat cepat. Gejolak harga minyak ini dianggap sangat berpengaruh pada kegiatan ekonomi (Bildirici & Ersin, 2015). Naik turunnya harga minyak mentah berpengaruh pada fluktuasi harga pasar komoditas sehingga setiap kenaikan atau penurunan harga minyak mentah secara tiba-tiba menyebabkan perlambatan ekonomi dan komoditas lainnya. Kebutuhan energy di dunia hingga kini masih sangat bergantung pada bahan bakar fosil terutama minyak bumi, hal ini menjadi penyebab harga minyak dunia sangat penting dalam sektor perdagangan, mengingat persebaran cadangan minyak di dunia yang tidak merata (Putra, 2016). Oleh karena itu, kemampuan prediksi harga minyak mentah merupakan hal yang berguna dalam pengelolaan sebagian besar sektor industri di seluruh dunia (Bildirici & Ersin, 2015). Karena hal ini sangat penting bagi Indonesia yang menjadi salah satu negara penghasil minyak dunia untuk memperoleh keuntungan berlipat ganda dari ekspor minyak saat harga minyak dunia mengalami kenaikan, sehingga mampu meningkatkan pertumbuhan ekonomi.

Data minyak mentah dunia merupakan salah satu data runtun waktu (*time series*) yang diperoleh dari masa lalu melalui situs *internet* dan dapat digunakan untuk memprediksi data di masa mendatang. Dikatakan data runtun waktu karena himpunan observasi data terurut dalam suatu waktu tertentu (Wulandari *et al*., 2017), seperti *dataset* harga minyak mentah yang beruntun waktu harian. Namun demikian, menurut Fazelabdolabadi (2019) melakukan prediksi harga minyak mentah menjadi tugas yang menantang karena prilakunya dianggap sebagai akibat dari beberapa faktor yang ada di pasar minyak mentah dunia.

Prediksi dapat dilakukan dengan berbagai cara dan alat yang digunakan, salah satunya yaitu dengan teknologi komputer. Prediksi adalah memperkirakan sesuatu yang akan terjadi di masa mendatang dengan mempelajari pola data yang ada sebelumnya (Aulia, 2018). Salah satu metode yang sering digunakan untuk melakukan suatu prediksi yaitu metode Jaringan Syaraf Tiruan (JST). JST merupakan suatu metode komputasi yang mencoba untuk mereplikasi cara kerja jaringan syaraf (Salsabila & Cholissodin, 2020). *Backpropagation* merupakan salah satu metode dalam JST yang merupakan algoritma pembelajaran yang terawasi (*supervised*) dan sering digunakan untuk melakukan prediksi dari suatu data *time series*.

Ramyar & Kianfar (2017) berhasil menunjukkan bahwa model JST degan arsitektur *multi layer perceptron* lebih akurat memprediksi harga minyak mentah dibandingkan dengan model *Vector Autoregressive*. Di dalam penelitian yang dilakukan oleh Alkronz *et al*. (2019), Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) juga telah berhasil digunakan dalam peramalan penentuan jamur beracun atau tidak dengan menghasilkan nilai akurasi sebesar 99.25% pada arsitektur *Multi Layer Perceptron* berisikan satu *input layer*, tiga *hidden layer*, dan satu *output layer*. Penentuan parameter yang digunakan dalam peramalan JST-BP sangatlah berpengaruh terhadap kinerja metode JST-BP. Sehingga diperlukannya algoritma yang digunakan untuk mengoptimasikan penentuan parameter masukan JST-BP supaya JST-BP dapat bekerja secara optimal.

Beberapa peneliti telah menggunakan algoritma *Particle Swarm Optimization* (PSO)untuk mengoptimalkan suatu metode dalam pencarian parameter terbaik untuk suatu peramalan. Zhu *et al*. (2020) menggunakan algoritma PSO untuk menentukan parameter bobot dan bias terbaik pada model *Extreme Learning Machine* (ELM) dalam memprediksi evapotranspirasi harian yang memperoleh nilai RMSE lebih kecil dibanding tanpa menggunakan algortima PSO yaitu 17,1 % : 17,9% pada model berbasis radiasi dan 20,6% : 21,7% pada model berbasis suhu. Begitu pula dengan penelitian peramalan harga saham menggunakan data *time series* yang dilakukan oleh Rusmalawati & Furqon (2018), algoritma PSO telah membantu mengoptimalkan *Support Vector Regression* (SVR) untuk membentuk model peramalan melalui *sequential learning* SVR, hasil evaluasi yang didapatkan tergolong memiliki tingkat akurasi yang tinggi karena memiliki nilai MAPE sebesar 0,8195% dengan *fitness* 0,5496.

Berdasarkan uraian permasalahan di atas, maka penelitian ini berfokus pada pengoptimalan JST-BP menggunakan algoritma PSO untuk memprediksi harga minyak mentah dunia. Hal tersebut juga menjadi latar belakang dilakukannya penelitian pada skripsi dengan mengambil judul **“Prediksi Harga Minyak Mentah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* (JST-BP) dan *Particle Swarm Optimization* (PSO)”**.

## Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana cara kerja algoritma PSO untuk pencarian bobot terbaik pada JST-BP dalam memprediksi harga minyak mentah?
2. Bagaimana hasil akurasi dari JST-BP setelah dilakukannya pencarian bobot terbaik menggunakan PSO untuk prediksi harga minyak mentah?

## Batasan Masalah

Batasan-batasan masalah dalam penelitian adalah sebagai berikut:

1. Algoritma prediksi yang digunakan adalah JST-BP
2. Menggunakan algoritma optimasi PSO untuk meningkatkan akurasi algoritma JST-BP
3. Algoritma PSO yang digunakan dalam penelitian ini untuk mencari optimasi nilai *w* atau bobot
4. Data yang digunakan dalam penelitian ini adalah *Crude Oil* standar WTI (CL=F) *dataset* dari situs finance.yahoo.com
5. Bahasa pemrograman yang digunakan dalam penelitian ini adalah *Python*

## Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui cara kerja algoritma PSO untuk meningkatkan akurasi JST-BP pada prediksi harga minyak mentah.
2. Menganalisis hasil optimasi JST-BP dengan PSO pada data harga minyak mentah melalui hasil akurasi yang diperoleh.

## Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mengetahui penerapan algoritma PSO untuk meningkatkan akurasi pada JST-BP dalam memprediksi harga minyak mentah.
2. Mengetahui hasil peningkatan akurasi algoritma PSO pada data harga minyak mentah menggunakan model prediksi JST-BP.

## Sistematika Penulisan

Sistematika penulisan berguna untuk memudahkan dalam memahami jalan pemikiran secara keseluruhan skripsi. Penulisan skripsi ini secara garis besar dibagi menjadi tiga bagian, yaitu sebagai berikut.

### Bagian Awal Skripsi

Bagian awal skripsi terdiri dari halaman judul, halaman pengesahan, halaman pernyataan, halaman motto dan persembahan, abstrak, kata pengantar, daftar isi, daftar gambar, daftar tabel dan daftar lampiran.

### Bagian Isi Skripsi

Bagian isi skripsi terdiri dari lima bab, yaitu sebagai berikut.

1. BAB 1: PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan skripsi.

1. BAB 2: TINJAUAN PUSTAKA

Bab ini berisi penjelasan mengenai definisi maupun pemikiran-pemikiran yang dijadikan kerangka teoritis yang menyangkut dan mendasari pemecahan masalah dalam skripsi ini.

1. BAB 3: METODE PENELITIAN

Bab ini berisi penjelasan mengenai studi pendahuluan, tahap pengambilan data, normalisasi data, pengujian parameter, dan peramalan.

1. BAB 4: HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian berserta pembahasannya.

1. BAB 5: PENUTUP

Bab ini berisi simpulan dari penulisan skripsi dan saran yang diberikan penulis untuk mengembangkan skripsi ini.

### Bagian Akhir Skripsi

Bagian akhir skripsi ini berisi daftar pustaka yang merupakan informasi mengenai buku-buku, sumber-sumber dan referensi yang digunakan penulis serta lampiran-lampiran yang mendukung dalam penulisan skripsi ini.

# BAB 2 TINJAUAN PUSTAKA



## Prediksi

Peramalan atau prediksi adalah proses untuk memperkirakan berapa kebutuhan dimasa yang akan datang. Peramalan merupakan salah satu metode statistik yang memiliki peran penting dalam mengambil keputusan di masa mendatang (Hartati, 2017). Peramalan yang digunakan untuk masa mendatang dilakukan berdasarkan informasi berupa data masa lalu. Metode yang dapat digunakan dalam peramalan salah satunya yaitu metode *time series*. *Time series* merupakan urutan nilai dalam interval waktu tertentu (Deb *et al.*, 2017).

Menurut Wulandari *et al.* (2017), pendekatan peramalan dapat diklasifikasikan menjadi dua pendekatan, yaitu:

1. Pendekatan Kualitatif
2. Pendekatan Kuantitatif

Pada penggunaan metode kualitatif, akan lebih banyak menuntut analisis yang didasarkan pada pemikiran *intuitif*, pemikiran logis dan informasi pengetahuan yang telah diperoleh peneliti sebelumnya. Pendekatan peramalan metode ini biasanya digunakan untuk peramalan jangka pendek atau apabila pengambilan keputusan lebih mepercayai intuisinya dari pada rumus matematika. Sedangkan metode pendekatan kuantitatif diperlukan informasi atau data masa lalu yang dikuantitatifkan dalam bentuk numerik yang ramalannya berdasarkan pada statistika dan matematika. Metode peramalan kuantitatif memiliki dua jenis model yaitu model runtun waktu (*time series*) dan model regresi (*regression*) (Wulandari *et al*., 2017).

Penelitian yang akan dilakukan penulis yaitu memanfaatkan metode pendekatan kuantitatif dengan model *time series* menggunakan data runtun waktu harga harian minyak mentah dunia dalam WTI dan Brent. Data runtun waktu (*time series data*) adalah kumpulan data yang telah dicatat maupun observasi berdasarkan urutan waktu tertentu. Tujuan menganalisis data *time series* yaitu untuk menemukan bentuk atau pola variasi dari data dimasa lalu yang digunakan untuk melakukan suatu prediksi terhadap data di masa mendatang (Deb *et al*., 2017).

## Jaringan Syaraf Tiruan (JST)

### Pengertian Jaringan Syaraf Tiruan

JST adalah alat untuk memodelkan proses nonlinier berdasarkan informasi yang dikumpulkan oleh vektor bernama lapisan *input*, dimana informasi disebarkan lapisan demi lapisan yang membangun hubungan antara *input* dan lapisan terakhir yang disebut lapisan *output* (Aulia, 2018). Lapisan menengah atau tersembunyi terdiri dari satu atau lebih unit yang disebut *neuron* yang saling berhubungan dengan *neuron* dari lapisan sebelumnya dan selanjutnya. Jumlah lapisan tersembunyi dan jumlah *neuron* dari masing-masing menentukan topologi dari jaringan (Navares *et al*., 2018). JST juga merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai Jaringan Syaraf Biologis (JSB), JST tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) (Sudarsono, 2016).

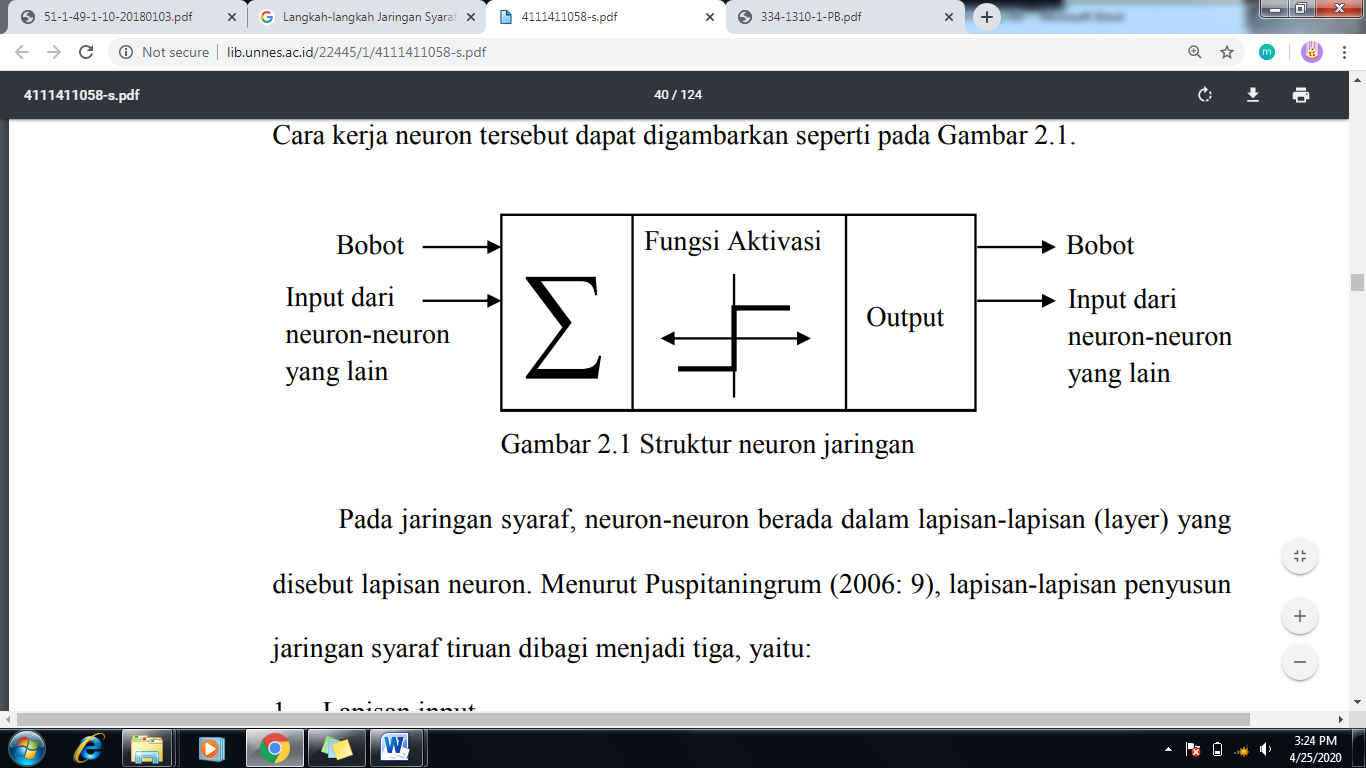
JST berjalan seperti halnya manusia yang belajar dari suatu contoh tertentu. JST bekerja dengan melalui proses pembelajaran dari contoh pelatihan yang diberikan untuk menyelesaikan suatu permasalahan (Almas & Setiawan, 2018). Dalam hal ini JST memiliki beberapa syarat yang harus dipenuhi untuk menggunakan fungsi aktivasi sebagai penentuan *output* suatu *neuron*, diantaranya yaitu kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun (Rahayu *et al*., 2018).

Terdapat beberapa istilah dalam JST yang sering ditemui (Giawa, 2017), antara lain sebagai berikut:

1. *Neuron/node* (unit) merupakan sel syaraf tiruan yang berperan sebagai elemen pengolahan jaringan syaraf dalam menerima data *input* atau masukan, kemudian memproses data *input* tersebut dan mengirimkan hasilnya berupa sebuah *output* atau keluaran.
2. Jaringan merupakan kumpulan dari *neuron*-*neuron* yang saling terkoneksi atau terhubung dan membentuk sebuah lapisan.
3. *Hidden Layer* (lapisan tersembunyi) merupakan lapisan yang secara langsung tidak berhubungan dengan dunia luar. Lapisan ini memiliki kegunaan memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah yang kompleks.
4. *Input* merupakan sebuah atribut tunggal dari suatu pola atau data lain dari dunia luar. Sinyal-sinyal *input* ini kemudian diteruskan ke lapisan berikutnya.
5. *Output* merupakan hasil pemahaman jaringan terhadap data *input*. Hal ini dikarenakan tujuan dari pembangunan jaringan syaraf tiruan adalah mendapatkan nilai *output*.
6. *Bobot* merupakan nilai matematis antar *neuron* yang berfungsi untuk mengatur *output* yang diinginkan sekaligus bertujuan untuk membuat jaringan tersebut dalam pembelajaran.
7. Fungsi aktivasi merupakan suatu fungsi yang digunakan untuk merubah nilai-nilai bobot per-iterasi dari semua nilai *input*.
8. Fungsi aktivasi sederhana yaitu mengalikan *input* dengan bobotnya kemudian menjumlahkannya. Perhitungan ini biasanya berbentuk linier atau tidak linier dan sigmoid.
9. Paradigma pembelajaran merupakan bentuk pembelajaran yang dibedakan menjadi dua yaitu *supervised learning* dan *unsupervised learning*.
10. Aturan pembelajaran merupakan aturan kerja dari teknik atau algoritma jaringan syaraf tiruan.

### Komponen Jaringan Syaraf Tiruan

JST memiliki hubungan seperti halnya dengan sistem syaraf otak pada manusia. Dalam JST hubungan yang dimaksud yaitu hubungan antar *neuron*, karena pada setiap JST terdapat beberapa *neuron* (Sudarsono, 2016). Dapat dikatakan bahwa *neuron* merupakan sel syaraf yang merupakan dasar operasi pada JST atau *unit* pemroses informasi. Hubungan antara *neuron-neuron* tersebut dikenal dengan nama bobot (Sudarsono, 2016). Cara kerja neuron dapat digambarkan seperti Gambar 2.1.



Gambar 2. Struktur *neuron* jaringan (Sudarsono, 2016)

Pada jaringan syaraf tiruan, *neuron-neuron* berada dalam lapisan-lapisan (*layer*) yang disebut lapisan *neuron*. Lapisan-lapisan penyusun JST dibagi menjadi tiga (Aulia, 2018), yaitu:

1. *Input Layer*

*Neuron-neuron* yang berada di dalam lapisan *input* disebut *neuron-neuron* *input*. *Neuron-neuron* ini menerima *input* dari luar yang tidak berubah pada fase pelatihan dan hanya bisa berubah jika diberikan nilai masukan baru. *Input* yang dimasukkan merupakan penggambaran dari suatu masalah.

1. *Hidden Layer*

*Neuron-neuron* di dalam lapisan tersembunyi disebut *neuron-neuron* tersembunyi. *Output* dari lapisan ini tidak secara langsung dapat diamati, akan tetapi semua proses pada fase pelatihan dan fase pengenalan dijalankan pada lapisan ini. Jumlah lapisan ini pada umumnya terdiri dari satu lapisan *hidden layer* atau tergantung dari arsitektur yang dirancang.

1. *Output Layer*

*Neuron-neuron* pada lapisan *output* disebut *neuron-neuron* *output*. Keluaran atau *output* dari lapisan ini merupakan *output* JST terhadap suatu permasalahan.

### Arsitektur Jaringan Syaraf Tiruan

JST memiliki tiga macam arsitektur yang sering digunakan dalam berbagai aplikasi (Revi *et al*., 2018), diantaranya adalah sebagai berikut:

1. JST Lapisan Tunggal (*Single Layer Net*)

Jaringan Syaraf dengan lapisan tunggal ini hanya memiliki satu lapis *input* dan *output* dengan bobot-bobot terhubung. Kinerja jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Cara kerja jaringan saraf lapisan tunggal ditunjukkan pada Gambar 2.2.

w1

w2

Nilai Input

Lapisan Input

Matriks bobot

Lapisan output

Nilai output

Gambar 2. Jaringan Syaraf Tiruan Lapisan Tunggal (Revi *et al*., 2018)

1. JST Lapisan Banyak (*Multi Layer Net*)

Jaringan syaraf dengan lapisan banyak memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output* dan memiliki satu atau lebih lapisan tersembunyi. Jaringan ini dapat menyelesaikan permasalahan yang lebih sulit dari pada jaringan lapisan tunggal. Namun, dalam proses pelatihannya akan memerlukan waktu yang cenderung lama. Jaringan syaraf lapisan banyak memiliki cara kerja seperti yang ditunjukkan oleh Gambar 2.3.

Nilai Input

Lapisan Input

Matriks bobot

Lapisan output

Nilai output

Lapisan Tersembunyi

Matriks bobot

w11

w12

w21

w22

w1

w2

Gambar 2. Jaringan Syaraf Tiruan Banyak Lapisan

(Revi *et al*., 2018)

1. JST Kompetitif (*Competitive Layer Net*)

Jaringan syaraf lapisan kompetitif memiliki bentuk yang berbeda dimana antar neuron dapat saling dihubungkan. Dalam jaringan ini, *neuron* akan bersaing untuk mendapatkan hak untuk aktif. Hal ini dapat digambarkan seperti pada Gambar 2.4.



Gambar 2. Jaringan Syaraf Tiruan Lapisan Kompetitif

(Revi *et al*., 2018)

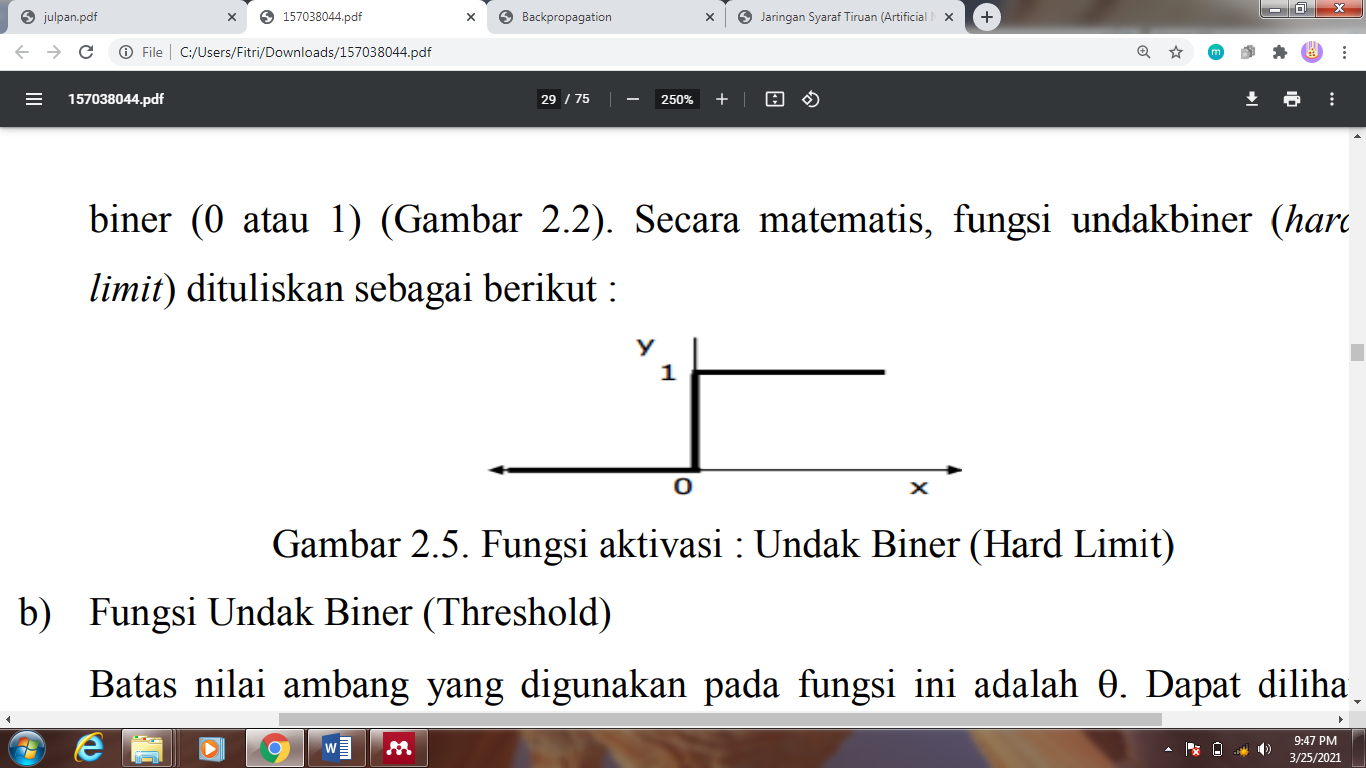
### Fungsi Aktivasi Jaringan Syaraf Tiruan

Neuron-neuron yang terdapat dalam JST memerlukan fungsi aktivasi supaya dapat menjalankan tugasnya. Fungsi aktivasi yaitu fungsi yang digunakan pada jaringan syaraf untuk mengaktifkan atau tidak mengaktifkan neuron. Berikut beberapa fungsi aktivasi pada JST (Siregar, 2019).

* 1. Fungsi Undak Biner (*Hard Limit*)

Fungsi ini biasa digunakan oleh jaringan lapisan tunggal untuk mengkonversi nilai *input* dari suatu *variable* yang bernilai kontinu ke suatu nilai *output* yang diberikan pada Persamaan 1.

(1)

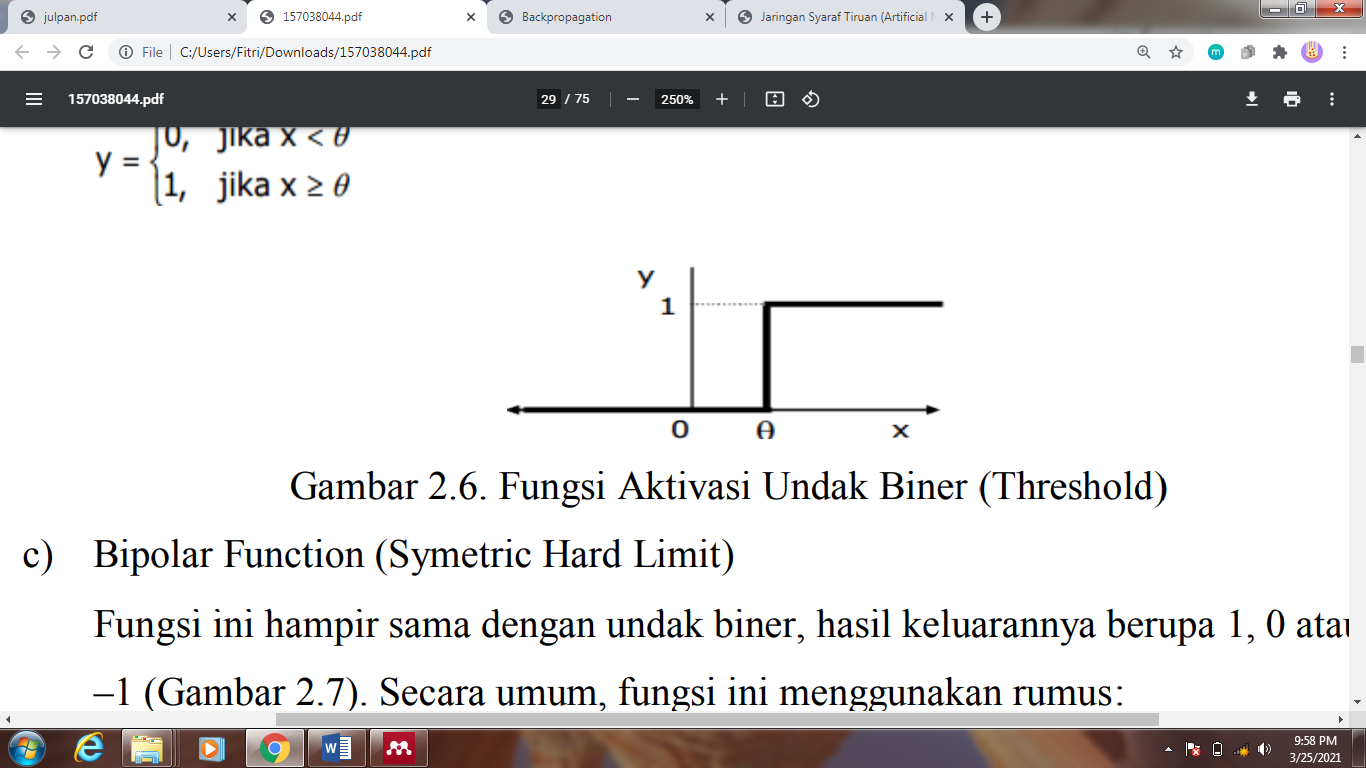


Gambar 2. Fungsi Aktivasi Undak Biner Hard Limit (Siregar, 2019)

* 1. Fungsi Undak Biner (*Threshold*)

Pada rumus Persamaan 2, fungsi ini memiliki batas nilai ambang yang digunakan yaitu θ.

(2)

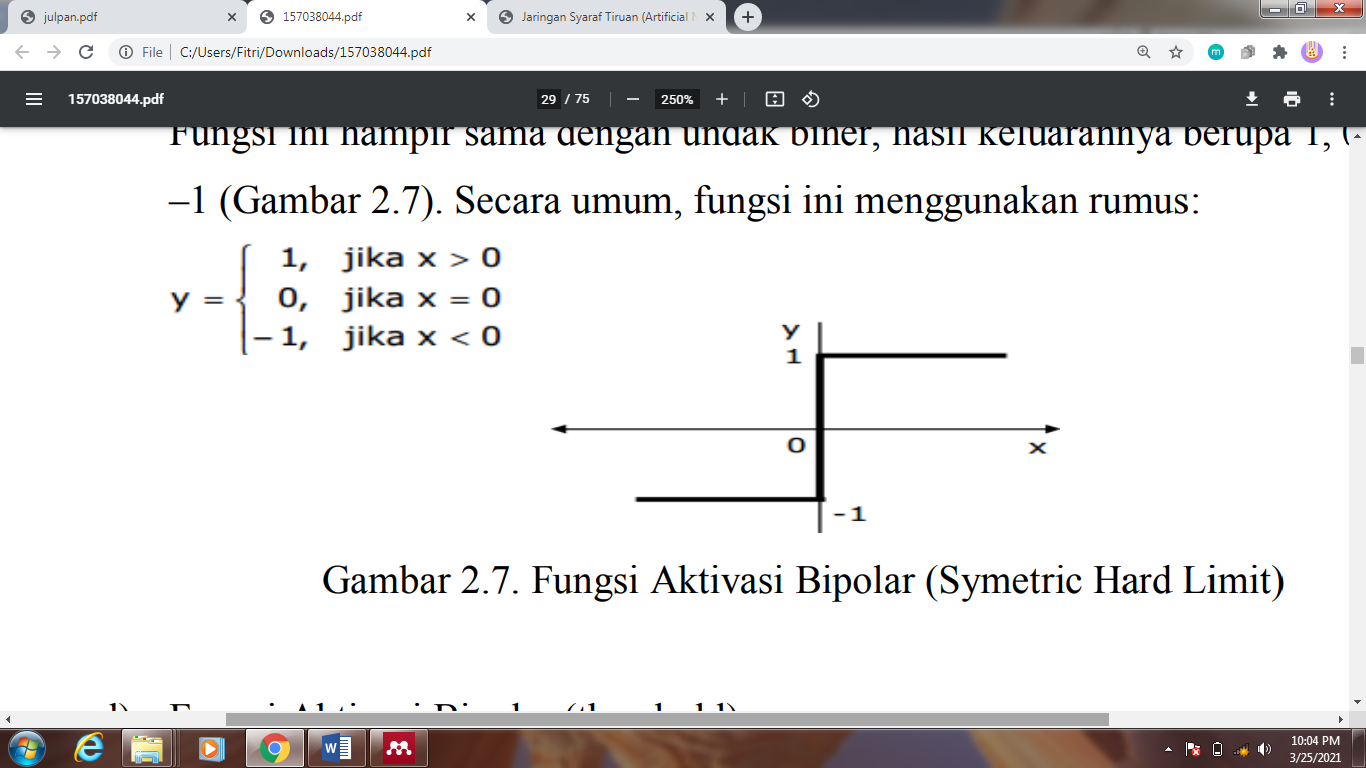


Gambar 2. Fungsi Aktivasi Undak Biner *Threshold* (Siregar, 2019)

* 1. Fungsi Bipolar (*Symetric Hard Limit*)

Hampir sama dengan undak biner, fungsi ini hasil keluarannya berupa 1,0 atau -1 seperti yang ditunjukkan oleh Persamaan 3.

(3)

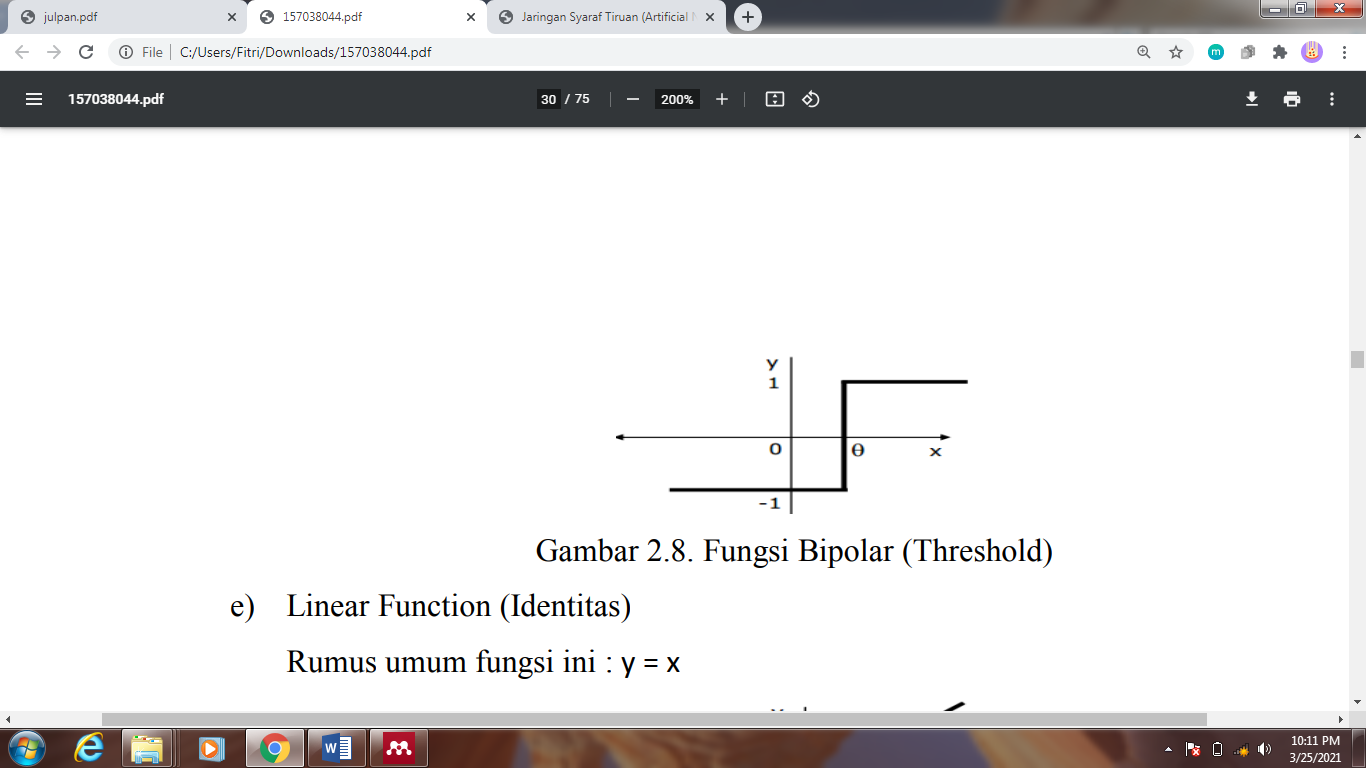


Gambar 2. Fungsi Bipolar *Symetric Hard Limit* (Siregar, 2019)

* 1. Fungsi Bipolar dengan *Threshold*

Mirip dengan undak biner, hasil keluarannya berupa 1, 0 atau -1 dengan nilai ambang θ. Fungsi ini dapat dilihat pada Persamaan 4 berikut.

(4)

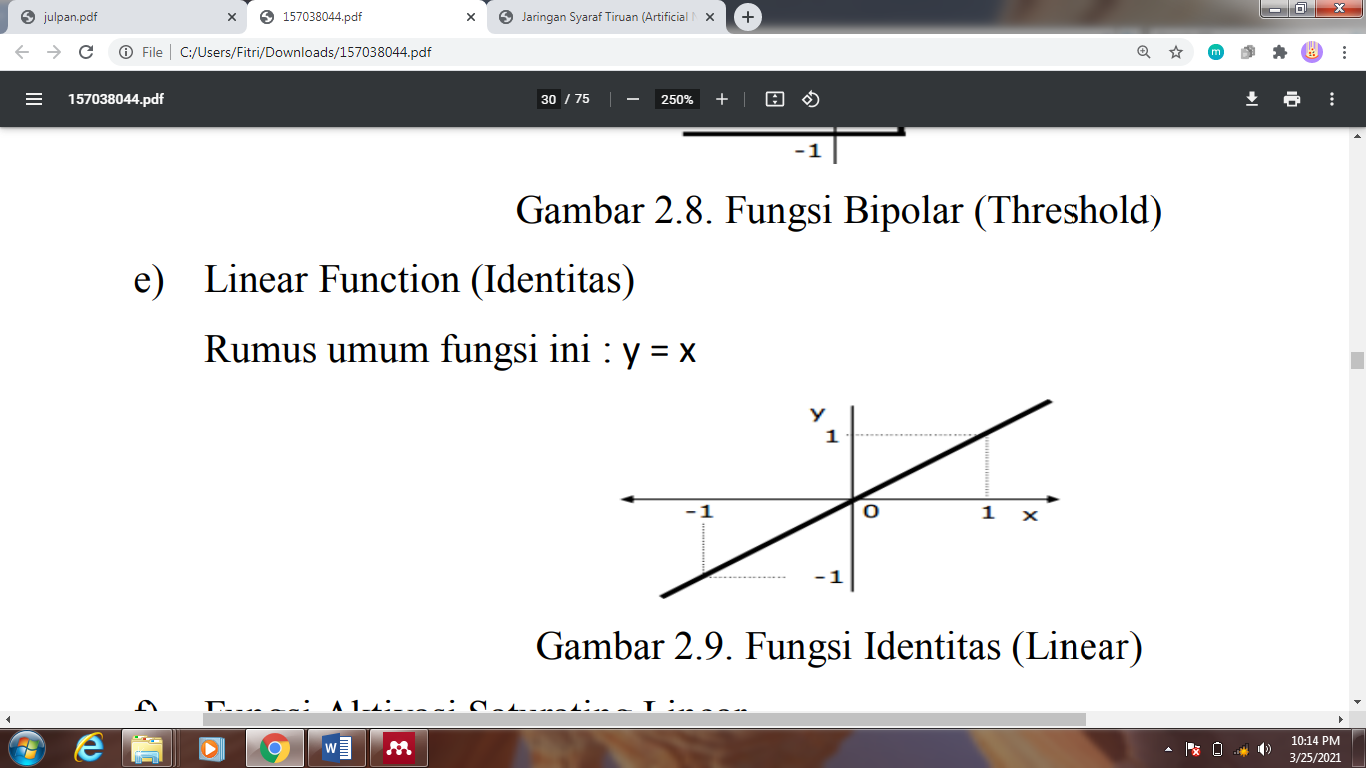


Gambar 2. Fungsi Bipolar *Threshold* (Siregar, 2019)

* 1. Fungsi Identitas (*Linear Function*)

Rumus fungsi ini dapat dilihat dalam Persamaan 5 berikut.

*y = x* (5)



Gambar 2. Fungsi Identitas (Siregar, 2019)

* 1. Fungsi *Saturating Linear*

Pada fungsi aktivasi ini, akan memiliki nilai 0 jika masukkannya lebih kecil dari dan akan bernilai 1 jika inputnya lebih dari . Sedangkan apabila nilai *input*nya terletak diantara dan , maka nilai *output*nya akan memiliki nilai yang sama dengan nilai masukkannya ditambah dengan . Rumus fungsi ini ditunjukkan oleh Persamaan 6 berikut.

(6)

* 1. Fungsi *Symetric Saturating Linear*

Fungsi ini bernilai -1 jika nilai masukannya tidak lebih dari -1. Apabila nilai masukannya >1, maka nilai fungsi ini adalah 1. Selain itu, apabila masukannya berkisar antara -1 dan 1, maka nilai keluarannya akan sama dengan nilai masukannya seperti yang ditunjukkan oleh Persamaan 7.

(7)

* 1. Fungsi Sigmoid Biner

Batas nilai untuk fungsi aktivasi ini antara 0 sampai 1. Fungsi aktivasi ini sering digunakan untuk jaringan syaraf yang nilai *output*mya antara 0 sampai 1. Tetapi fungsi ini juga dapat digunakan untuk jaringan syaraf yang nilai keluarannya 0 atau 1. Rumus fungsi sigmoid biner ditunjukkan oleh Persamaan 8 dan 9.

(8)

(9)

* 1. Fungsi Sigmoid Bipolar

Fungsi ini hamper sama dengan sigmoid biner, perbedaannya terdapat pada nilai *output*nya yang memiliki batas antara 1 sampai -1 yang diberikan pada Persamaan 10 dan 11 berikut.

(10)

(11)

## JST-Backpropagation

### Backpropagation

*Backpropagation* atau perambat galat mundur adalah salah satu metode JST yang telah banyak digunakan dalam peramalan karena memiliki sistem kerja dapat meniru perilaku kompleks dan *non-linear* melalui *neuron* (Wanto & Windarto, 2017). JST dengan menggunakan algoritma *Backpropagation* akan bekerja secara terawasi, sehingga metode ini dapat digunakan sebagai prediksi dan klasifikasi data (Salsabila & Cholissodin, 2020). Pada JST-BP gradient *error* pada *neuron-neuron* tersembunyi diturunkan dari penyiaran kembali *error-error* diasosiasikan dengan *neuron-neuron output*. Nilai *error* dihitung berdasarkan rata-rata kuadrat kesalahan atau sering disebut dengan MSE (*Mean Square Error*) (Almas & Setiawan, 2018). Hal ini terjadi karena nilai target pada *neuron –neuron* tersembunyi tidak diberikan.

*Backpropagation* pada umumnya, digunakan dalam jaringan dengan lapisan banyak (*multi layer network*) atau berlebih untuk memberikan perubahan bobot yang terhubung dengan neuron pada lapisan tersembunyi. Model algoritma *backpropagation* memiliki satu lapisan masukkan (*input layer*), lapisan tersembunyi (*hidden layer*) dan beberapa lapisan keluaran (*output layer*) dengan di tiap lapisan terdiri dari beberapa *neuron*. Algoritma *backpropagation* terdiri dari langkah pelatihan dan pengujian dalam menyelesaikan suatu permasalahan. Pelatihan merupakan suatu proses dimana jaringan belajar untuk mengenali pola-pola dari dataset *input*. Pelatihan pada jaringan *backpropagation* melibatkan tiga tahap yaitu umpan maju pelatihan pola masukan (*feedforward*), komputasi dan perambatan balik galat (*backpropagation*) (Julpan *et al*., 2015). Sedangkan pengujian merupakan proses mengevaluasi kemampuan jaringan untuk uji kelayakan dalam menyelesaikan permasalahan dengan menentukan seberapa dekat *output* aktual, apakah sesuai dengan *output* yang diinginkan dalam situasi baru. *Backpropagation* memiliki struktur jaringan seperti yang dapat dilihat pada Gambar 2.10.

Input Layer

Hidden Layer

Output Layer

Gambar 2. Struktur Jaringan *Backpropagation* (Revi *et al*., 2018)

### Algoritma Backpropagation

Pelatihan pada algoritma *backpropagation* memiliki tiga fase (Salsabila & Cholissodin, 2020) , diantaranya yaitu sebagai berikut:

1. Fase 1: Propagasi maju (*feedforward*)

Pada fase pertama, pola masukan dihitung maju dari *input layer* hingga *output layer* menggunakan fungsi aktivasi yang ditentukan.

1. Fase 2: Propagasi mundur (*backpropagation*)

Fase yang kedua selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi tersebut dipropagasi mundur. Propagasi mundur dimulai dari garis yang berhubungan langsung dengan unit-unit pada *output layer*.

1. Fase 3: Perubahan bobot

Terakhir, modifikasi bobot untuk menurunkan tingkat kesalahan yang terjadi.

Langkah-langkah dari 3 fase algoritma *backpropagation* dijabarkan sebagai berikut:

1. Inisialisasi bobot secara acak pada jaringan
2. Tentukan nilai *epoch* dan *error* yang diharapkan
3. Lakukan langkah 4 sampai 8 apabila kondisi berhenti belum tercapai
4. Lakukan langkah 5 sampai 7 pada tiap data *training*
5. Fase 1: *Feed Forward Propagation*

* Jumlahkan seluruh sinyal yang masuk pada *neuron hidden* menggunakan Persamaan 12.

(12)

dimana,

: total sinyal *input* pada *neuron hidden j*

: *bias* awal yang telah diinisialisasikan

: nilai *input* pada *neuron i*

: bobot antara *neuron input i* dengan *neuron hidden j*

* Hitung keluaran seluruh *neuron* *hidden* j pada *hidden layer* dengan Persamaan 13.

(13)

: keluaran dari *neuron hidden j*

: total sinyal *input* pada *neuron hidden j*

* Jumlahkan seluruh sinyal yang masuk pada *neuron output k* sesuai Persamaan 14.

(14)

: total sinyal *input* pada *neuron output k*

: nilai *input* pada *neuron j*

: bobot antara *neuron hidden j* dengan *neuron output k*

* Hitung keluaran semua *neuron* pada *output layer* menggunakan Persamaan 15.

(15)

: keluaran dari *neuron output k*

: total sinyal *input* pada *neuron output k*

1. Fase 2: *Backward Propagation*

* Hitung faktor kesalahan pada *output layer* menggunakan Persamaan 16.

(16)

: faktor kesalahan dari *neuron output k*

: target pada *neuron output k*

: keluaran dari *neuron output k*

* Hitung suku perubahan bobot dari *hidden layer* menggunakan Persamaan 17.

(17)

: *learning rate* dengan nilai 0 sampai 1

* Hitung penjumlahan *error* dari *neuron hidden* dengan menggunakan Persamaan 18.

(18)

* Hitung faktor kesalahan pada *hidden layer* menggunakan Persamaan 19.

(19)

* Hitung suku perubahan bobot dari *input layer* menggunakan Persamaan 20.

(20)

1. Fase 3: Perubahan Bobot

* Mengubah bobot diantara *hidden layer* dan *output layer* dengan menggunakan Persamaan 21.

(21)

* Mengubah bobot diantara *input layer* dan *hidden layer* menggunakan Persamaan 22.

(22)

: *momentum*

1. Hitung nilai *Mean Square Error* (MSE) di setiap *epoch* atau iterasi menggunakan Persamaan 23.

(23)

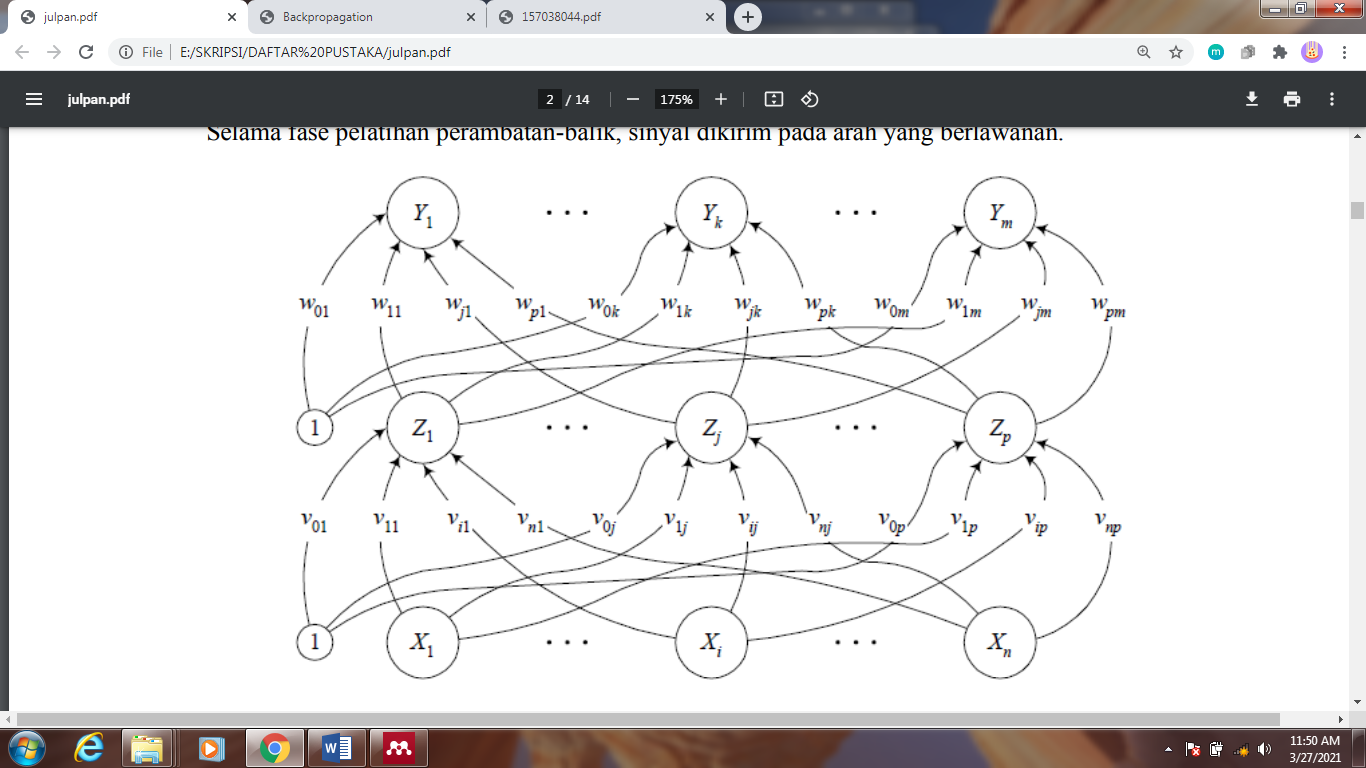
: target nilai pada *neuron output k*

: keluaran dari *neuron output k*

: jumlah pola

### Arsitektur Backpropagation

Terdapat beberapa algoritma yang dapat digunakan untuk membuat jaringan syaraf tiruan, namun *backpropagation* sering dipilih karena dianggap mudah diimplementasikan sekaligus menjaga efisiensi jaringan (Siregar, 2019). Oleh sebab itu, algoritma *backpropagation* menjadi algoritma yang paling dikenal diantara Jaringan Syaraf Tiruan. Algoritma *backpropagation* dipilih karena menggunakan lebih dari satu lapisan atau sering disebut dengan *multi layer* yaitu terdiri dari *input layer*, *output layer*, dan *hidden layer*. Arsitektur jaringan syaraf *backpropagation* dengan *multi layer* memiliki lapisan tersebunyi yaitu unit Z, lapisan keluaran pada unit Y, serta masukan pada unit X yang dapat dilihat pada Gambar 2.11. Bias untuk unit keluaran Yk dilambangkan dengan W0k, begitu pula dengan bias yang dilambangkan V0j merupakan bias pada unit tersembunyi Zj. Nilai bias berlaku sebagai bobot yang menghubungkan unit-unit tersebut dengan keluaran selalu satu (Julpan *et al*., 2015).



Gambar 2. Arsitektur JST-BP(Julpan *et al*., 2015)

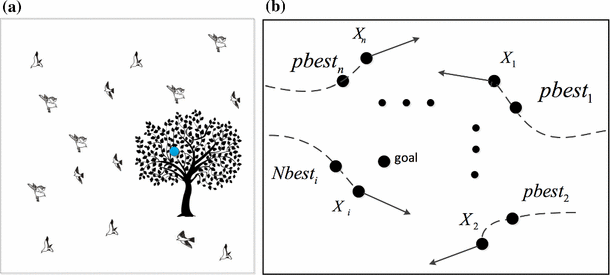
## *Particle Swarm Optimization* (PSO)

PSO merupakan suatu teknologi pengoptimalan heuristik untuk menemukan solusi optimal berdasarkan fungsi objektif yang didasarkan pada perilaku segerombolan burung dalam menjalankan tugasnya (Xu *et al*., 2018). PSO yang termasuk dalam salah satu teknik dasar dari *swarm intelligence system* yang dimanfaatkan untuk menyelesaikan masalah optimasi dalam pencarian ruang sebagai solusi.Perilaku segerombolan burung atau serangga terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok (Mukhlis, 2016).

PSO pertama kali diusulkan dan dirancang oleh James Kennedy dan Eberhart untuk mensimulasikan burung dalam proses pencarian makanan pada tahun 1995. Beberapa istilah yang biasanya digunakan dalam metode PSOdidefinisikan sebagai berikut (Lubis, 2017):

* + - * 1. *Swarm*, merupakan populasi dari suatu algoritma PSO.
        2. *Particle,* adalah anggota atau individu pada suatu *swarm*. Setiap *particle* mempresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu *particle* ditentukan oleh representasi solusi saat itu.
        3. *Pbest* (*Personal best*), adalah posisi *Pbest* suatu *particle* yang menunjukkan posisi *particle* yang dipersiapkan untuk mendapatkan suatu posisi terbaik.
        4. *Gbest* (*Global best*), merupakan posisi terbaik *particle* pada *swarm* atau posisi terbaik diantara *Pbest*.
        5. *Velocity* (*v*), merupakan vektor yang menggerakkan proses optimasi untuk menentukan arah dimana suatu *particle* deperlukan untuk berpindah (*move*) supaya memperbaiki posisinya semula atau kecepatan yang menggerakkan proses optimasi yang menentukan arah dimana *particle* tersebut diperlukan untuk berpindah dan memperbaiki posisi semulanya.
        6. *Inertia weight* (θ), disimbolkan dengan *w* yang digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu *particle*.
        7. *Learning rates* (c1 dan c2), merupakan suatu konstanta untuk menilai kemampuan *particle* (c1) dan kemampuan social *swarm* (c2) yang menunjukkan bobot dari *particle* terhadap memorinya.

Seekor burung yang berada dalam suatu kelompok (*swarm*) menunjukkan kata partikel, setiap partikel diasumsikan memiliki dua karakteristik yaitu posisi dan kecepatan. Tiap partikel atau individu berperilaku secara terdistribusi dengan menggunakan kecerdasannya (*intelligence*) sendiri, juga dipengaruhi oleh perilaku kelompok kolektifnya. Seperti halnya sekawanan burung yang apabila salah satu atau individu menemukan jalan yang tepat dan pendek menuju ke sumber makanan, maka sisa kawanan yang lain juga akan segera dapat mengikuti jalan tersebut meskipun lokasi mereka jauh (Mukhlis, 2016). Seperti pada Gambar 2.11 yang menunjukkan bahwa sekumpulan burung tersebar di suatu area dan hanya ada satu makanan saja (Xu *et al*., 2018).



Gambar 2. Simulasi burung yang sedang mencari makanan (Xu *et al*., 2018)

Setiap partikel dalam PSO akan bergerak di ruang (*space*) tertentu dan akan tetap mengingat posisi terbaik yang pernah ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel yang bergerak akan menyampaikan informasi atau posisi terbaik pada partikel lain sehingga posisi dan kecepatan masing-masing partikel akan menyesuaikan berdasarkan informasi yang mereka terima mengenai posisi terbaik. Posisi dari tiap partikel dianggap sebagai calon solusi (*candidate* solution) bagi suatu masalah optimasi. Dalam simulasi Gambar 2.6 a titik yang ada di pohon melambangkan makanan yang posisinya tidak diketahui oleh burung meskipun mereka tahu jarak diantaranya. Burung yang berada di dekat makanan akan memberitahu kawannya untuk menuju makanan. Sedangkan pada Gambar 2.5 b makanan diasumsikan memiliki nilai optimal, burung sebagai partikel, dan jarak burung antar makanan sebagai nilai fungsi objektif. Proses ini dinamakan sebagai proses fungsi optimalisasi. Dimana Xi adalah partikel terdekat dengan sasaran yang dijadikan partikel optimal global sementara, *Nbest*i adalah nilai optimal global. Semua partikel yang bergerak akan memperbarui posisinya sesuai dengan posisi optimal lokal masing-masing dan posisi optimal global yang telah dikunjungi. Penentuan posisi terbaru sebuah partikel didifinisikan seperti pada Persamaan 24 berikut.

(24)

dimana,

: status saat ini

: status setelah pembaruan

: posisi partikel saat ini

: kecepatan baru partikel

Kecepatan partikel *i* didefinisikan seperti pada Persamaan 25 berikut.

(25)

dimana,

: kecepatan partikel saat ini

: posisi terbaik sejauh ini yang dikunjungi partikel (posisi lokal)

: posisi terbaik global sebuah partakel pada tingkat gerombolan

: konstanta yang menekankan pentingnya setiap kecepatan

: nilai acak dalam [0,1]

## Harga Minyak Mentah (*Crude Oil Price*)

*Crude oil* atau disebut juga dengan minyak bumi adalah bahan bakar fosil yang terdapat di bumi dan terbentuk dari tumbuhan dan hewan fosil selama jutaan tahun lamanya. Minyak mentah kemudian disuling menjadi berbagai produk minyak bumi, diantaranya yaitu bensin atau gasoline (Chen, 2020). Minyak mentah dikategorikan menjadi ringan atau berat, serta manis atau asam. Pemanfaatan minyak mentah sangatlah beragam, selain sebagai sumber bahan bakar kendaraan minyak mentah juga banyak digunakan di berbagai industry seperti pembuatan pelarut, baja, sabun, detergen, bahan peledak dan plastik. Pentingnya minyak mentah sebagai penunjang dalam kehidupan sehari hari menjadikan harga minyak sangat berpengaruh seiring dengan peningkatan jumlah penduduk di suatu Negara. Salah satu contoh dapat dibuktikan dari penggunaan Bahan Bakar Minyak (BBM) yang semakin meningkat pesat (Vinet & Zhedanov, 2011).

Harga minyak dunia merupakan suatu harga yang telah dibuat karena permintaan dan penawaran seluruh komoditas minyak dunia. Harga minyak dijadikan suatu penetapan nilai tukar yang ditetapkan oleh perekonomian suatu negara di setiap penjual dan pembeli untuk memperoleh produk. Harga minyak mentah diukur berdasarkan harga spot pasar minyak dunia. Standar ukuran harga minyak mentah yaitu *West Texas Intermediate* (WTI) atau *Brent* (Putra, 2016).

Minyak mentah yang diperdagangkan dalam WTI merupakan minyak mentah yang memiliki kualitas tinggi dan berkadar belerang yang rendah atau sering disebut dengan kualitas *light weight*. Harga minyak jenis ini dijadikan patokan untuk harga perdagangan minyak dunia. Hal tersebut dikarenakan minyak yang tergolong dalam WTI sangat cocok untuk dijadikan bahan bakar, sehingga harganya lebih tinggi lima sampai enam *dollar* dibandingkan harga minyak OPEC dan lebih tinggi satu hingga dua *dollar* dari harga minyak *Brent*. OPEC memegang posisi penting sebagai kartel yang berperan mempengaruhi sisi suplai dalam perdagangan minyak mentah dunia (Faozi & Sulistijanti, 2016). Terdapat beberapa faktor yang mempengaruhi harga minyak dunia (Setiyowati *et al*., 2018), diantaranya yaitu:

1. Faktor Fundamental, terdiri dari permintaan minyak, pasokan minyak, stok minyak, kapasitas produksi cadangan dunia dan kemampuan kilang dunia.
2. Faktor Non fundamental, terdiri dari geopolitik, kebijakan pemerintah, bencana alam, cuaca, pemogokan, kerusakan instalasi ranai produksi, pelemahan nilai dollar dan spekulasi.
3. Pengaruh dan kebijakan pasokan OPEC (*Organization of the Petroleum Countries*).

Data yang digunakan dalam penelitian ini menggunakan *dataset* *Crude Oil* (CL=F) yang merupakan standar harga WTI dari situs finance.yahoo.com. *Dataset* harga minyak mentah WTI ini diukur dalam satuan dollar Amerika (USD). Beberapa *dataset* harga minyak mentah dunia standar WTI dapat ditunjukkan pada Tabel 2.1.

Tabel 2. D*ataset* Harga Minyak Mentah Dunia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj Close | Volume |
| 1/2/2018 | 60.2 | 60.74 | 60.1 | 60.37 | 60.37 | 510313 |
| 1/3/2018 | 60.39 | 61.97 | 60.28 | 61.63 | 61.63 | 673859 |
| 1/4/2018 | 61.96 | 62.21 | 61.59 | 62.01 | 62.01 | 654363 |
| 1/5/2018 | 61.9 | 62.04 | 61.09 | 61.44 | 61.44 | 563036 |
| 1/7/2018 | null | null | null | null | null | null |
| 1/8/2018 | 61.61 | 61.97 | 61.34 | 61.73 | 61.73 | 616474 |
| 1/9/2018 | 61.92 | 63.48 | 61.8 | 62.96 | 62.96 | 844720 |
| 1/10/2018 | 63.41 | 63.67 | 63.09 | 63.57 | 63.57 | 697844 |

## Penelitian Terkait

Penelitian ini dikembangkan dari beberapa referensi penelitian terdahulu yang mempunyai keterkaitan dengan metode dan objek penelitian. Penggunaan referensi ini ditujukan untuk memberikan batasan-batasan terhadap metode yang nantinya akan dikembangkan lebih lanjut. Berikut adalah beberapa penelitian terkait dengan penelitian yang diusulkan.

Bildirici & Ersin (2015) dalam penelitiannya yang berjudul “*Forecasting volatility in oil prices with a class of nonlinear volatility models: smooth transition RBF and MLP neural networks augmented GARCH approach*” yang mengusulkan dua model yaitu, model LSTAR-LST-GARCH ditambah dengan integrasi pecahan dan beberapa istilah daya asimentris (FIGARCH, APGARCH, dan FIAPGARCH), dan model LSTAR-LST-GARCH ditambah dengan Jaringan Syaraf Tiruan MLP dan RBF. Kedua model tersebut dibandingkan dalam hal MSE, RMSE, dan MAPE untuk peramalan volatilitas harga minyak mentah. Hasil menunjukkan bahwa model LSTAR Jaringan Syaraf Tiruan memberikan keuntungan lebih penting dengan nilai RMSE terendah yaitu 0,0000024 dibandingkan model *single-regime baseline* GARCH, diikuti oleh model LSTAR-LST-GARCH dalam hal pemodelan dan peramalan volatilitas harga minyak mentah.

Wulandari *et al*., (2017) dalam jurnalnya yang berjudul “*Crude Oil Price Forecasting Using Radial Basis Function Neural Network Method (RBFNN)*” yang menggunakan model *RBFNN* dengan fungsi aktivasi basis sebagai fungsi aktivasi untuk setiap *neuron* pada lapisan tersembunyi dalam Jaringan Syaraf. Penelitian ini menghasilkan bahwa peramalan harga minyak mentah dunia menggunakan metode *RBFNN* menggunakan 2 input jaringan yaitu *X*t-1 dan *X*t-2 dengan 3 lapisan tersembunyi memiliki tingkat akurasi paling baik dengan nilai MAPE 6,8150%.

Faozi & Sulistijanti (2016) dalam penelitiannya berjudul “Peramalan Harga Minyak Mentah Standar *West Texas Intermediate* dengan Pendekatan Metode ARIMA” yang mengidentifikasi model ARIMA terbaik dalam meramalkan harga minyak mentah dunia skala WTI dengan menggunakan metode *Box-Jenkins* sehingga mendapatkan hasil hanya model ARIMA (1,1,1) yang parameternya signifikan (*P-value* yang berada di bawah level toleransi (α = 0,05)) yang dianggap terbaik dalam peramalan harga minyak mentah dunia WTI.

Dalam jurnalnya yang berjudul “Model Kombinasi ARIMA dalam Peramalan Harga Minyak Mentah Dunia” melakukan penelitian dengan menggunakan data harga minyak mentah dunia dalam runtut waktu bulanan terhitung sejak tahun 2003 – 2017 menggunakan model ARIMA. Hasil yang diperoleh yaitu model ARIMA terbaik 1, 1, 0 dengan nilai RMSE terkecil 8,345%.

Dalam sebuah jurnal dengan judul “Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan dengan Menggunakan Metode *Backpropagation*” menguji 8 model arsitektur yakni 12-5-1, 12-26-1, 12-29-1, 12-35-1, 12-40-1, 12-60-1, 12-70-1, 12-75-1 dengan data bulanan sejak tahun 2014 - 2016. Dari 8 model tersebut, model arsitektur terbaik yaitu 12-70-1 yang menghasilkan tingkat akurasi sebesar 92% dan nilai MSE 0,3659742 dengan menggunakan tingkat *error* 0,001 – 0,05.

Alkronz *et al*., (2019) dalam sebuah jurnal yang berjudul “*Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network*” telah berhasil memperoleh nilai akurasi yang tinggi dalam memprediksi apakah jamur beracun atau tidak menggunakan Jaringan Syaraf Tiruan *Backpropagation* yaitu sebesar 99,25% dengan menerapkan arsitektur *Multi Layer Perceptron* berisikan satu *input layer*, tiga *hidden layer*, dan satu *output layer*.

Zhu *et al*.,(2020) dalam penelitiannya yang berjudul “*Hybrid particle swarm optimization extreme learning machine for daily reference evapotranspiration prediction from limited climatic data*” yang telah menggunakan algoritma PSO untuk meningkatkan kinerja ELM (*Extreme Learning Machine*). Karena ELM tidak memiliki kemampuan generalisasi yang bobot dan biasnya ditentukan secara acak, oleh karena itu PSO dimanfaatkan untuk menemukan bobot dan bias optimal dalam ELM sehingga disebut dengan model *hybrid* PSO-ELM. Model *hybrid* PSO-ELM menunjukkan kinerja yang lebih baik dari pada model lainnya untuk prediksi evapotranspirasi harian.

Makade *et al*.,(2020) dalam artikel jurnal yang berjudul “*Real-time estimation and prediction of the mortality caused due to COVID-19 using particle swarm optimization and finding the most influential parameter*” memanfaatkan algoritma PSO sebagai penentu parameter yang paling berpengaruh terhadap kasus kematian akibat *COVID-19* dan model regresi terbaik dalam prediksi kasus kematian *COVID-19*. Hasil menunjukkan bahwa teknik algoritma PSO mampu menunjukkan kesesuaian prediksi kematian *COVID-19* yang baik dengan rata-rata kesalahan 0,00957%, koefisien determinasi 0,9828 dan RMSE (*Root Mean Square Error*) 4,95201.

Ghasemi, Ebrahim (2017) dalam artikel jurnalnya yang berjudul “*Particle swarm optimization approach for forecasting backbreak induced by bench blasting*” menerapkan teknik *Particle Swarm Optimization* (PSO) untuk memperkirakan kerusakan akibat *bench blasting* berdasarkan parameter peledakan utama yang dapat dikontrol. Peneliti menggunakan dua bentuk model PSO yaitu linier dan kuadrat yang dikembangkan berdasarkan data peledakan tambang tembaga Sungun, Iran. Hasil yang diperoleh menunjukkan bahwa kedua model PSO dapat digunakan untuk memprediksi kerusakan, akan tetapi perbandingan dua model tersebut dalam hal indeks kinerja statistik menunjukkan bahwa model kuadrat memberikan hasil yang lebih baik dari pada model linier. Dapat dilihat bahwa nilai RMSE PSO-*linear* 0,3526 dan RMSE PSO-*quadratic* 0,2788.

# BAB 3 METODE PENELITIAN



## Studi Pendahuluan

Studi pendahuluan yang dilakukan untuk mendapatkan gambaran secara lengkap mengenai permasalahan dalam pengimplementasian pada prediksi harga minyak mentah. Pada tahap ini, dilakukan pengumpulan data, informasi, serta rujukan dari buku, artikel, jurnal dan karya ilmiah lainnya yang berhubungan dengan objek penelitian. Bahan referensi yang digunakan mencakup penemuan atau modifikasi terhadap algoritma JST-BP sebagai algoritma peramalan dan algoritma PSO sebagai algoritma optimasi*.* Dalam upaya meningkatkan nilai akurasi peramalan harga minyak mentah dunia, penelitian ini memiliki 6 tahap utama. Keempat tahapan tersebut yaitu tahap pengambilan data, tahap normalisasi, tahap pengujian parameter, tahap penentuan model, tahap prediksi dan penarikan kesimpulan seperti yang ditunjukkan oleh Gambar 3.1.

|  |
| --- |
| Pengambilan Data  Normalisasi  Penentuan Model  Prediksi  Penarikan Kesimpulan  Pengujian Parameter |

Gambar 3. Tahapan Penelitian

## Pengambilan Data

Dalam penelitian ini, dilakukan pencarian data sebagai objek untuk eksperimen penelitian. Penulis melakukan usaha-usaha untuk mendapatkan data yang dapat dipercaya kebenarannya sehingga informasi yang didapat dapat dipertanggung jawabkan kebenarannya. Jenis data yang digunakan yaitu data kuantitatif berupa numerik harga minyak mentah dunia dalam satuam U.S Dollars. *Dataset* yang digunakan pada penelitian ini adalah *dataset* *public* yaitu *Crude Oil* standar WTI (CL=F) *dataset* yang diperoleh dari situs finance.yahoo.com*.* Data ini telah dicatat oleh yahoo finance secara harian. Sebanyak 1058 data digunakan dalam penelitian terhitung sejak 3 Januari 2017 hingga 31 Maret 2021 yang diakses pada tanggal 5 April 2021 dengan stadar ukuran *West Texas Intermediate* (WTI). Data harian harga minyak mentah dunia sesuai dengan *trading days* yaitu hari Senin – Jum’at dengan jumlah 5 hari kerja dalam satu minggu (tidak termasuk hari libur). Tabel 3.1 menunjukkan beberapa data harga minyak mentah dunia.

Tabel 3. Data Harga Minyak Mentah Dunia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj Close | Volume |
| 2-Jan-2018 | 60.2 | 60.74 | 60.1 | 60.37 | 60.37 | 510313 |
| 3-Jan-2018 | 60.39 | 61.97 | 60.28 | 61.63 | 61.63 | 673859 |
| 4-Jan-2018 | 61.96 | 62.21 | 61.59 | 62.01 | 62.01 | 654363 |
| 5-Jan-2018 | 61.9 | 62.04 | 61.09 | 61.44 | 61.44 | 563036 |
| 7-Jan-2018 | null | null | null | null | null | null |
| 8-Jan-2018 | 61.61 | 61.97 | 61.34 | 61.73 | 61.73 | 616474 |
| 9-Jan-2018 | 61.92 | 63.48 | 61.8 | 62.96 | 62.96 | 844720 |
| 10-Jan-2018 | 63.41 | 63.67 | 63.09 | 63.57 | 63.57 | 697844 |
| 11-Jan-2018 | 63.5 | 64.77 | 63.43 | 63.8 | 63.8 | 876642 |
| 12-Jan-2018 | 63.57 | 64.5 | 63.06 | 64.3 | 64.3 | 678727 |
| 14-Jan-2018 | null | null | null | null | null | null |
| 16-Jan-2018 | 64.43 | 64.89 | 63.39 | 63.73 | 63.73 | 867771 |
| 17-Jan-2018 | 63.89 | 64.17 | 63.31 | 63.97 | 63.97 | 444384 |

Seperti yang terdapat dalam Tabel 3.1, *dataset* Harga Minya Mentah Dunia memiliki lima jenis harga pasar. Harga yang akan digunakan untuk penelitian ini yaitu harga *close* karena merupakan harga yang dapat menjadi acuan untuk prediksi harga *open* di hari berikutnya.

## Normalisasi Data

Normalisasi data merupakan penskalaan nilai atribut data menjadi dalam *range* tertentu. Melakukan tahap normalisasi pada data bertujuan supaya menghasilkan kesesuaian nilai data dengan jarak atau cakupan (interval) yang telah ditentukan pada jaringan (Almas & Setiawan, 2018). Tahapan normalisasi data numerik harga harian minyak mentah dunia diperoleh dari situs finance.yahoo.com. Dalam penelitian ini data akan ditransformasi dalam interval [0,1]. Rumus yang digunakan dalam normalisasi data harian harga minyak mentah dunia yaitu seperti pada Persamaan 26 berikut.

(26)

dimana,

: data normalisasi

: data yang akan dinormalisasi

: data terkecil

: data terbesar

*Dataset* yang telah ternormalisasi kemudian akan melalui tahap implementasi metode yang digunakan yaitu PSO dan JST-BP. Setelah melalui tahap tersebut, data perlu dikembalikan ke bentuk data semula yang dinamakan denormalisasi. Dalam penelitian (Perdana, 2016), denormalisasi berguna untuk memberikan atau mengembalikan data. Adapun rumus denormalisasi dalam interval [0,1] diberikan pada Persamaan 27 berikut.

(27)

dimana,

: nilai data normal

: hasil *output* jaringan

: data terkecil

: data terbesar

## Pengujian Parameter

Tahap pertama yang dilakukan dalam JST-BP yaitu pembagian data. Pembagian data dalam tahap ini akan dibagi menjadi dua data yaitu data pelatihan dan data pengujian. *Datasets* dalam penelitian ini dibagi sebanyak 70% data pelatihan dan 30% data pengujian. Pembagian data ini didasarkan pada penelitian yang telah dilakukan oleh Alkronz *et al*., (2019) yang berhasil mendapatkan tingkat akurasi 99,25%. Pada tahap ini dilakukan pengujian parameter untuk *Backpropagation* dan juga parameter PSO. Parameter *Backpropagation* yang diuji meliputi pengujian jumlah neuron *input* dan neuron *hidden*, jumlah iterasi (*epoch*), dan *learning rate*. Sedangkan pengujian parameter PSO yang diuji meliputi *epoch*, nilai r1 dan r2.

## Penentuan Model

Penentuan model ditentukan menggunakan metode PSO dan proses pelatihan (*training*) algoritma *Backpropagation*. Tahapan ini bertujuan memperoleh model JST-BPterbaik yang telah dioptimasi parameter bobotnya untuk melakukan prediksi. *Dataset* yang digunakan yaitu dataset *training* sebesar 70%. Berikut tahapan proses penentuan model menggunakan PSO dan *training Backpropagation*:

* + - 1. Inisialisasi parameter PSO yaitu kecepatan dan posisi awal pada tiap partikel dalam N dimensi secara acak (*random*). Sehingga semua partikel bergerak menuju titik optimal dengan suatu kecepatan. Kecepatan awal semua partikel diasumsikan sama dengan nol.
      2. Lakukan proses pelatihan (*training*) dengan langkah-langkah seperti pada urutan 3 hingga 5.
      3. *Preprocessing*/Normalisasi

Proses normalisasi yaitu data *input* dan target diskalakan berdasarkan *range* sehingga data menjadi sinkron dan akan memudahkan proses komputasi. Normalisasi dilakukan dengan menggunakan Persamaan 26.

* + - 1. Perancangan Struktur Jaringan

Dalam proses ini menentukan banyaknya lapisan tersembunyi, *neuron*, fungsi aktivasi, dan keluaran yang akan digunakan dalam jaringan. Fungsi aktivasi yang digunakan yaitu fungsi aktivasi sigmoid biner pada Persamaan 8.

* + - 1. Pemilihan Parameter Jaringan

Pemilihan parameter jaringan diantaranya meliputi insialisasi bobot yang bisa ditentukan secara manual maupun *random*, besarnya kinerja tujuan, jumlah maksimum iterasi, besarnya laju pembelajaran (*learning rate*), dan *momentum*. Setelah mentukan parameter jaringan kemudian proses pelatihan dijalankan hingga iterasi JST-BP selesai dan terjadi perubahan bobot jaringan yang nantinya akan digunakan dalam tahap pengujian.

* + - 1. Perbarui *Pbest* dan *Gbest* dengan Persamaan 28 dan 29.

(28)

(29)

* + - 1. Kemudian, perbarui kecepatan menggunakan Persamaan 30.

(30)

dimana,

|  |  |
| --- | --- |
|  | : parameter *inertia weight*, digunakan untuk mengontrol perilaku konvergensi PSO |
|  | : posisi awal partikel i pada d dimensi |
|  | : *pbest* (*personal best*) partikel i pada d dimensi |
|  | : *gbest* (*global best*) pada d dimensi |
| dan | : parameter acak (*random*) antara 0 sampai 1 |
| dan | : konstanta akselerasi (*learning rate*), menunjukkan gerakan iterasi partikel control |

* + - 1. Perbarui posisi menggunakan Persamaan 31 hingga iterasi PSO selesai.

(31)

dimana,

: posisi awal partikel i pada d dimensi

: kecepatan baru partikel i pada d dimensi

* + - 1. Denormalisasi/*Postprocessing*

Hasil dari proses pelatihan jaringan, didapatkan nilai *output* pelatihan yang masih ternormalisasi. Harga tersebut harus dikembalikan ke harga aslinya, inilah yang disebut proses denormalisasi untuk mendapatkan nilai *output* pada *range* yang sebenarnya.

* + - 1. Hitung nilai MAPE (*Mean Absolute Percentage Error*)dan MSE (*Mean Squared Error*)

Menghitung nilai MAPE dan MSE berdasarkan Persamaan 32 dan Persamaan 33 berikut.

(32)

dimana,

: Nilai deret waktu pada periode ke-t

: Nilai ramalan pada periode ke-t

(33)

dimana,

: Jumlah titik data

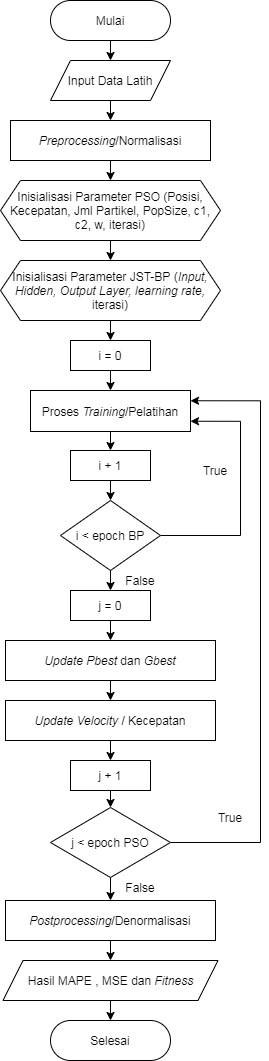
: Nilai yang diamati

: Nilai yang diprediksi

* + - 1. Hitung *fitness* akhirdari partikel menggunakan rumus fungsi *fitness* pada Persamaan 34 berikut.

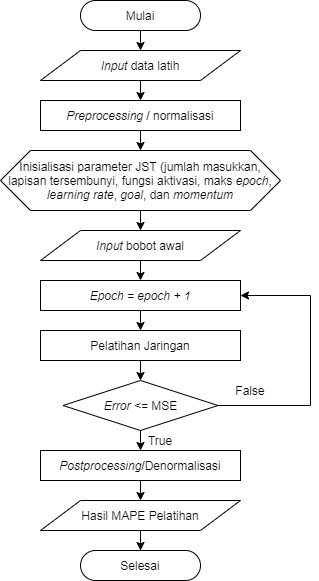
(34)

Berikut *flowchart* penentuan model menggunakan JST-BP dan PSO ditunjukkan oleh Gambar 3.2.



Gambar 3. *Flowchart* Penentuan Model JST-BP dan PSO

Proses pelatihan yang dijalankan dalam tahapan penentuan model ditunjukkan oleh *flowchart* pada Gambar 3.3 berikut.



Gambar 3. *Flowchart* Pelatihan JST-BP

## Prediksi

Tahapan prediksi harga minyak mentah dengan metode JST-BP menggunakan proses pengujian (*testing*). Dalam proses *testing*, *dataset* yang digunakan yaitu sebesear 30% *dataset*. Selanjutnya masuk pada proses pengujian, berikut tahapan pada proses pengujian JST-BP:

* + - * 1. *Preprocessing*/normalisasi

Sama seperti di tahap pelatihan, yaitu penskalaan *input* dan target data uji berdasrkan *range* menggunakan rumus Persamaan 26.

* + - * 1. Pengujian Jaringan

Proses ini dilakukan pengujian jaringan dengan masukan data uji berdasarkan hasil jaringan yang telah dilatih pada tahap pelatihan jaringan dan pembaruan bobot. Proses ini bertujuan untuk mengetahui apakah jaringan dapat bekerja dengan baik dalam memprediksi pola data yang telah dilatih dengan tingkat kesalahan yang terendah.

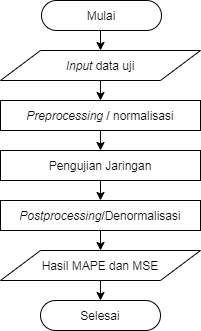
* + - * 1. Denormalisasi/*Postprocessing*

Nilai yang didapat dari proses pengujian kemudian dikembalikan ke nilai aslinya (*denormalisasi*) seperti pada Persamaan 27 untuk mendapatkan nilai *output* dan *range* yang sebenarnya.

* + - * 1. Hitung MAPE dan MSE

Dari hasil *output* di tahap pengujian, kemudian hitung nilai MAPE dan MSE sesuai Persamaan 32 dan Persamaan 33. Hal ini bertujuan untuk mengetahui tingkat kesalahan yang diperoleh dari pengujian jaringan yang telah dioptimasi oleh PSO.

*Flowchart* proses pengujian JST-BP ditunjukkan oleh Gambar 3.4 berikut.



Gambar 3. *Flowchart* Pengujian JST-BP

## Perancangan Sistem

Perancangan sistem dalam penelitian ini menggunakan bahasa pemrograman Python versi 3 untuk penerapan algoritma yang dipakai disisi *back-end*, sedangkan untuk tampilan sistem disisi *front-end* menggunakan *framework* Bootstrap. Python merupakan bahasa *scripting* berorientasi objek yang banyak digunakan dalam pengolahan data karena memiliki banyak keunggulan yaiu fleksibilitas, *open source*, skalabilitas yang baik dan *library* yang berlimpah (Zhang *et al.*, 2017). Selain itu, bahasa Python dipilih karena memiliki karakteristik yang portable sehingga *scripts* Python dapat digunakan pada sistem operasi yang berbeda seperti: Windows, Linux, UNIX, Amigo, Mac OS (Srinath, 2017). Python memiliki sejumlah *Library* yang memudahkan pekerjaan karena kita tidak perlu menulis sema kodenya sendiri, *library* yang sering digunakan untuk pengolahan data yaitu: NumPy, untuk manipulasi data berbasis *array* homogeny; Pandas, untuk menipulasi data heterogen dan berlabel; SciPy, untuk tugas komputasi ilmiah umum; Matplotlib visualisasi kualitas publikasi; IPython, untuk eksekusi interaktif dan berbagi kode; Scikit-Learn, untuk *machine learning* (VanderPlas, 2016).

Pembuatan sistem dilakukan dengan menggunakan *library* Flask dalam *back-end*nya untuk membuat API (*Application Programming Interface*), kemudian menggunakan *library* Matplotlib dan Pandas untuk membuat pengkodean data yang nantinya dapat terhubung dengan API tersebut. Flask merupakan sebuah *web framework* yang ditulis dengan menggunakan bahasa Python dan tergolong sebagai jenis *microframework*. Flask ini berfungsi sebagai *framework* atau kerangka kerja aplikasi dan tampilan dari suatu *web* (Irsyad, 2018).

Bootstrap merupakan salah satu *front-end* *framework* yang digunakan untuk membangun desain *web* secara responsif sehingga dapat dengan mudah menyesuaikan ukuran layar dari *browser* yang kita gunakan baik desktop, tablet maupun *mobile device*. Bootstrap menyediakan HTML, CSS dan *Javascript* siap pakai yang mudah dikembangkan. Selain itu, kerangka Bootstrap dibangun menggunakan Less, sebuah teknologi CSS yang sederhana dan mudah untuk mengakses informasi dan fungsi warna, variable, dan operasi penggunaan (Effendy & Nuqoba, 2016).

## Penarikan Kesimpulan

Berdasarkan studi literatur, pengambilan data, optimasi bobot hingga peramalan harga minyak mentah dunia (WTI), simpulan yang diperoleh adalah hasil perbandingan nilai MAPE dan MSE menggunakan metode JST-BP dan JST-BP dengan optimalisasi algoritma PSO. Perhitungan tingkat akurasi menggunakan rumus pada Persamaan 35.

(35)

# BAB 4 HASIL DAN PEMBAHASAN

## Hasil Penelitian

Pengoptimalan algoritma JST-BP dengan menerapkan PSO pada penentuan bobot terbaik untuk memprediksi harga minyak mentah dunia memiliki enam tahap penelitian. Kelima tahapan tersebut antara lain tahap pengambilan data, normalisasi data, pengujian, penentuan bobot, peramalan dan penarikan kesimpulan. Adapun pembahasan yang lebih lengkap terkait penelitian tersebut diuraikan sebagai berikut.

### Tahap Pengambilan Data

Data yang digunakan dalam penelitian ini adalah *dataset* *Crude Oil* standar WTI (CL=F) yang diperoleh dari situs finance.yahoo.com. *Dataset* ini terdiri dari 1058 *record* data yang bersifat numerik menurut standar ukuran harga minyak WTI. Data diakses pada tanggal 5 April 2021, terhitung sejak 3 Januari 2017 hingga 31 Maret 2021 data disimpan. Tabel 4.1 menunjukkan numerik satuan USD dari *dataset* harga minyak mentah dunia yang tersusun berdasarkan urutan waktu harian.

Tabel 4. Harga Minyak Mentah Dunia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | High | Low | Open | Close | Volume | Adj Close |
| 1/3/2017 | 55.24 | 52.11 | 54.2 | 52.33 | 727793 | 52.33 |
| 1/4/2017 | 53.43 | 52.15 | 52.49 | 53.26 | 512641 | 53.26 |
| 1/5/2017 | 54.12 | 52.79 | 53.39 | 53.76 | 517362 | 53.76 |
| 1/6/2017 | 54.32 | 53.32 | 53.73 | 53.99 | 528333 | 53.99 |
| 1/9/2017 | 53.83 | 51.76 | 53.75 | 51.96 | 564893 | 51.96 |
| 1/10/2017 | 52.37 | 50.71 | 51.83 | 50.82 | 632573 | 50.82 |
| 1/11/2017 | 52.78 | 50.75 | 50.81 | 52.25 | 815846 | 52.25 |
| 1/12/2017 | 53.5 | 52.12 | 52.37 | 53.01 | 660703 | 53.01 |
| 1/13/2017 | 53.17 | 52.27 | 53.05 | 52.37 | 477986 | 52.37 |
| 1/17/2017 | 53.52 | 52.12 | 52.55 | 52.48 | 606023 | 52.48 |
| 1/18/2017 | 52.79 | 50.91 | 52.52 | 51.08 | 236238 | 51.08 |
| 1/19/2017 | 51.87 | 51.02 | 51.39 | 51.37 | 122593 | 51.37 |
| 1/20/2017 | 52.9 | 51.39 | 51.45 | 52.42 | 567231 | 52.42 |
| 1/23/2017 | 53.47 | 52.21 | 53.33 | 52.75 | 455333 | 52.75 |
| 1/24/2017 | 53.56 | 52.67 | 52.86 | 53.18 | 520285 | 53.18 |
| ..... | ..... | ..... | ..... | ..... | ..... | ..... |
| ….. | ….. | ….. | ….. | ….. | ….. | ….. |
| 3/31/2021 | 61.17 | 58.85 | 60.45 | 59.16 | 414162 | 59.16 |

Data diatas belum merupakan data yang dipilih untuk dijadikan data dalam penelitian prediksi. *Dataset* yang telah diambil dari sumber finance.yahoo.com diseleksi terlebih dahulu sebelum proses selanjutnya. Data yang dipilih untuk digunakan sebagai data penelitian yaitu jenis harga minyak mentah *close* yang ditunjukkan oleh Tabel 4.2.

Tabel 4. Harga *Close* Minyak Mentah Dunia

|  |  |
| --- | --- |
| Tanggal | Close |
| 1/3/2017 | 52.33 |
| 1/4/2017 | 53.26 |
| 1/5/2017 | 53.76 |
| 1/6/2017 | 53.99 |
| 1/9/2017 | 51.96 |
| 1/10/2017 | 50.82 |
| 1/11/2017 | 52.25 |
| 1/12/2017 | 53.01 |
| 1/13/2017 | 52.37 |
| 1/17/2017 | 52.48 |
| … | … |
| 3/31/2021 | 59.16 |

### Tahap Normalisasi Data

Supaya metode yang akan digunakan dapat mengenali data sebagai masukan (*input*), maka diperlukannya normalisasi datamenggunakan penskalaan yang telah ditentukan pada Persamaan 3. *Dataset* harga minyak mentah yang diperoleh dari situs finance.yahoo.com telah memiliki format *.csv* yang merupakan standar ASCII (*American Standard Code for Information Interchange*). Format *file .csv* memiliki tingkat kompabilitas yang cukup tinggi, karena hampir semua program pengolahan data sudah mendukung format *.csv* diantaranya yaitu *Microsoft Office, Notepad, MySQL* dan *Oracle*. Oleh karenanya, format *file .csv* digunakan sebagai standar dalam pengolahan data. Hasil dari data yang telah dinormalisasi ditunjukkan pada Tabel 7.

### Tahap Pengujian Parameter

Proses pada tahap pengujian ini dilakukan pengujian parameter dalam metode yang digunakan, sehingga dibagi menjadi dua yaitu pengujian parameter *Backpropagation* dan parameter PSO sebagai berikut:

#### Pengujian Parameter *Backpropagation*

Parameter *Backpropagation* yang diujikan meliputi jumlah neuron *input* dan *hidden*, jumlah iterasi (*epoch*), dan nilai *learning rate*. Kemudian parameter *Backpropagation* yang lain yaitu nilai *output* adalah 1 karena akan hanya ada satu atribut saja yaitu harga minyak mentah.

1. **Pengujian Jumlah Neuron *Input* dan *Hidden*, Iterasi Terbaik**

Tujuan dilakukannya pengujian jumlah neuron *input* dan *hidden layer* ini adalah untuk memperoleh arsitektur JST-BP terbaik berdasarkan nilai *error* yang telah dihitung menggunakan MSE (Almas & Setiawan, 2018). Data yang digunakan dalam penelitian ini yaitu data *training* sebanyak 70% dari total data 1058 *dataset*. Data nilai *input* yang diuji adalah 1, 3, 5, 7, dan 10 sedangkan data nilai *hidden* yang digunakan yaitu 1 hingga 10. Jumlah iterasi yang diujicobakan yaitu 20, 40, 60, 80, 100. Nilai *learning rate* sementara yang digunakan sebesar 0,1. Kemudian dilakukan *running* program sesuai percobaan, sehingga didapatkan nilai MSE dan MAPE. Dari 250 hasil MSE dan MAPE yang diperoleh dari ke 250 percobaan, diambil nilai yang sesuai, ditunjukkan oleh Tabel 4.3.

Tabel 4.Hasil Pengujian *Input*, *Hidden Layer* dan Iterasi BP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Input | Hidden | Output | Iterasi | MSE | MAPE |
| 1 | 10 | 2 | 1 | 80 | 3.078677 | 2.356216 |
| 2 | 7 | 1 | 1 | 60 | 3.148959 | 2.389554 |
| 3 | 7 | 4 | 1 | 80 | 3.263824 | 2.356018 |
| 4 | 1 | 7 | 1 | 60 | 3.277772 | 2.437512 |
| 5 | 7 | 2 | 1 | 60 | 3.298756 | 2.393545 |
| 6 | 10 | 1 | 1 | 60 | 3.466233 | 2.50332 |
| 7 | 5 | 7 | 1 | 80 | 3.487729 | 2.43617 |
| 8 | 3 | 7 | 1 | 60 | 3.493921 | 2.393767 |
| 9 | 5 | 4 | 1 | 60 | 3.526741 | 2.415525 |
| 10 | 5 | 3 | 1 | 60 | 3.5444 | 2.489686 |

1. **Pengujian Nilai *Learning Rate* Terbaik**

Dalam pengujian ini, akan dicari nilai *learning rate* yang menghasilkan nilai MSE dan MAPE terendah. Pengujian dilakukan dengan menggunakan nilai *learning rate* 0,1 hingga 0,9 pada 10 hasil pengujian sebelumnya. Dari pengujian ini didapatkan 90 nilai MAPE dan MSE, yang kemudian dipilih empat data dengan nilai MSE rendah. Hasil pengujian *learning rate* dapat dilihat pada Tabel 4.4.

Tabel 4. Hasil Pengujian *Learning Rate* BP

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Learning rate | Input | Hidden | Output | Iterasi | MSE | MAPE |
| 1 | 0,2 | 5 | 3 | 1 | 60 | 3.039764635 | 2.259384714 |
| 2 | 0,1 | 10 | 2 | 1 | 80 | 3.083131629 | 2.359136482 |
| 3 | 0,1 | 7 | 1 | 1 | 60 | 3.152367238 | 2.392114299 |
| 4 | 0,6 | 10 | 1 | 1 | 80 | 3.078266013 | 2.381523427 |

#### Pengujian Parameter PSO

Pada pengujian parameter PSO yang diujikan yaitu nilai r1, r2 dan jumlah iterasi (*epoch*). Parameter PSO yang lain menggunakan nilai yang telah digunakan oleh penelitian sebelumnya yaitu nilai parameter c1 dan c2 sebesar 1 dan 1,5 (Ika & Setiawan, 2018). Kemudian parameter bobot inersia pada PSO menggunakan nilai 0,5 yang telah digunakan oleh (Sari, 2019) dengan hasil nilai MAPE sebesar 18,3583% untuk prediksi kualitas udara. Jumlah *popsize* yang digunakan sebanyak 5 (Maori, 2017). Jumlah partikel yang digunakan dalam PSO ditentukan dari arsitektur jaringan JST-BP yang digunakan yaitu jumlah *input layer* dikali dengan jumlah *hidden layer*.

**Pengujian Jumlah Iterasi (*epoch*)**

Hasil dari keempat pengujian parameter *Backpropagation* yang telah dilakukan sebelumnya, kemudian diuji dengan penambahan parameter PSO. Pengujian jumlah iterasi PSO dilakukan dengan mengujikan nilai iterasi 1 hingga 50. Tabel 4.5 menunjukkan hasil dari pengujian jumlah iterasi PSO.

Tabel 4. Hasil Pengujian Jumlah Iterasi PSO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hasil | Iterasi | Parameter (Input; Hidden; Output; Learning Rate) | | | |
| (5; 3; 1; 0.2) | (7; 1; 1; 0.1) | (10; 1; 1; 0.6) | (10; 2; 1; 0.1) |
| MSE | 1 | 2.874706111 | 3.08029324 | 3.07826601 | 3.083131629 |
| MAPE | 2.200273749 | 2.36843795 | 2.38152343 | 2.359136482 |
|  |  |  |  |  |  |
| MSE | 2 | 2.392364447 | 2.89720981 | 3.03652345 | 2.982981346 |
| MAPE | 2.001933937 | 2.29268813 | 2.36103326 | 2.318089388 |
|  |  |  |  |  |  |
| MSE | 3 | 2.266719766 | 2.82466208 | 2.95552567 | 2.744944593 |
| MAPE | 1.952414178 | 2.25420575 | 2.3241376 | 2.248877914 |
|  |  |  |  |  |  |
| MSE | 4 | 2.342322389 | 2.77757638 | 2.8942179 | 2.713878767 |
| MAPE | 1.981272978 | 2.23682281 | 2.29221354 | 2.226684883 |
|  |  |  |  |  |  |
| MSE | 5 | 2.227067113 | 2.77602999 | 2.88886386 | 2.738036715 |
| MAPE | 1.968335245 | 2.23660914 | 2.28529946 | 2.230819908 |
| … | … | … | … | … | … |
| … | … | … | … | … | … |
| MSE | 50 | 2.409583535 | 2.39177319 | 2.36514092 | 2.066895665 |
| MAPE | 1.997649519 | 2.05360948 | 2.03200573 | 1.911476607 |

Dari hasil percobaan, kemudian dipilih jumlah iterasi PSO dengan nilai MSE dan MAPE terendah sesuai dengan keempat jaringan *Backpropagation*. Hasil pemilihan iterasi terbaik dapat dilihat pada Tabel 4.6.

Tabel 4. Iterasi Terbaik

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch | Input | Hidden | Output | Lr | t | c1 | c2 | w | MSE PSO | MAPE PSO |
| 60 | 5 | 3 | 1 | 0.2 | 16 | 1 | 1.5 | 0.5 | 1.967375196 | 1.85356127 |
| 60 | 7 | 1 | 0.1 | 50 | 2.391773192 | 2.053609476 |
| 80 | 10 | 1 | 0.6 | 36 | 2.355958813 | 2.033130932 |
| 80 | 10 | 2 | 0.1 | 49 | 2.066192509 | 1.911867578 |

**Pengujian Nilai r1 dan r2**

Dalam pengujian nilai r1 dan r2, jaringan *Backpropagation* yang digunakan yaitu hasil terbaik dari pengujian parameter *Backpropagation* dan iterasi PSO. Hasil MSE dan MAPE PSO terbaik ditunjukkan oleh parameter jaringan dengan sebagai berikut; 0,2 *learning rate*; 5 *input layer*; 3 *hidden layer*; 1 *output layer*; dan 60 iterasi (*epoch*) dengan hasil MSE 1,967375196 dan MAPE 1,85356127. Karena parameter r1 dan r2 berdasarkan nilai acak (*random*), pengujian dipilih nilai acak dengan hasil MAPE dan MSE yang terendah. Hasil dari pengujian nilai r1 dan r2 yang berhasil dipilih ditunjukkan pada Tabel 4.7 yang terdapat nilai r1 dan r2 masing-masing berjumlah 75 nilai r.

Tabel 4.Hasil Nilai r1 dan r2 Terbaik

|  |  |
| --- | --- |
| List r1 | List r2 |
| [0.02339881415026357, 0.8879191269632195, 0.9632800020094943, 0.10266577517415809, 0.14650702407980176, 0.376193776215101, ….. , 0.9557995894745693] | [0.35488007726168247, 0.7862765993455152, 0.4336464325856312, 0.45751702992693666, 0.33425441322266836, 0.021979474682622957, 0.827935680082634, ….. , 0.9221782820590523] |

### Tahap Penentuan Model

Penentuan model JST-BP dalam penelitian ini menggunakan penambahan metode optimasi yaitu PSO dengan data *training* sebanyak 70% dari total data 1058 *dataset*. Hasil pengujian parameter terbaik yang telah dilakukan sebelumnya, digunakan dalam tahap ini. Penentuan parameter terbaik yang digunakan diperoleh dari hasil nilai MSE dan MAPE terkecil juga estimasi waktu yang digunakan untuk proses *running* program. Nilai parameter terbaik yang dihasilkan dalam proses *training* data ditunjukkan oleh Tabel 4.8 berikut.

Tabel 4. Parameter PSO dan JST-BP

|  |  |
| --- | --- |
| Parameter | Nilai/Jumlah |
| **PSO** | |
| Jumlah partikel | 15 |
| *Popsize* | 5 |
| c1 | 1 |
| c2 | 1,5 |
| Bobot inersia (*w*) | 0,5 |
| *Epoch* | 16 |
| **JST-BP** | |
| *Input layer* | 5 |
| *Hidden layer* | 3 |
| *Output layer* | 1 |
| *Epoch* | 60 |
| *Learning rate* | 0,2 |

Hasil perbandingan nilai MSE dan MAPE untuk proses *training* diberikan dalam Tabel 4.9.

Tabel 4. Hasil MSE dan MAPE Proses *Training*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Metode | MSE | MAPE |
| 1 | JST-BP | 2,25938 | 3,03976. |
| 2 | JST-BP - PSO | 1,96737 | 1,85356 |

Nilai MSE dan MAPE yang dihasilkan dari proses *training* dalam pencarian model parameter terbaik menggunakan metode JST-BP dan PSO yaitu 1,96737 dan 1,85356. Sedangkan nilai MSE dan MAPE dengan hanya menggunakan metode JST-BP dalam proses *training* yaitu 2,25938 dan 3,03976. Hasil ini menunjukkan bahwa PSO telah berhasil mengoptimasi JST-BP untuk mendapatkan nilai *error* terkecil.

### Tahap Prediksi

Proses prediksi yang dilakukan menggunakan *datasets* sebanyak 30% dari total data yang digunakan. Optimasi yang dilakukan PSO dalam JST-BP bertujuan untuk menghasilkan tingkat *error* paling rendah untuk prediksi harga minyak mentah. PSO melakukan optimasi pada parameter JST-BP yaitu bobot awal sehingga diharapkan dapat meningkatkan akurasi prediksi. Proses prediksi menggunakan tahapan sesuai dengan tahapan pengujian (*testing*) JST-BP berdasarkan hasil model dari proses pelatihan (*training*) JST-BP dan PSO. Tahapan yang dilakukan yaitu dimulai dari normalisasi data, kemudian pengujian jaringan atau model hingga denormalisasi. Setelah tahapan *testing* tersebut selesai, selanjutnya yaitu dilakukan perhitungan tingkat akurasi, MSE dan MAPE. Nilai tingkat akurasi, MSE dan MAPE yang dihasilkan dari proses ini diberikan pada Tabel 4.10.

Tabel 4. Hasil Prediksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Metode | Tingkat Akurasi | MSE | MAPE |
| 1 | JST-BP | 93,71677% | 13,86345 | 6,28323 |
| 2 | JST-BP - PSO | 94,97993% | 7,15827 | 5,02007 |

Hasil yang diperoleh dalam proses prediksi yaitu nilai tingkat akurasi, MSE dan MAPE. Nilai tingkat akurasi, MSE dan MAPE yang dihasilkan proses prediksi menggunakan metode JST-BP dan PSO masing-masing yaitu 94,97993%, 7,15827, dan 5,02007. Sedangkan hasil tingkat akurasi, MSE dan MAPE yang hanya menggunakan metode JST-BP yaitu 93,71677%, 13,86345, 6,28323.

## Implementasi Sistem

### Tahap Perancangan User Interface

Pembuatan *user interface* yang digunakan sebagai sistem berbasis *web* menggunakan bahasa HTML (*Hyper Text Markup Language*) dengan *framework* Bootstrap. Sementara untuk perhitungan metode menggunakan bahasa pemrograman *python* *versi* 3 dengan *library* Matplotlib, Pandas dan Flask. Terdapat beberapa menu yang ada dalam *user interface*, diantaranya yaitu:

* + - 1. Beranda

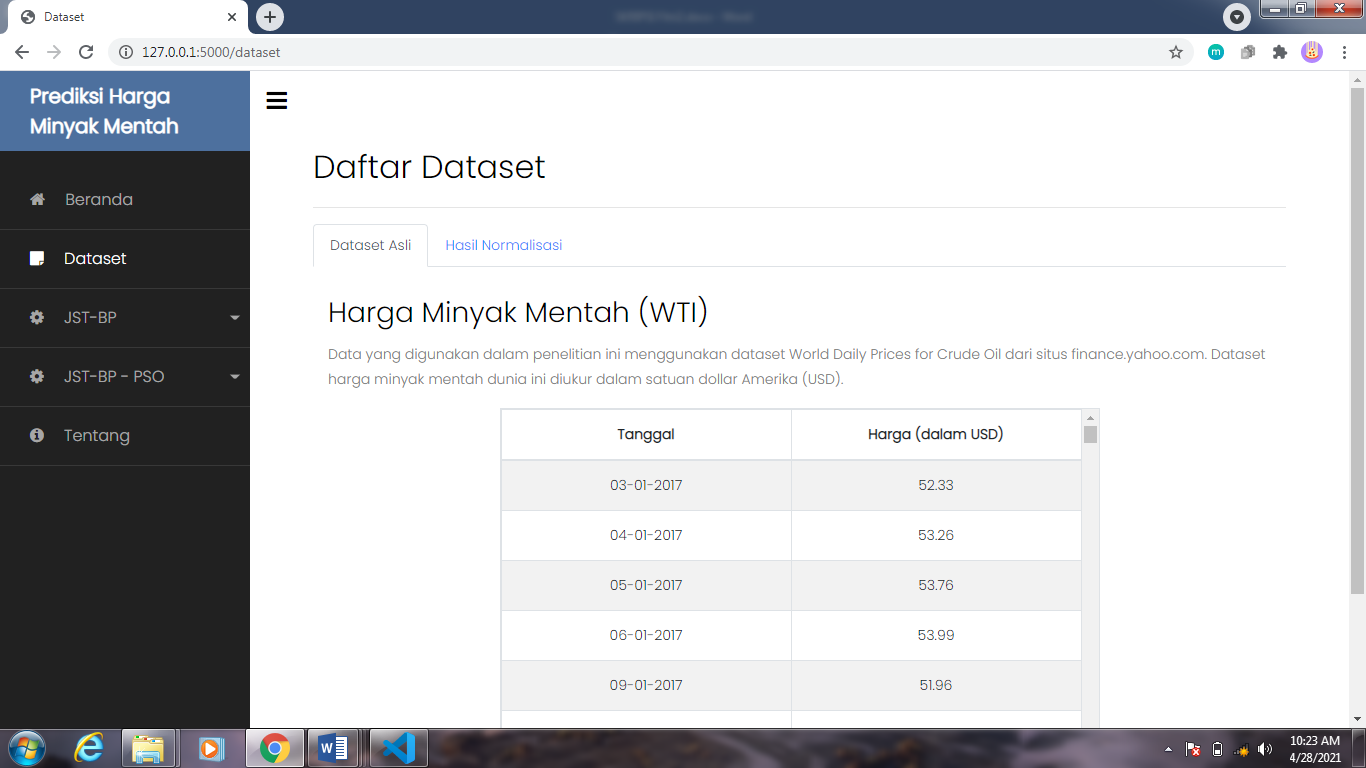
Tampilan yang ada pada menu Beranda adalah judul skripsi, logo UNNES, nama beserta NIM, dan nama instansi terkait. Tampilan Beranda ditunjukkan pada Gambar 4.1.



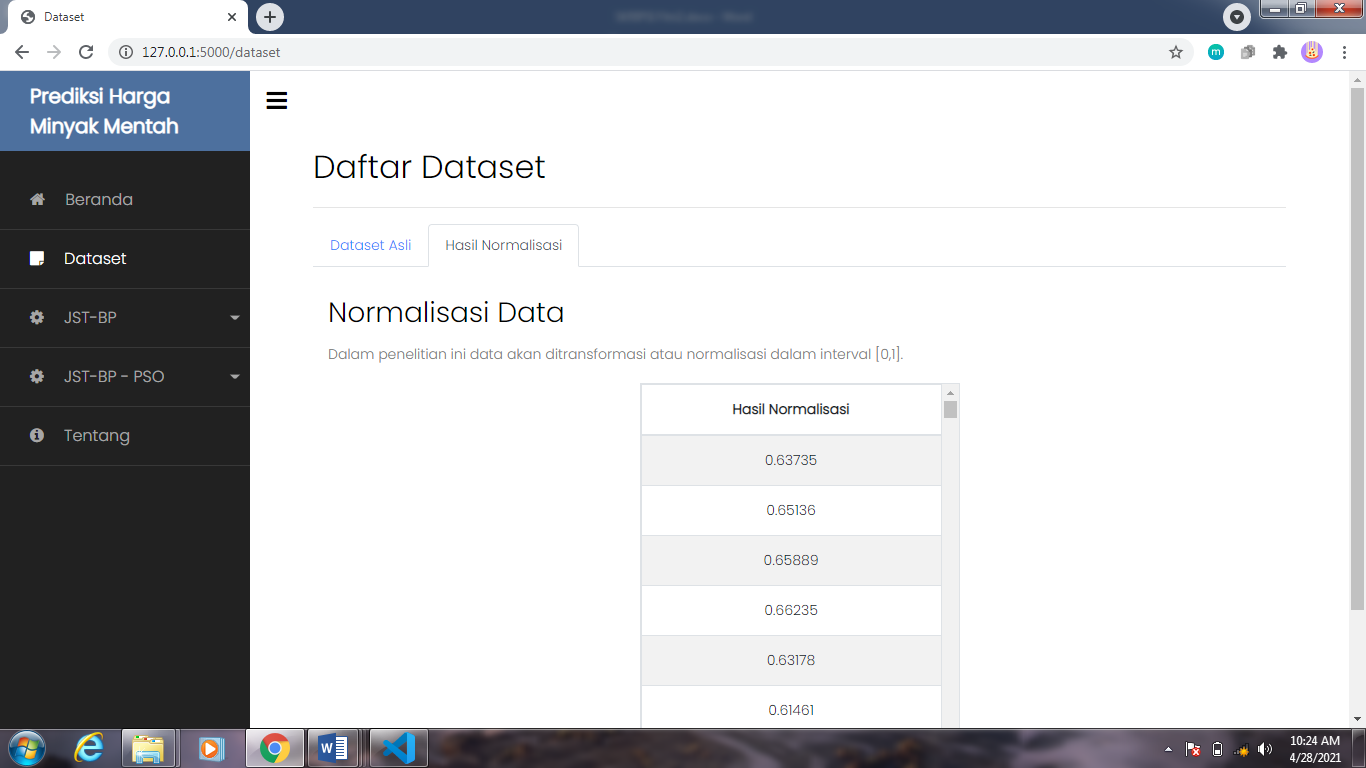
Gambar 4. Tampilan Halaman Beranda

* + - 1. Dataset

Menu Dataset terdapat sub menu bagian yaitu Dataset Asli dan Hasil Normalisasi. Dataset Asli menampilkan tabel *dataset* Harga Minyak Mentah tanggal 3 Januari 2017 hingga 31 Maret 2021. Kemudian untuk sub menu Hasil Normalisasi menampilkan tabel yang berisi perhitungan normalisasi *dataset* dalam interval [0,1]. Tampilan sub menu Dataset Asli dan Hasil Normalisasi ditunjukkan pada Gambar 4.2 dan Gambar 4.3.



Gambar 4. Tampilan Sub Menu Dataset Asli



Gambar 4. Tampilan Sub Menu Hasil Normalisasi

*Dataset* yang memiliki format *.csv* ditampilkan dalam bentuk *list* dalam tabel yang diurutkan menurut tanggal. Berikut kode fungsi untuk menampilkan *dataset* ada pada Tabel 4.11.

Tabel 4. Kode Menampilkan *Dataset*

|  |
| --- |
| def data():      data = pd.read\_csv('dataset/wti.csv')      data['Date'] = pd.to\_datetime(data['Date'])      data['Date'] = data['Date'].dt.strftime("%d-%m-%Y")      length = len(data) |

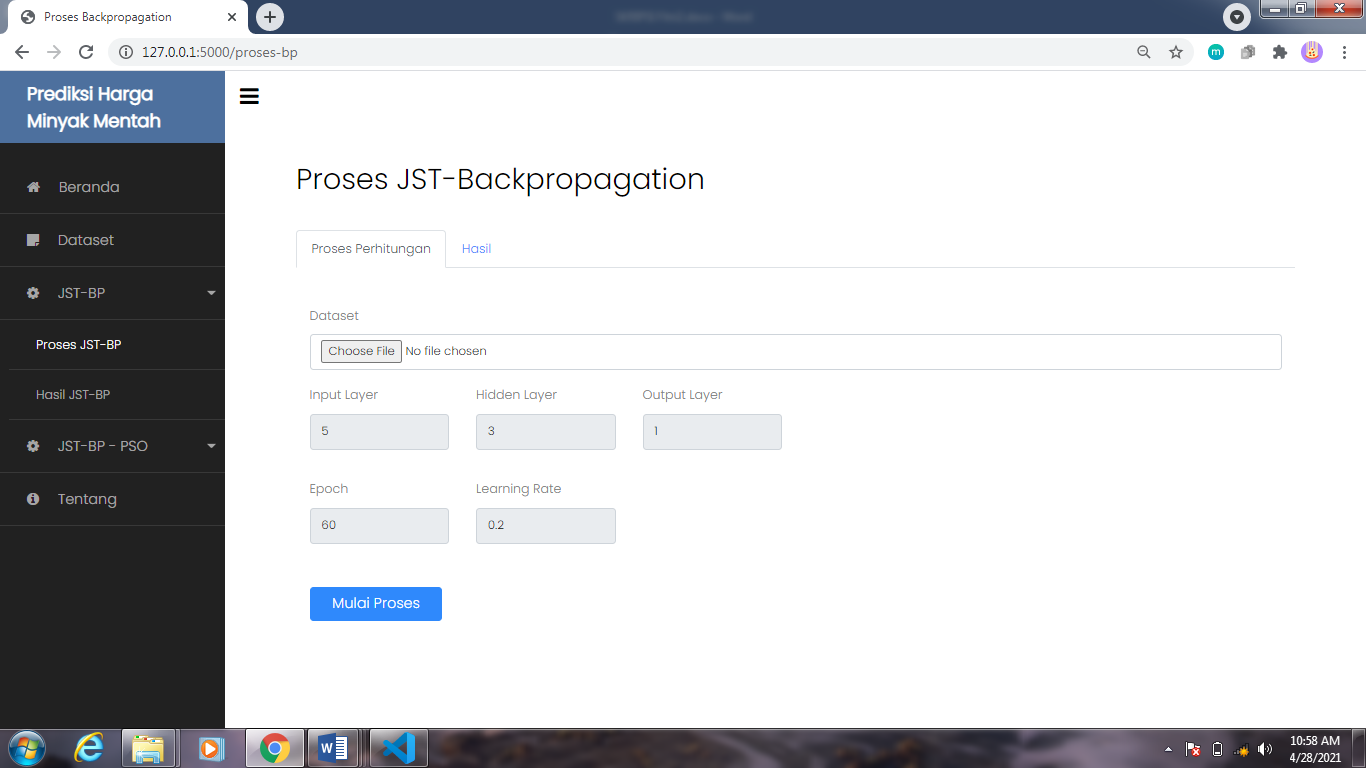
Untuk melakukan normalisasi *dataset* dalam rentang [0,1] sesuai dengan Persamaan 26 perlu dilakukannya pencarian nilai minimal dan maksimal *dataset* terlebih dahulu kemudian rumus normalisasi. Kode normalisasi data ditunjukkan oleh Tabel 4.12 berikut.

Tabel 4. Kode Normalisasi *Dataset*

|  |
| --- |
| data\_value = data.iloc[:, 1].values      max\_value = max(data\_value)      min\_value = min(data\_value)      normalized\_data = list()      for x in range(len(data\_value)):          norm\_value = (data\_value[x] - min\_value) / (max\_value - min\_value)          normalized\_data.append(norm\_value)      data['Normalized'] = pd.DataFrame(normalized\_data, columns=['Normalized'])      data = data.round(5) |

* + - 1. JST-BP

Menu JST-BP dalam sistem Prediksi Harga Minyak Mentah terbagi 2 menu lain yaitu menu Proses JST-BP dan Hasil JST-BP. Menu Proses JST-BP merupakan bagian yang digunakan untuk prediksi Harga Minyak Mentah yang hanya menggunakan metode JST-BP. Di dalam menu tersebut terdapat form *input* dataset sesuai format yaitu .csv dan beberapa parameter terbaik yang telah dihasilkan dari tahap pengujian. Kemudian, pada menu Hasil JST-BP memberikan tampilan hasil *training* dan *testing* prediksi JST-BP yang didalamnya sudah terdapat grafik MSE dan MAPE, nilai MSE dan Mape, grafik perbandingan harga prediksi, dan nilai tingkat akurasi. Tampilan menu Proses JST-BP Gambar 4.4.

Gambar 4. Tampilan Menu Proses JST-BP

Sistem ini memiliki nilai parameter sebagai data yang dibutuhkan untuk proses perhitungan. Parameter yang diperlukan telah terdapat dalam form tampilan proses yaitu *FormData* yang kemudian akan diambil untuk proses *training*. Kode program yang digunakan terdapat dalam Tabel 4.13.

Tabel 4. Kode Fungsi untuk mengambil nilai parameter

|  |
| --- |
| var data = new FormData();  data.append('input\_layer', $('#input\_layer').val());  data.append('hidden\_layer', $('#hidden\_layer').val());  data.append('output\_layer', $('#output\_layer').val());  data.append('epoch', $('#epoch').val());  data.append('learning\_rate', $('#learning\_rate').val());  data.append('dataset\_source', $('#dataset\_source')[0].files[0]); |

Data nilai parameter yang telah diambil dikirim dari AJAX untuk dilakukannya proses *training* kemudian ditampilkan hasilnya dalam bentuk JSON. Data JSON inilah yang nantinya akan diteruskan ke tampilan *user interface*. Fungsi tersebut dapat dilihat pada Tabel 4.14.

Tabel 4. Kode fungsi AJAX untuk mengirim, memproses dan menampilkan data

|  |
| --- |
| $.ajax({          type: 'POST',          url: '/training-bp',          data: data,          contentType: false,          cache: false,          processData: false,          async: true,          success: function (data) {                      $('#mse\_train').html(data.mse);                      $('#mape\_train').html(data.mape);                      $('#akurasi').html(data.akurasi);                      $('#mse\_test').html(data.mse\_test);                      $('#mape\_test').html(data.mape\_test);                      $('#result\_loading').hide();                      $('#img\_mse').attr('src','{{url\_for('static', filename='images/mse\_bp.png')}}');                      $('#img\_mape').attr('src','{{url\_for('static', filename='images/mape\_bp.png')}}');                      $('#img\_prediksi').attr('src','{{url\_for('static', filename='images/prediksi\_bp.png')}}');                      $('#result\_field').show();                  },         error: function (data) {         console.log(data);                  }              }); |

Dalam menu Hasil JST-BP, sistem menampilkan hasil dari proses *training* dan *testing*. Tampilan hasil proses *training* dan *testing* JST-BP ditunjukkan oleh Gambar 4.5 dan Gambar 4.6 berikut.



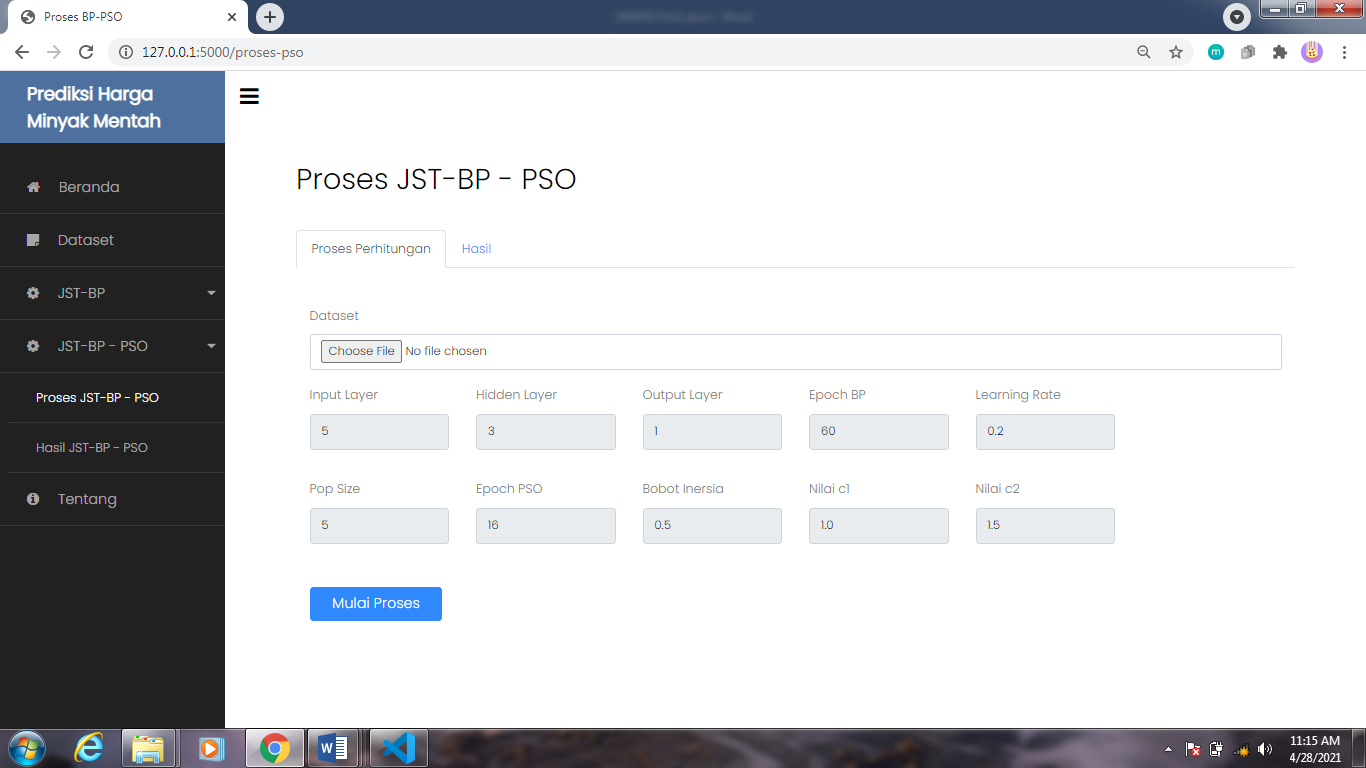
Gambar 4. Tampilan Menu Hasil *Training* JST-BP



Gambar 4. Tampilan Menu Hasil *Testing* JST-BP

* + - 1. JST-BP – PSO

Dadalm sistem Prediksi Harga Minyak Mentah disini yang paling utama yaitu menggunakan penggabungan metode JST-BP dan PSO. Sama seperti menu JST-BP, dalam menu JST-BP – PSO ini juga terdapat menu Proses JST-BP – PSO dan Hasil JST-BP – PSO. Dimana dalam menu Proses JST-BP – PSO sudah terdapat parameter JST-BP dan PSO terbaik dari pengujian parameter. Tampilan menu Proses JST-BP – PSO Gambar 4.7.

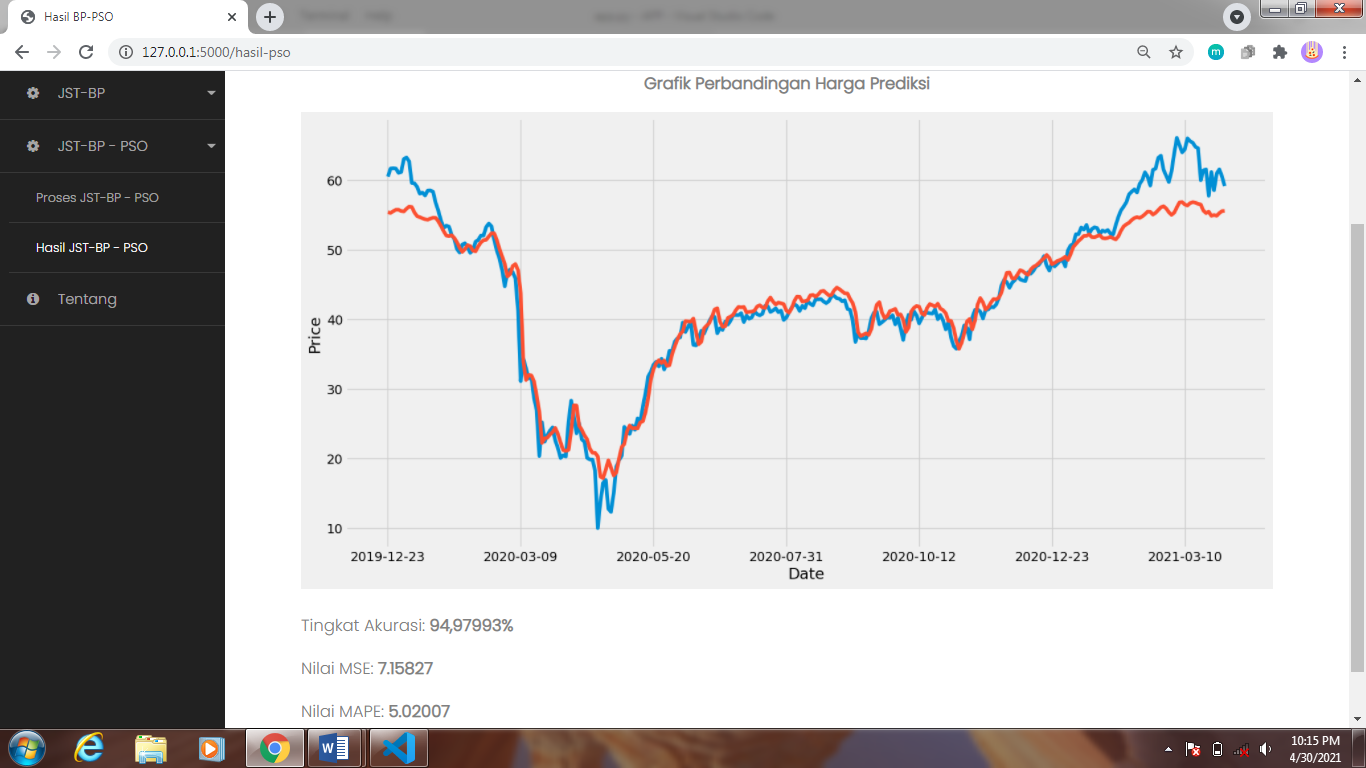


Gambar 4. Tampilan Proses JST-BP – PSO

Menu Hasil JST-BP – PSO dalam sistem memberikan tampilan hasil dari proses *training* dan *testing*. Tampilan hasil proses *training* dan *testing* JST-BP – PSO ditunjukkan oleh Gambar 4.8 dan Gambar 4.9 berikut.



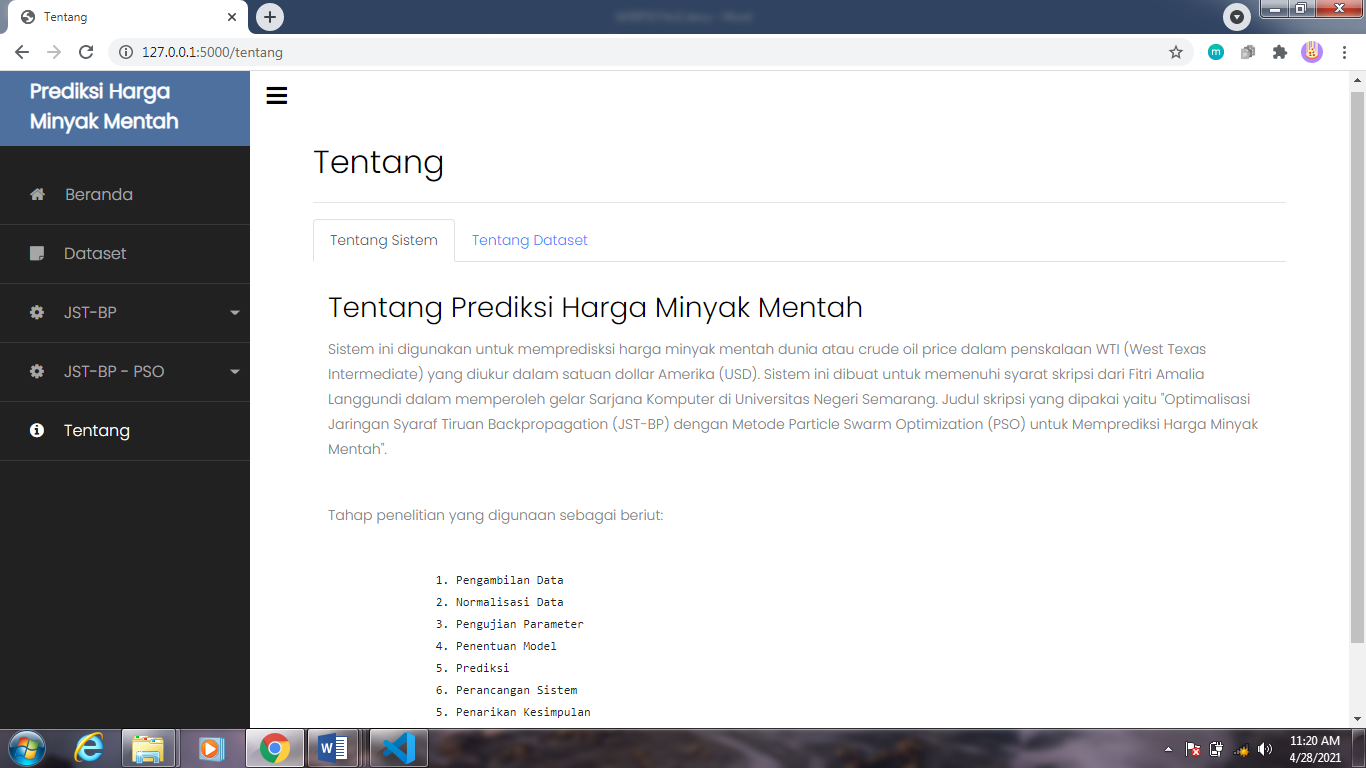
Gambar 4. Tampilan Hasil Proses *Training* JST-BP – PSO



Gambar 4. Tampilan Hasil Proses *Testing* JST-BP – PSO

* + - 1. Tentang

Menu Tentang berisi penjelasan singkat mengenai sistem yang dibuat dan tahap penelitian yang digunakan. Selain itu juga menjelaskan mengenai *dataset* yang digunakan dalam sistem. Tampilan halaman Tentang dapat dilihat pada Gambar 4.10.



Gambar 4. Tampilan Halaman Tentang

### Tahap Implementasi Algoritma

Tahap implementasi algoritma merupakan tahap pembuatan program utama yaitu sekumpulan kode Python untuk menjalankan keseluruhan algoritma mulai dari proses pengambilan data hingga mendapatkan hasil MSE, MAPE dan akurasi. Dalam pembuatan program ini, menggunakan beberapa *library* yaitu *Pandas*, *NumPy*, dan *Matplotlib*. Program utama dibungkus dalam satu file bernama BPxPSO.py. dalam file BPxPSO.pyterdapat beberapa *class*, yaitu:

*Neuron*

*Class Neuron* berisi satu fungsi bernama *\_init­\_ (self, weight)* yang bertugas melakukan inisialisasi *layer* untuk arsitektur jaringan JST-BP. Kode untuk *class Neuron* ditampilkan pada Tabel 4.15.

Tabel 4. *Class Neuron*

|  |
| --- |
| class Neuron:      def \_\_init\_\_(self, weight):          self.weight = weight          self.output = list()          self.delta = list() |

Pada kode di Tabel 4.15 inisialisasi *layer* yang berisi nilai parameter bobot, bias, *output* dan delta.

*Backpropagation*

*Class Backpropagation* berisi beberapa fungsi yangberfungsi untuk perhitungan prediksi harga minyak mentah menggunakan JST-BP. Potongan kode *class Backpropagation* ditampilkan pada Tabel 4.16.

|  |
| --- |
| class Backpropagation:      def \_\_init\_\_(self, input, hidden, output, learning\_rate):          self.INPUT\_LAYER = input          self.HIDDEN\_LAYER = hidden          self.OUTPUT\_LAYER = output          self.LEARNING\_RATE = learning\_rate      def initWeight(self, partikel):          layer = list()          partikel\_dimens\_idx = 0          input\_to\_hidden = list()          for i in range(self.HIDDEN\_LAYER):              w = list()              for j in range(self.INPUT\_LAYER):                  w.append(partikel[partikel\_dimens\_idx])                  partikel\_dimens\_idx += 1              w.append(1.0)              input\_to\_hidden.append(Neuron(w))          layer.append(input\_to\_hidden) |

Tabel 4. Potongan Kode *Class Backpropagation*

Fungsi yang terdapat di dalam *class Backpropagation* diantaranya yaitu inisialisasi parameter, inisialisasi bobot, penjumlahan sinyal, aktivasi, *feed forward*, *backward*, hitung faktor kesalahan, perubahan bobot, *training*, *testing*. Kode inilah yang menghasilkan nilai MSE, MAPE, dan akurasi prediksi menggunakan JST-BP.

*BackpropagationParticle*

*Class BackpropagationParticle* digunakan untuk keperluan inisialisasi partikel yaitu posisi partikel, kecepatan partikel dan nilai menghitung nilai *fitness* disetiap iterasi. *Class* ini digunakan pada saat proses *training* JST-BP – PSO, potongan kode program *Class BackpropagationParticle* diberikan pada Tabel 4.17 berikut.

Tabel 4. Potongan Kode *Class BackpropagationParticle*

|  |
| --- |
| class BackpropagationParticle():      def \_\_init\_\_(self, particle\_size, position, x\_train, y\_train, x\_test, y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, epoch, input\_layer, hidden\_layer, output\_layer, lr):          self.x\_train, self.y\_train = x\_train, y\_train          self.x\_test, self.y\_test = x\_test, y\_test          self.max\_val, self.min\_val, self.max\_val\_test, self.min\_val\_test = max\_val, min\_val, max\_val\_test, min\_val\_test          self.epoch = epoch          self.position = position[:particle\_size]          self.velocity = [0 for i in range(particle\_size)]          self.input\_layer = input\_layer          self.hidden\_layer = hidden\_layer          self.output\_layer = output\_layer          self.learning\_rate = lr          self.calculate\_fitness()      def calculate\_fitness(self):          particle\_position = self.position          backPro = Backpropagation(self.input\_layer, self.hidden\_layer, self.output\_layer, self.learning\_rate)          layer = backPro.initWeight(particle\_position)          mape, hasil, mse, mselist, mapelist = backPro.training(self.x\_train, self.y\_train, self.max\_val, self.min\_val, self.epoch)          mape\_test, mse\_test, hasil\_test, akurasi = backPro.testing(self.x\_test, self.y\_test, self.max\_val\_test, self.min\_val\_test) |

*PSOxBackpro*

*Class PSOxBackpro* merupakan kode untuk melakukan prediksi harga minyak mentah yang menggunakan JST-BP – PSO. Dalam *class PSOxBackpro* terdapat pula beberapa fungsi sebagai urutan algortima. Fungsi-fungsi tersebut diantaranya yaitu inisialisasi parameter JST-BP dan PSO, pengambilan *Gbest, update* kecepatan, posisi, *Pbest*, pencarian nilai *fitness* dan optimalisasi. Potongan kode dalam *class PSOxBackpro* ditunjukkan pada Tabel 4.18.

Tabel 4. Potongan Kode *Class PSOxBackpro*

|  |
| --- |
| Class PSOxBackpro:      def \_\_init\_\_(self, pop\_size, particle\_size, k=None):          self.k = k          self.pop\_size, self.particle\_size = pop\_size, particle\_size      def set\_backpro\_param(self, x\_train, y\_train, x\_test, y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, epoch, input\_layer, hidden\_layer, output\_layer, lr):          self.x\_train, self.y\_train = x\_train, y\_train          self.x\_test, self.y\_test = x\_test, y\_test          self.max\_val, self.min\_val, self.max\_val\_test, self.min\_val\_test = max\_val, min\_val, max\_val\_test, min\_val\_test          self.input\_layer, self.hidden\_layer, self.output\_layer = input\_layer, hidden\_layer, output\_layer          self.learning\_rate = lr          self.epoch = epoch        def initPops(self):          self.pops = [] |

## Pembahasan

Penelitian ini menggunakan algoritma PSO untuk mengoptimalkan kinerja JST-BP pada bobot (*w*) awal proses *training* sehingga menghasilkan tingkat *error* rendah dan akurasi yang lebih baik. *Dataset* yang digunakan dalam penelitian ini adalah *Crude Oil* standar WTI (CL=F) *dataset* yang diperoleh dari situs finance.yahoo.com*.* Data ini telah dicatat oleh yahoo finance secara harian dan bersifat publik sehingga dapat diakses oleh siapa saja. Sebanyak 1058 data digunakan dalam penelitian terhitung sejak 3 Januari 2017 hingga 31 Maret 2021 yang diakses pada tanggal 5 April 2021 dengan stadar ukuran *West Texas Intermediate* (WTI). Data harian harga minyak mentah dunia sesuai dengan *trading days* yaitu hari Senin – Jum’at dengan jumlah 5 hari kerja dalam satu minggu (tidak termasuk hari libur).

Pada penelitian ini dilakukan perbandingan MSE, MAPE dan tingkat akurasi yang dihasilkan oleh JST-BP tanpa menggunakan PSO dengan JST-BP yang menggunakan optimalisasi PSO untuk mengetahui pengaruh optimalisasi PSO dalam melakukan peningkatan akurasi pada JST-BP.

Algoritma PSO digunakan untuk optimasi parameter bobot (*w*) awal pada proses *training* JST-BP. Pada setiap iterasi *training* JST-BP dan PSO dijalankan, akan didapatkan posisi terbaik yang kemudian dilanjutkan dengan *update* bobot, kecepatan, posisi, *Pbest* dan *fitness* untuk penentuan *Gbest* hingga iterasi selesai. Proses yang dilakukan tersebut dinamakan proses optimalisasi parameter JST-BP menggunakan PSO. Hasil jaringan dan parameter yang didapatkan, selanjutnya diuji menggunakan proses *testing*.

Penelitian ini mencatat hasil MSE, MAPE dan tingkat akurasi yang dihasilkan dari tiap proses perhitungan JST-BP maupun JST-BP – PSO yang telah dilakukan. Adapun hasil tersebut dapat dilihat pada Tabel 4.19.

Tabel 4. Hasil Tiap Proses Perhitungan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Metode | MSE | | MAPE | | Akurasi |
| *Training* | *Testing* | *Training* | *Testing* |
| 1 | JST-BP | 2,25938 | 13,86345 | 3,03976 | 6,28323 | 93,71677% |
| 2 | JST-BP - PSO | 1,96737 | 7,15827 | 1,85356 | 5,02007 | 94,97993% |

Berdasarkan Tabel 4.19 diketahui terjadi peningkatan akurasi pada metode yang digunakan. JST-BP tanpa menggunakan optimalisasi PSO menghasilkan akurasi sebesar 93,71677%, dengan nilai MSE dan MAPE *training* sebesar 2,25938 dan 3,03976, kemudian nilai MSE dan MAPE *testing* sebesar 13,86345 dan 6,28323. Metode JST-BP juga sebelumnya sudah pernah digunakan pada penelitian lain, salah satunya oleh Wanto & Windarto (2017) untuk memprediksi data *time series* yaitu indeks harga konsumen dengan hasil arsitektur terbaik 12-70-1 memperoleh akurasi 92% dan MSE 0,3659742.

Sedangkan pada saat melakukan prediksi dengan menggunakan metode JST-BP dan optimalisasi PSO menghasilkan akurasi sebesar 94,97993%, dimana nilai MSE dan MAPE *training* 1,96737 dan 1,85356, kemudian nilai MSE dan MAPE *testing* 7,15827 dan 5,02007. Sehingga dapat disimpulkan bahwa terjadi peningkatan akurasi sebesar 1,26316%.

Hasil peningkatan akurasi tersebut tidak lepas dari pemilihan nilai parameter yang telah dilakukan melalui tahap pengujian parameter. Parameter yang diperoleh dari pengujian yang telah dilakukan adalah arsitektur model JST-BP dan nilai parameter PSO. Arsitektur model JST-BP yang diperoleh berupa 5 *input layer*, 3 *hidden layer*, 1 *output layer*,nilai *epoch* 60, dan nilai *learning rate* 0,2. Sedangkan nilai parameter PSO yang diperoleh berupa 15 partikel, 5 *popsize*, nilai *epoch* 16, nilai c1 adalah 1, nilai c2 adalah 1,5 dan bobot inersia adalah 0,5.

Meskipun algoritma PSO mampu meningkatkan akurasi dan meminimalkan nilai *error* pada metode JST-BP, tetapi pada saat proses *training* memakan waktu yang cukup lama. Hal itu dikarenakan setiap *epoch* yang dimiliki JST-BP melakukan perhitungan *update* bobot di tiap *epoch* PSO. Sehingga seiring besarnya nilai *epoch* JST-BP dan PSO, maka akan semakin lama pula proses *update* bobot yang dilakukan.

Terlepas dari kelemahan waktu yang diperlukan lebih lama untuk proses *update* bobot pada JST-BP dan PSO, model ini diharapkan mampu memprediksi kasus peramalan harga dengan lebih akurat pada berbagai data *time series* harian.

Peramalan harga minyak mentah dunia telah dilakukan dalam beberapa penelitian dengan berbagai macam metode dan *dataset* yang diambil, beberapa diantaranya yaitu penelitian dengan menggunakan *dataset* harga minyak mentah WTI berdasarkan urutan waktu harian dilakukan oleh Faozi & Sulistijanti (2016) menggunakan metode ARIMA berhasil mendapatkan nilai MSE 0,972, akan tetapi data yang digunakan hanya sebanyak 48 data, lalu penelitian lain yang dilakukan oleh Wulandari *et al*., (2017) dengan menggunakan metode RBFNN memperoleh nilai MAPE sebesar 6,8150, tetapi data harga minyak yang digunakan berdasarkan urutan waktu bulanan. Penelitian yang lainnya juga dilakukan oleh Setiyowati *et al.*, (2018) dengan menggunakan metode ARIMA dan mendapatkan nilai RMSE 8,345%. Dari beberapa penelitian yang telah dilakukan sebelumnya, dapat dikatakan bahwa model JST-BP – PSO yang digunakan pada penelitian ini memperoleh hasil peramalan yang baik dengan tingkat akurasi yang cukup tinggi.

Untuk penelitian selanjutnya, diharapkan dapat mengembangkan model JST-BP dan *Swarm Intelligence* lainnya menggunakan *dataset* *Crude Oil* standar WTI (CL=F), karena dibandingkan dengan penelitian yang pernah diusulkan sebelumnya bahwa model JST-BP sudah mampu melakukan peramalan dengan akurasi yang lebih baik.

# 

# BAB 5 PENUTUP

1. 1. **Kesimpulan**

Berdasarkan hasil penelitian dan pembahasan, terkait optimasi JST-BP menggunakan algoritma PSO untuk meningkatkan akurasi pada prediksi harga minyak mentah WTI maka dapat ditarik kesimpulan sebagai berikut:

1. Penerapan algoritma PSO dalam mengoptimasi parameter bobot (*w*) JST-BP membuat kualitas prediksi harga minyak mentah meningkat, terbukti dari hasil MSE, MAPE dan akurasi JST-BP – PSO lebih baik dibandingkan dengan hanya menggunakan JST-BP.
2. Hasil nilai MSE, MAPE dan akurasi menggunakan algoritma PSO dalam meotde JST-BP sebesar 7,15827, 5,02007 dan 94,97993%. Nilai akurasi mengalami peningkatan 1,26316% dari pada hanya menggunakan model JST-BP saja yang memiliki nilai MSE, MAPE dan akurasi sebesar 13,86345, 6,28323 dan 93,71677% untuk objek *Crude Oil* standar WTI (CL=F) *dataset* yang berasal dari situs finance.yahoo.com.
   1. **Saran**

Adapun saran dari penelitian ini adalah sebagai berikut:

* + - 1. Penelitian lebih lanjut diharapkan dapat mengembangkan penelitian ini dengan menggunakan algoritma optimasi *Swarm Intelligence* lain untuk lebih meningkatkan akurasi dengan waktu komputasi yang lebih singkat pada prediksi harga minyak mentah seperti ACO (*Ant Colony Optimization*), BCO (*Bee Colony Optimization*), CSO\_a (*Cat Swarm Optimization*), CSO\_b (*Cockroach Swarm Optimization*).
      2. Melakukan pengembangan dari penelitian ini melalui pengujian nilai parameter JST-BP menggunakan teknik Level DOE.

# DAFTAR PUSTAKA

Alkronz, Eyad Sameh., Khaled A. Moghayer., Mohamad Meimeh., et al Mohannad Gazzaz, Bassem S. Abu-Nasser, S. S. A.-N. (2019). Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network . *International Journal of Academic and Applied Research*, *3*(2), 1–8. http://www.ijeais.org/ijaar

Almas, M. F., & Setiawan, B. D. (2018). Implementasi Metode Backpropagation untuk Prediksi Harga Batu Bara. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(12), 6502–6511.

Aulia, R. (2018). Penerapan Metode Backpropagation Untuk Memprediksi Jumlah Kunjungan Wisatawan Berdasarkan Tingkat Hunian Hotel. *Jurteksi*, *4*(2), 115–122. https://doi.org/10.33330/jurteksi.v4i2.45

Bildirici, M., & Ersin, Ö. (2015). Forecasting volatility in oil prices with a class of nonlinear volatility models: smooth transition RBF and MLP neural networks augmented GARCH approach. *Petroleum Science*, *12*(3), 534–552. https://doi.org/10.1007/s12182-015-0035-8

Chen, J. (2020). *Crude Oil*. Investopedia.Com. https://www.investopedia.com/terms/c/crude-oil.asp

Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, *74*(November), 902–924. https://doi.org/10.1016/j.rser.2017.02.085

Effendy, F., & Nuqoba, B. (2016). Penerapan Framework Bootsrap Dalam Pembangunan Sistem Informasi Pengangkatan Dan Penjadwalan Pegawai (Studi Kasus:Rumah Sakit Bersalin Buah Delima Sidoarjo). *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *11*(1), 9. https://doi.org/10.30872/jim.v11i1.197

Faozi, S., & Sulistijanti, W. (2016). Peramalan Harga Minyak Mentah Standar West Texas Intermediate dengan Pendekatan Metode ARIMA. *Seminar Nasional Pendidikan, Sains Dan Teknologi*, 308–316.

Fazelabdolabadi, B. (2019). A hybrid Bayesian-network proposition for forecasting the crude oil price. *Financial Innovation*, *5*(1). https://doi.org/10.1186/s40854-019-0144-2

Giawa, D. I. (2017). Aplikasi Jaringan Syaraf Tiruan untuk Memprediksi Penjualan Handphone dengan Menggunakan Metode Backpropagation (Studi Kasus : CV.Bryan Ponsel). *Informasi Dan Teknologi Ilmiah (INTI)*, *12*, 162–167.

Hartati. (2017). Penggunaan Metode Arima Dalam Meramal Pergerakan Inflasi. *Jurnal Matematika Sains Dan Teknologi*, *18*(1), 1–10. https://doi.org/10.33830/jmst.v18i1.163.2017

Ika, R. C., & Setiawan, B. D. (2018). Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization untuk Peramalan Nilai Pembayaran Penjaminan Kredit Macet. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(6), 2364–2373.

Irsyad, R. (2018). Penggunaan Python Web Framework Flask Untuk Pemula. *Institut Teknologi Bandung*. https://doi.org/10.31219/osf.io/t7u5r

Julpan, Nababan, E. B., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma Backpropagation Pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, *02*(1), 103–116.

Li, X., Shang, W., & Wang, S. (2019). Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting*, *35*(4), 1548–1560. https://doi.org/10.1016/j.ijforecast.2018.07.006

Lubis, M. R. (2017). Metode Hybrid Particle Swarm Optimization - Neural Network Backpropagation Untuk Prediksi Hasil Pertandingan Sepak Bola. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *1*(1), 71. https://doi.org/10.30645/j-sakti.v1i1.30

Maori, N. A. (2017). Perbandingan Metode ANN-PSO dan ANN-GA untuk Peningkatan Akurasi Prediksi Harga Emas ANTAM. *Disprotek*, *8*(1), 67–80.

Mukhlis, I. F. (2016). Optimasi PSO Untuk Peramalan Harga Emas Secara Rentet Waktu. *GEMA AKTUALITA*, *5*(1), 73–77.

Navares, R., Díaz, J., Linares, C., & Aznarte, J. L. (2018). Comparing ARIMA and computational intelligence methods to forecast daily hospital admissions due to circulatory and respiratory causes in Madrid. *Stochastic Environmental Research and Risk Assessment*, *32*(10), 2849–2859. https://doi.org/10.1007/s00477-018-1519-z

Perdana, I. A. (2016). Jaringan Syaraf Tiruan Backpropagation ( Study Kasus : PDAM TIRTA KEPRI ). *Universitas Maritim Raja Ali Haji*, 1–12.

Putra, H. E. (2016). *Pengaruh Harga Minyak Dunia , Nilai Tukar Netto Terhadap Pertumbuhan Ekonomi Di Indonesia Tahun 1985-2014*. Universitas Airlangga.

Rahayu, D., Wihandika, R. C., & Perdana, R. S. (2018). Implementasi Metode Backpropagation Untuk Klasifikasi Kenaikan Harga Implementasi Metode Backpropagation Untuk Klasifikasi Kenaikan Harga Minyak Kelapa Sawit. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(April), 1547–1552.

Ramyar, S., & Kianfar, F. (2017). Forecasting Crude Oil Prices : A Comparison Between Artificial Neural Networks and Vector Autoregressive Models. *Computational Economics*. https://doi.org/10.1007/s10614-017-9764-7

Revi, A., Solikhun, S., & Safii, M. (2018). Jaringan Syaraf Tiruan Dalam Memprediksi Jumlah Produksi Daging Sapi Berdasarkan Provinsi. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *2*(1), 297–304. https://doi.org/10.30865/komik.v2i1.941

Rusmalawati, V., & Furqon, M. T. (2018). Peramalan Harga Saham Menggunakan Metode Support Vector Regression ( SVR ) Dengan Particle Swarm Optimization ( PSO ). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(5), 1980–1990.

Salsabila, B., & Cholissodin, I. (2020). Prediksi Permintaan Keripik Buah dengan Metode Jaringan Syaraf Tiruan Backpropagation ( Studi Kasus : CV . Arjuna 999 ). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *4*(6), 1667–1674.

Sari, A. P. (2019). *Optimasi Interval Fuzzy Time Series Menggunakan Particle Swarm Optimization Untuk Memprediksi Kualitas Udara Di Kota Pekanbaru*. Universitas Islam Negeri Sultan Syarif Kasim Riau.

Setiyowati, E., Rusgiyono, A., & Tarno. (2018). Model kombinasi arima dalam peramalan harga minyak mentah dunia. *Jurnal Gaussian*, *7*(1), 54–63.

Siregar, E. (2019). *Analisis Metode Backpropagation dengan Fungsi Aktivasi Sigmoid Bipolar dan Fungsi Linear dalam Prediksi Pertumbuhan Penduduk*. Universitas sumatera Utara.

Srinath, K. R. (2017). Python – The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*, *04*(12), 354–357. https://d1wqtxts1xzle7.cloudfront.net/55458585/IRJET-V4I1266.pdf?1515226715=&response-content-disposition=inline%3B+filename%3DPython\_The\_Fastest\_Growing\_Programming\_L.pdf&Expires=1593202307&Signature=HBD7oa85wDxqRzTWX01uVRBlMacGX5mkGk1b~SVVTTkENJ6cf5diKz

Sudarsono, A. (2016). Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation (Studi Kasus di Kota Bengkulu). *Media Infotama*, *12*(1), 61–69.

VanderPlas, J. (2016). Python Data Science Handbook. In D. Schanafelt (Ed.), *O’Reilly*. O’Reilly Media.

Vinet, L., & Zhedanov, A. (2011). A “missing” family of classical orthogonal polynomials. *Journal of Physics A: Mathematical and Theoretical*, *44*(8), 1689–1699. https://doi.org/10.1088/1751-8113/44/8/085201

Wanto, A., & Windarto, A. P. (2017). Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode Backpropagation. *Jurnal & Penelitian Teknik Informatika Sinkron*, *2*(2), 37–43. https://zenodo.org/record/1009223#.Wd7norlTbhQ

Wulandari, A., Wahyuningsih, S., & Amijaya, dan F. D. T. (2017). Peramalan Harga Minyak Mentah Dunia ( Crude Oil ) Menggunakan Metode Radial Basis Function Neural Network ( RBFNN ) Crude Oil Price Forecasting Using Radial Basis Function Neural Network Method ( RBFNN ). *Jurnal EKSPONENSIAL*, *8*(2), 161–168.

Xu, X., Rong, H., Trovati, M., Bessis, N., & Liptrott, M. (2018). CS-PSO : chaotic particle swarm optimization algorithm for solving combinatorial optimization problems. *Soft Computing*, *22*(3), 783–795. https://doi.org/10.1007/s00500-016-2383-8

Zhang, K., Li, G., & Zhang, M. (2017). Assembly Data Mining Platform based on Python. *Proceedings of Science*, *2017*-*Decem*(51775552), 1–6. https://doi.org/10.22323/1.300.0056

Zhu, B., Feng, Y., Gong, D., Jiang, S., Zhao, L., & Cui, N. (2020). Hybrid particle swarm optimization with extreme learning machine for daily reference evapotranspiration prediction from limited climatic data. *Computers and Electronics in Agriculture*, *173*(October 2019), 105430. https://doi.org/10.1016/j.compag.2020.105430

# 

# LAMPIRAN

Lampiran . Kode Program Sistem

*Source Code* app.py

|  |
| --- |
| import pandas as pd  import json  import os as os  from werkzeug.utils import secure\_filename  from BPxPSO import PSOxBackpro, Backpropagation, testing  from matplotlib import pyplot as plt  from flask import Flask, render\_template, url\_for, send\_from\_directory, request, redirect, jsonify  import matplotlib  matplotlib.use("Agg")  app = Flask(\_\_name\_\_)  def bagi\_data(file\_path):      dataset = pd.read\_csv(file\_path)      len\_train = int((len(dataset)-5)\*0.7)        date = dataset['Date']      date\_test = date.iloc[len\_train+5:]        dataset\_value\_train = dataset.iloc[:len\_train, 1].values      dataset\_value\_test = dataset.iloc[len\_train:, 1].values      max\_val = max(dataset\_value\_train)      min\_val = min(dataset\_value\_train)      max\_val\_test = max(dataset\_value\_test)      min\_val\_test = min(dataset\_value\_test)      normalized\_dataset\_train = list()      for x in range(len(dataset\_value\_train)):          norm\_value = (dataset\_value\_train[x] - min(dataset\_value\_train)) / (              max(dataset\_value\_train) - min(dataset\_value\_train))          normalized\_dataset\_train.append(norm\_value)      normalized\_dataset\_test = list()      for x in range(len(dataset\_value\_test)):          norm\_value = (dataset\_value\_test[x] - min(dataset\_value\_test)) / (              max(dataset\_value\_test) - min(dataset\_value\_test))          normalized\_dataset\_test.append(norm\_value)        time\_series\_dataset\_train = [          normalized\_dataset\_train[:len(normalized\_dataset\_train) - 5],          normalized\_dataset\_train[1:len(normalized\_dataset\_train) - 4],          normalized\_dataset\_train[2:len(normalized\_dataset\_train) - 3],          normalized\_dataset\_train[3:len(normalized\_dataset\_train) - 2],          normalized\_dataset\_train[4:len(normalized\_dataset\_train) - 1],          normalized\_dataset\_train[5:]]      time\_series\_dataset\_test = [          normalized\_dataset\_test[:len(normalized\_dataset\_test) - 5],          normalized\_dataset\_test[1:len(normalized\_dataset\_test) - 4],          normalized\_dataset\_test[2:len(normalized\_dataset\_test) - 3],          normalized\_dataset\_test[3:len(normalized\_dataset\_test) - 2],          normalized\_dataset\_test[4:len(normalized\_dataset\_test) - 1],          normalized\_dataset\_test[5:]]      df\_train = pd.DataFrame([list(i) for i in zip(\*time\_series\_dataset\_train)])      df\_test = pd.DataFrame([list(i) for i in zip(\*time\_series\_dataset\_test)])      X\_train = df\_train.iloc[:, :5]      Y\_train = df\_train.iloc[:, 5]      X\_test = df\_test.iloc[:, :5]      Y\_test = df\_test.iloc[:, 5]      return X\_train, Y\_train, X\_test, Y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, date\_test    def un\_normalized(value, min, max):      return value \* (max - min) + min  def plot\_mse(x, list, nama):      plt.plot(x, list)      plt.ylabel('Nilai MSE')      plt.xlabel('Epoch')      plt.savefig('static/images/' + nama + '.png', bbox\_inches='tight')      plt.close()  def plot\_mape(x, list, nama):      plt.plot(x, list)      plt.ylabel('Nilai MAPE')      plt.xlabel('Epoch')      plt.savefig('static/images/' + nama + '.png', bbox\_inches='tight')      plt.close()    def plot\_prediksi(asli, hasil, nama):      asli.plot(figsize=(16,8))      plt.plot(hasil)      plt.ylabel('Price (in USD)')      plt.savefig('static/images/' + nama + '.png', bbox\_inches='tight')      plt.close()  @app.route("/")  @app.route('/index')  def main():      return render\_template('index.html')  @app.route('/dataset')  def data():      data = pd.read\_csv('dataset/wti.csv')      data['Date'] = pd.to\_datetime(data['Date'])      data['Date'] = data['Date'].dt.strftime("%d-%m-%Y")      length = len(data)        data\_value = data.iloc[:, 1].values      max\_value = max(data\_value)      min\_value = min(data\_value)      normalized\_data = list()      for x in range(len(data\_value)):          norm\_value = (data\_value[x] - min\_value) / (max\_value - min\_value)          normalized\_data.append(norm\_value)      data['Normalized'] = pd.DataFrame(normalized\_data, columns=['Normalized'])      data = data.round(5)        return render\_template('dataset.html', data=data, length=length)  @app.route('/proses-bp')  def prosesbp():      return render\_template('proses-bp.html')    @app.route('/training-bp', methods=['POST'])  def trainingbp():      datasets = request.files['dataset\_source']      basepath = os.path.dirname(\_\_file\_\_)      file\_path = os.path.join(          basepath, 'dataset', secure\_filename(datasets.filename))        learning\_rate = float(request.form['learning\_rate'])      epoch = int(request.form['epoch'])      input\_layer = int(request.form['input\_layer'])      hidden\_layer = int(request.form['hidden\_layer'])      output\_layer = int(request.form['output\_layer'])      particle\_size = (input\_layer\*hidden\_layer)      weight = [0.3522731340418094, 0.14640481330378685, 0.40211953183863935, 0.8250205036766922, 0.8731580679027796, 0.9375709821301595, 0.46340809666941096, 0.46181264237637687, 0.0696680142964593, 0.7476107133759758, 0.7257548359958343, 0.057108187494075535, 0.3548478004722213, 0.732463316869996, 0.687735063846984]      X\_train, Y\_train, X\_test, Y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, date\_test = bagi\_data(file\_path)        print('Training data...')        backpro = Backpropagation(input\_layer, hidden\_layer, output\_layer, learning\_rate)      backpro.initWeight(weight)      mape\_train, hasil, mse\_train, mselist, mapelist = backpro.training(X\_train, Y\_train, max\_val, min\_val, epoch)        x = [x+1 for x in range(epoch)]        plot\_mse(x, mselist, 'mse\_bp')      plot\_mape(x, mapelist, 'mape\_bp')        asli\_test = un\_normalized(Y\_test, min\_val\_test, max\_val\_test)      mape\_test, mse\_test, hasil\_test, akurasi = backpro.testing(X\_test, Y\_test, max\_val\_test, min\_val\_test)      asli = pd.DataFrame(date\_test, columns=['Date'])      asli.reset\_index(drop=True, inplace=True)      asli['Harga'] = asli\_test      asli = pd.Series(asli['Harga'].values, index=asli['Date'])        plot\_prediksi(asli, hasil\_test, 'prediksi\_bp')        mse\_train = round(mse\_train, 5)      mape\_train = round(mape\_train, 5)      mse\_test = round(mse\_test, 5)      mape\_test = round(mape\_test, 5)      akurasi = round(akurasi, 5)        result = {'mse': mape\_train, 'mape': mse\_train, 'mse\_test': mse\_test, 'mape\_test': mape\_test, 'akurasi': akurasi}      return jsonify(result)  @app.route('/hasil-bp')  def hasilbp():      return render\_template('hasil-bp.html')  @app.route('/proses-pso')  def prosespso():      return render\_template('proses-pso.html')  @app.route('/training-pso', methods=['POST'])  def trainingpso():      datasets = request.files['dataset\_source']      basepath = os.path.dirname(\_\_file\_\_)      file\_path = os.path.join(          basepath, 'dataset', secure\_filename(datasets.filename))        learning\_rate = float(request.form['learning\_rate'])      epoch\_bp = int(request.form['epoch'])      input\_layer = int(request.form['input\_layer'])      hidden\_layer = int(request.form['hidden\_layer'])      output\_layer = int(request.form['output\_layer'])      pop\_size = int(request.form['pop\_size'])      epoch\_pso = int(request.form['epoch\_pso'])      w\_inersia = float(request.form['w\_inersia'])      c1 = float(request.form['c1'])      c2 = float(request.form['c2'])      particle\_size = (input\_layer\*hidden\_layer)        X\_train, Y\_train, X\_test, Y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, date\_test = bagi\_data(file\_path)        coba = PSOxBackpro(pop\_size, particle\_size)      coba.set\_backpro\_param(X\_train, Y\_train, X\_test, Y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, epoch\_bp, input\_layer, hidden\_layer, output\_layer, learning\_rate)      coba.initPops()      print('Optimizing...')      \_, fitness, \_, listfit, listmse, listmape = coba.optimize(epoch\_pso, w\_inersia, c1, c2)      mape\_train = round(coba.g\_best.mape, 5)      mse\_train = round(coba.g\_best.mse, 5)      fitness = round(fitness, 5)        x = [x+1 for x in range(epoch\_pso)]        mselist = listmse      mapelist = listmape        plot\_mse(x, mselist, 'mse\_pso')      plot\_mape(x, mapelist, 'mape\_pso')        asli\_test = un\_normalized(Y\_test, min\_val\_test, max\_val\_test)        testing = PSOxBP.pops[0]        mape\_test = round(testing.mape\_test, 5)      mse\_test = round(testing.mse\_test, 5)      akurasi = round(testing.akurasi, 5)      hasil\_test = testing.hasil        asli = pd.DataFrame(date\_test, columns=['Date'])      asli.reset\_index(drop=True, inplace=True)      asli['Harga'] = asli\_test      asli = pd.Series(asli['Harga'].values, index=asli['Date'])        plot\_prediksi(asli, hasil\_test, 'prediksi\_pso')        result = {'mse': mape\_train, 'mape': mse\_train, 'fitness': fitness, 'mse\_test': mse\_test, 'mape\_test': mape\_test, 'akurasi': akurasi}      return jsonify(result)    @app.route('/hasil-pso')  def hasilpso():      return render\_template('hasil-pso.html')    @app.route('/tentang')  def tentang():      return render\_template('tentang.html')  @app.route('/<path:resource>')  def serveStaticResource(resource):      return send\_from\_directory('static/', resource)    if \_\_name\_\_ == "\_\_main\_\_" :      app.run(debug=True) |

Lampiran . Kode Program User Interface

*Source Code* dataset.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %} Dataset {% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <div class="container">      <h2>Daftar Dataset</h2>      <hr>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#home">Dataset Asli</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#menu1">Hasil Normalisasi</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="home" class="container tab-pane active"><br>              <h3>Harga Minyak Mentah (WTI)</h3>              <p>Data yang digunakan dalam penelitian ini menggunakan dataset World Daily Prices for Crude Oil dari situs                  finance.yahoo.com. Dataset harga minyak mentah dunia ini diukur dalam satuan dollar Amerika (USD). </p>              <div class="row justify-content-center">                  <div class="col-auto">                      <table style="max-height: 400px; overflow: auto; display:inline-block; max-width: 600px;" class="table table-striped table-bordered table-hover" id="dataTables-dataset">                          <thead>                              <tr>                                  <th class="text-center" style="width:300px;">Tanggal</th>                                  <th class="text-center" style="width:300px">Harga (dalam USD)</th>                              </tr>                          </thead>                          <tbody>                              {% for i in range(length): %}                              <tr>                                  <td class="text-center">{{ data.iloc[i,0] }}</td>                                  <td class="text-center">{{ data.iloc[i,1] }}</td>                              </tr>                              {% endfor %}                          </tbody>                      </table>                  </div>              </div>          </div>          <div id="menu1" class="container tab-pane fade"><br>              <h3>Normalisasi Data</h3>              <p>Dalam penelitian ini data akan ditransformasi atau normalisasi dalam interval [0,1].</p>              <div class="row justify-content-center">                  <div class="col-auto">                      <table style="max-height: 400px; overflow: auto; display:inline-block; max-width: 600px;" class="table table-striped table-bordered table-hover" id="dataTables-dataset">                          <thead>                              <tr>                                  <th class="text-center" style="width:300px">Hasil Normalisasi</th>                              </tr>                          </thead>                          <tbody>                              {% for i in range(length): %}                              <tr>                                  <td class="text-center">{{ data.iloc[i,2] }}</td>                              </tr>                              {% endfor %}                          </tbody>                      </table>                  </div>              </div>          </div>      </div>  </div>  {% endblock %} |

*Source Code* index.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Beranda{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <div id="content" class="p-4 p-md-5 pt-5">      <div class="mb-4">          <h3 class="text-center"><strong>OPTIMALISASI JARINGAN SYARAF TIRUAN BACKPROPAGATION (JST-BP) DENGAN METODE                  PARTICLE                  SWARM OPTIMIZATION                  (PSO) UNTUK MEMPREDIKSI HARGA MINYAK MENTAH</strong</h3> <br> <br>                  <img src={{url\_for('static',filename='images/unnes.png')}} height="210" width="140"> <br> <br>                  <h4 class="text-center">Fitri Amalia Langgundi                  </h4>                  <h4 class="text-center">4611417009</h4><br><br>                  <h4 class="text-center">                      <strong>                          Program Studi Teknik Informatika<br>                          Jurusan Ilmu Komputer<br>                          Fakultas Matematika dan Ilmu Pengetahuan Alam<br>                          Universitas Negeri Semarang<br>                          2021                      </strong>                  </h4>      </div>  </div>  {% endblock %} |

*Source Code* layout.html

|  |
| --- |
| <!doctype html>  <html lang="en">      <head>          {% block head %} {% endblock %}          <title>{% block title %} {% endblock %}</title>          <meta charset="utf-8">          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">          <link rel="stylesheet" href="{{ url\_for('serveStaticResource', resource='fonts/fonts.css') }}">          <link rel="stylesheet" href="{{ url\_for('serveStaticResource', resource='css/font-awesome.min.css') }}">          <link rel="stylesheet" href="{{ url\_for('serveStaticResource', resource='css/style.css') }}">          <script src="{{ url\_for('serveStaticResource', resource='js/chart.min.js') }}"></script>          <script type="text/javascript" src="{{ url\_for('serveStaticResource', resource='js/canvasjs.stock.min.js') }}"></script>          <script type="text/javascript" src="{{ url\_for('serveStaticResource', resource='js/jquery-1.11.1.min.js') }}"></script>          {% block css %} {% endblock %}      </head>      <body>          <div class="wrapper d-flex align-items-stretch">              <nav id="sidebar">                  <div class="custom-menu">                      <button type="button" id="sidebarCollapse" class="btn btn-primary">                          <i class="fa fa-bars"></i>                          <span class="sr-only">Toggle Menu</span>                      </button>                  </div>                  <h1><a href="index" class="logo">Prediksi Harga Minyak Mentah</a></h1>                  <ul class="list-unstyled components mb-5">                      <li>                          <a href="index"><span class="fa fa-home mr-3"></span> Beranda</a>                      </li>                      <li>                          <a href="dataset"><span class="fa fa-sticky-note mr-3"></span> Dataset</a>                      </li>                      <li>                          <a href="#menubp" data-toggle="collapse" aria-expanded="false" class="dropdown-toggle"><span class="fa fa-gear mr-3" ></span> JST-BP</a>                          <ul class="collapse list-unstyled" id="menubp">                              <li>                                  <a href="proses-bp">Proses JST-BP</a>                              </li>                              <li>                                  <a href="hasil-bp">Hasil JST-BP</a>                              </li>                          </ul>                      </li>                      <li>                          <a href="#menupso" data-toggle="collapse" aria-expanded="false" class="dropdown-toggle"><span class="fa fa-gear mr-3"></span> JST-BP - PSO</a>                          <ul class="collapse list-unstyled" id="menupso">                              <li>                                  <a href="proses-pso">Proses JST-BP - PSO</a>                              </li>                              <li>                                  <a href="hasil-pso">Hasil JST-BP - PSO</a>                              </li>                          </ul>                      </li>                      <li>                          <a href="tentang"><span class="fa fa-info-circle mr-3"></span> Tentang</a>                      </li>                  </ul>              </nav>              <!-- Page Content  -->              <div id="content" class="p-4 p-md-5 pt-5">                  <h2 class="mb-4">{% block judul %} {% endblock %}</h2>                  {% block content %} {% endblock %}              </div>          <script src="{{ url\_for('serveStaticResource', resource='js/jquery.min.js') }}"></script>          <script src="{{ url\_for('serveStaticResource', resource='js/jquery-3.6.0.min.js') }}"></script>          <script src="{{ url\_for('serveStaticResource', resource='js/popper.js') }}"></script>          <script src="{{ url\_for('serveStaticResource', resource='js/bootstrap.min.js') }}"></script>          <script src="{{ url\_for('serveStaticResource', resource='js/main.js') }}"></script>          <script type="text/javascript">              $(function($) {                  let url = window.location.href;                  $('nav ul li a').each(function() {                      if (this.href === url) {                          $(this).closest('li').addClass('active');                          if (url === 'http://127.0.0.1:5000/hasil-bp' || url === 'http://127.0.0.1:5000/proses-bp') {                              $('#menubp').addClass('show');                          }                          else if (url === 'http://127.0.0.1:5000/hasil-pso' || url === 'http://127.0.0.1:5000/proses-pso') {                              $('#menupso').addClass('show');                          }                      }                  });              });          </script>          {% block js %} {% endblock %}          </div>      </body>  </html> |

*Source Code* tentang.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Tentang{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <div class="container">      <h2>Tentang</h2>      <hr>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#home">Tentang Sistem</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#menu1">Tentang Dataset</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="home" class="container tab-pane active"><br>              <h3>Tentang Prediksi Harga Minyak Mentah</h3>              <p>Sistem ini digunakan untuk mempredisksi harga minyak mentah dunia atau crude oil price dalam penskalaan                  WTI (West Texas Intermediate) yang diukur dalam satuan dollar Amerika (USD). Sistem ini dibuat untuk                  memenuhi syarat skripsi dari Fitri Amalia Langgundi dalam memperoleh gelar Sarjana Komputer di                  Universitas Negeri Semarang. Judul skripsi yang dipakai yaitu "Optimalisasi Jaringan Syaraf Tiruan                  Backpropagation (JST-BP) dengan Metode Particle Swarm Optimization (PSO) untuk Memprediksi Harga Minyak                  Mentah". </p> <br>              <p>Tahap penelitian yang digunaan sebagai beriut:</p><br>              <pre>                  1. Pengambilan Data                  2. Normalisasi Data                  3. Pengujian Parameter                  4. Penentuan Model                  5. Prediksi                  6. Perancangan Sistem                  5. Penarikan Kesimpulan                  </pre>          </div>          <div id="menu1" class="container tab-pane fade"><br>              <h3>Tentang Dataset</h3>              <p class="text-justify">Jenis data yang digunakan yaitu data kuantitatif berupa numerik harga minyak mentah dunia dalam satuam                  U.S.Dollars. Dataset yang digunakan pada penelitian ini adalah dataset publik yaitu World Daily Prices                  for Crude Oil dataset yang diperoleh dari situs finance.yahoo.com dan secara harian terupdate otomatis.                  Data ini telah dicatat oleh yahoo finance secara harian terhitung sejak 23 Agustus hingga sekarang.                  Sebanyak 911 data digunakan dalam penelitian terhitung sejak 3 Januari 2017 hingga 31 Maret 2021 yang                  diakses pada tanggal . Data harian harga minyak mentah dunia sesuai dengan trading days yaitu hari Senin                  – Jum’at dengan jumlah 5 hari kerja dalam satu minggu (tidak termasuk hari libur). </p><br>          </div>          <div id="menu2" class="container tab-pane fade"><br>              <h3>PSO</h3>              <p>PSO digunakan untuk optimasi jaringan JST-BP melalui proses training sehingga didapatkan model jaringan terbaik.              </p>          </div>          <div id="menu2" class="container tab-pane fade"><br>              <h3>JST Backpropagation</h3>              <p>JST Backpropagation digunakan sebagai metode prediksi harga minyak mentah.</p>          </div>      </div>  </div>  {% endblock %} |

*Source Code* proses-bp.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Proses Backpropagation{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <br>  <div class="container">      <h2>Proses JST-Backpropagation</h2>      <br>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#proses">Proses Perhitungan</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#hasil">Hasil</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="proses" class="container tab-pane active"><br>              <form method="POST" enctype="multipart/form-data" id="train\_data">                  <div class="form-group mt-3">                      <label for="dataset\_source">Dataset</label>                      <input type="file" name="dataset\_source" id="dataset\_source" required class="form-control" placeholder="Sumber dataset" required>                  </div>                  <div class="row mt-3">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="input\_layer">Input Layer</label>                              <input value="5" type="number" name="input\_layer" id="input\_layer" required placeholder="Input Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="hidden\_layer">Hidden Layer</label>                              <input value="3" type="number" name="hidden\_layer" id="hidden\_layer" required placeholder="Hidden Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="output\_layer">Output Layer</label>                              <input value="1" type="number" name="output\_layer" id="output\_layer" required placeholder="Output Layer" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="row">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="epoch">Epoch</label>                              <input value="60" type="number" name="epoch" id="epoch" required placeholder="Epoch Pelatihan" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="learning\_rate">Learning Rate</label>                              <input value="0.2" type="number" name="learning\_rate" id="learning\_rate" required placeholder="Learning Rate" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="form-group mt-3">                      <button class="btn btn-primary px-4" type="button" id="proses\_btn">Mulai Proses</button>                  </div>              </form>              <div class="row mb-5" id="result\_train" style="display: none;">                  <div class="col-md-12" id="result\_loading" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body text-center">                              <img src="\static\images\load2.gif" alt="Loading" height="100">                          </div>                      </div>                  </div>                  <div class="col-md-12" id="result\_field" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body">                              <h5 align="center">Proses perhitungan sudah selesai, silahkan cek tab Hasil</h5>                          </div>                      </div>                  </div>              </div>          </div>            <div id="hasil" class="container tab-pane fade"><br>              <h4>Hasil Training</h4>              <hr>              <div class="row">                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MSE</p>                      <img width="100%" id="img\_mse"></img>                  </div>                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MAPE</p>                      <img width="100%" id="img\_mape"></img>                  </div>              </div>              <br><br>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_train"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_train"></span></p>              <hr>              <br>              <h4>Hasil Testing</h4>              <hr>              <p align="center" style="font-weight: bold; font-size:130%;">Grafik Perbandingan Harga Prediksi</p>              <img width="100%" id="img\_prediksi"></img>              <br><br>              <p style="font-size:130%;">Tingkat Akurasi: <span id="akurasi"></span>%</p>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_test"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_test"></span></p>          </div>      </div>  </div>  <script>      $('#proses\_btn').bind("click",function()      {          var imgVal = $('#dataset\_source').val();          if(imgVal=='')          {              alert("Masukkan file dataset!");              return false;          }            else{              $('#result\_train').show();              $('#result\_loading').show();              var data = new FormData();              data.append('input\_layer', $('#input\_layer').val());              data.append('hidden\_layer', $('#hidden\_layer').val());              data.append('output\_layer', $('#output\_layer').val());              data.append('epoch', $('#epoch').val());              data.append('learning\_rate', $('#learning\_rate').val());              data.append('dataset\_source', $('#dataset\_source')[0].files[0]);              $.ajax({                  type: 'POST',                  url: '/training-bp',                  data: data,                  contentType: false,                  cache: false,                  processData: false,                  async: true,                  success: function (data) {                      $('#mse\_train').html(data.mse);                      $('#mape\_train').html(data.mape);                      $('#akurasi').html(data.akurasi);                      $('#mse\_test').html(data.mse\_test);                      $('#mape\_test').html(data.mape\_test);                      $('#result\_loading').hide();                      $('#img\_mse').attr('src','{{url\_for('static', filename='images/mse\_bp.png')}}');                      $('#img\_mape').attr('src','{{url\_for('static', filename='images/mape\_bp.png')}}');                      $('#img\_prediksi').attr('src','{{url\_for('static', filename='images/prediksi\_bp.png')}}');                      $('#result\_field').show();                      console.log(data);                  },                  error: function (data) {                      console.log(data);                  }              });          }      });  </script>  {% endblock %} |

*Source Code* proses-pso.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Proses BP-PSO{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <br>  <div class="container">      <h2>Proses JST-BP - PSO</h2>      <br>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#proses">Proses Perhitungan</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#hasil">Hasil</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="proses" class="container tab-pane active"><br>              <form method="POST" enctype="multipart/form-data" id="train\_data">                  <div class="form-group mt-3">                      <label for="dataset\_source">Dataset</label>                      <input type="file" name="dataset\_source" id="dataset\_source" required class="form-control" placeholder="Sumber dataset">                  </div>                  <div class="row mt-3">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="input\_layer">Input Layer</label>                              <input value="5" type="number" name="input\_layer" id="input\_layer" required placeholder="Input Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="hidden\_layer">Hidden Layer</label>                              <input value="3" type="number" name="hidden\_layer" id="hidden\_layer" required placeholder="Hidden Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="output\_layer">Output Layer</label>                              <input value="1" type="number" name="output\_layer" id="output\_layer" required placeholder="Output Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="epoch">Epoch BP</label>                              <input value="60" type="number" name="epoch" id="epoch" required placeholder="Epoch Pelatihan" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="learning\_rate">Learning Rate</label>                              <input value="0.2" type="number" name="learning\_rate" id="learning\_rate" required placeholder="Learning Rate" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="row">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="pop\_size">Pop Size</label>                              <input value="5" type="number" name="pop\_size" id="pop\_size" required placeholder="Ukuran Populasi" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="epoch\_pso">Epoch PSO</label>                              <input value="16" type="number" name="epoch\_pso" id="epoch\_pso" required placeholder="Epoch PSO" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="w\_inersia">Bobot Inersia</label>                              <input value="0.5" type="number" name="w\_inersia" id="w\_inersia" required placeholder="Bobot Inersia" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="c1">Nilai c1</label>                              <input value="1.0" type="number" name="c1" id="c1" required placeholder="Nilai c1" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="c2">Nilai c2</label>                              <input value="1.5" type="number" name="c2" id="c2" required placeholder="Nilai c2" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="form-group mt-3">                      <button class="btn btn-primary px-4" type="button" id="proses\_btn">Mulai Proses</button>                  </div>              </form>              <div class="row mb-5" id="result\_train" style="display: none;">                  <div class="col-md-12" id="result\_loading" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body text-center">                              <img src="\static\images\load2.gif" alt="Loading" height="100">                          </div>                      </div>                  </div>                  <div class="col-md-12" id="result\_field" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body">                              <h5 align="center">Proses perhitungan sudah selesai, silahkan cek tab Hasil</h5>                          </div>                      </div>                  </div>              </div>          </div>            <div id="hasil" class="container tab-pane fade"><br>              <h4>Hasil Training</h4>              <hr>              <div class="row">                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MSE</p>                      <img width="100%" id="img\_mse"></img>                  </div>                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MAPE</p>                      <img width="100%" id="img\_mape"></img>                  </div>              </div>              <br>              <div class="row">                  <div class="col-sm-6 mx-auto">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik Fitness</p>                      <img width="50%"  width="100%" align="center" id="img\_fitness"></img>                  </div>              </div>              <br><br>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_train"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_train"></span></p>              <p style="font-size:130%;">Nilai Fitness: <span id="fitness"></span></p>              <br>              <h4>Hasil Testing</h4>              <hr>              <p align="center" style="font-weight: bold; font-size:130%;">Grafik Perbandingan Harga Prediksi</p>              <img width="100%" id="img\_prediksi"></img>              <br><br>              <p style="font-size:130%;">Tingkat Akurasi: <span id="akurasi"></span>%</p>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_test"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_test"></span></p>          </div>        </div>  </div>  <script>      $('#proses\_btn').bind("click",function()      {          var imgVal = $('#dataset\_source').val();          if(imgVal=='')          {              alert("Masukkan file dataset!");              return false;          }            else{              $('#result\_train').show();              $('#result\_loading').show();              var data = new FormData();              data.append('dataset\_source', $('#dataset\_source')[0].files[0]);              data.append('input\_layer', $('#input\_layer').val());              data.append('hidden\_layer', $('#hidden\_layer').val());              data.append('output\_layer', $('#output\_layer').val());              data.append('epoch', $('#epoch').val());              data.append('learning\_rate', $('#learning\_rate').val());              data.append('pop\_size', $('#pop\_size').val());              data.append('epoch\_pso', $('#epoch\_pso').val());              data.append('w\_inersia', $('#w\_inersia').val());              data.append('c1', $('#c1').val());              data.append('c2', $('#c2').val());              $.ajax({                  type: 'POST',                  url: '/training-pso',                  data: data,                  contentType: false,                  cache: false,                  processData: false,                  async: true,                  success: function (data) {                      $('#mse\_train').html(data.mse);                      $('#mape\_train').html(data.mape);                      $('#akurasi').html(data.akurasi);                      $('#mse\_test').html(data.mse\_test);                      $('#mape\_test').html(data.mape\_test);                      $('#fitness').html(data.fitness);                      $('#img\_mse').attr('src','{{url\_for('static',filename='images/mse\_pso.png')}}');                      $('#img\_mape').attr('src','{{url\_for('static',filename='images/mape\_pso.png')}}');                      $('#img\_fitness').attr('src','{{url\_for('static',filename='images/fitness.png')}}');                      $('#img\_prediksi').attr('src','{{url\_for('static',filename='images/prediksi\_pso.png')}}');                      $('#result\_loading').hide();                      $('#result\_field').show();                      console.log(data);                  },                  error: function (data) {                      console.log(data);                  }              });          }      });  </script>  {% endblock %} |

*Source Code* hasil-bp.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Proses BP-PSO{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <br>  <div class="container">      <h2>Proses JST-BP - PSO</h2>      <br>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#proses">Proses Perhitungan</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#hasil">Hasil</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="proses" class="container tab-pane active"><br>              <form method="POST" enctype="multipart/form-data" id="train\_data">                  <div class="form-group mt-3">                      <label for="dataset\_source">Dataset</label>                      <input type="file" name="dataset\_source" id="dataset\_source" required class="form-control" placeholder="Sumber dataset">                  </div>                  <div class="row mt-3">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="input\_layer">Input Layer</label>                              <input value="5" type="number" name="input\_layer" id="input\_layer" required placeholder="Input Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="hidden\_layer">Hidden Layer</label>                              <input value="3" type="number" name="hidden\_layer" id="hidden\_layer" required placeholder="Hidden Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="output\_layer">Output Layer</label>                              <input value="1" type="number" name="output\_layer" id="output\_layer" required placeholder="Output Layer" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="epoch">Epoch BP</label>                              <input value="60" type="number" name="epoch" id="epoch" required placeholder="Epoch Pelatihan" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="learning\_rate">Learning Rate</label>                              <input value="0.2" type="number" name="learning\_rate" id="learning\_rate" required placeholder="Learning Rate" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="row">                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="pop\_size">Pop Size</label>                              <input value="5" type="number" name="pop\_size" id="pop\_size" required placeholder="Ukuran Populasi" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="epoch\_pso">Epoch PSO</label>                              <input value="16" type="number" name="epoch\_pso" id="epoch\_pso" required placeholder="Epoch PSO" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="w\_inersia">Bobot Inersia</label>                              <input value="0.5" type="number" name="w\_inersia" id="w\_inersia" required placeholder="Bobot Inersia" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="c1">Nilai c1</label>                              <input value="1.0" type="number" name="c1" id="c1" required placeholder="Nilai c1" class="form-control" disabled>                          </div>                      </div>                      <div class="form-group col-lg-2">                          <div class="form-group">                              <label for="c2">Nilai c2</label>                              <input value="1.5" type="number" name="c2" id="c2" required placeholder="Nilai c2" class="form-control" disabled>                          </div>                      </div>                  </div>                  <div class="form-group mt-3">                      <button class="btn btn-primary px-4" type="button" id="proses\_btn">Mulai Proses</button>                  </div>              </form>              <div class="row mb-5" id="result\_train" style="display: none;">                  <div class="col-md-12" id="result\_loading" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body text-center">                              <img src="\static\images\load2.gif" alt="Loading" height="100">                          </div>                      </div>                  </div>                  <div class="col-md-12" id="result\_field" style="display: none;">                      <div class="card border-light shadow-sm">                          <div class="card-body">                              <h5 align="center">Proses perhitungan sudah selesai, silahkan cek tab Hasil</h5>                          </div>                      </div>                  </div>              </div>          </div>            <div id="hasil" class="container tab-pane fade"><br>              <h4>Hasil Training</h4>              <hr>              <div class="row">                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MSE</p>                      <img width="100%" id="img\_mse"></img>                  </div>                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MAPE</p>                      <img width="100%" id="img\_mape"></img>                  </div>              </div>              <br>              <div class="row">                  <div class="col-sm-6 mx-auto">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik Fitness</p>                      <img width="50%"  width="100%" align="center" id="img\_fitness"></img>                  </div>              </div>              <br><br>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_train"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_train"></span></p>              <p style="font-size:130%;">Nilai Fitness: <span id="fitness"></span></p>              <br>              <h4>Hasil Testing</h4>              <hr>              <p align="center" style="font-weight: bold; font-size:130%;">Grafik Perbandingan Harga Prediksi</p>              <img width="100%" id="img\_prediksi"></img>              <br><br>              <p style="font-size:130%;">Tingkat Akurasi: <span id="akurasi"></span>%</p>              <p style="font-size:130%;">Nilai MSE: <span id="mse\_test"></span></p>              <p style="font-size:130%;">Nilai MAPE: <span id="mape\_test"></span></p>          </div>        </div>  </div>  <script>      $('#proses\_btn').bind("click",function()      {          var imgVal = $('#dataset\_source').val();          if(imgVal=='')          {              alert("Masukkan file dataset!");              return false;          }            else{              $('#result\_train').show();              $('#result\_loading').show();              var data = new FormData();              data.append('dataset\_source', $('#dataset\_source')[0].files[0]);              data.append('input\_layer', $('#input\_layer').val());              data.append('hidden\_layer', $('#hidden\_layer').val());              data.append('output\_layer', $('#output\_layer').val());              data.append('epoch', $('#epoch').val());              data.append('learning\_rate', $('#learning\_rate').val());              data.append('pop\_size', $('#pop\_size').val());              data.append('epoch\_pso', $('#epoch\_pso').val());              data.append('w\_inersia', $('#w\_inersia').val());              data.append('c1', $('#c1').val());              data.append('c2', $('#c2').val());              $.ajax({                  type: 'POST',                  url: '/training-pso',                  data: data,                  contentType: false,                  cache: false,                  processData: false,                  async: true,                  success: function (data) {                      $('#mse\_train').html(data.mse);                      $('#mape\_train').html(data.mape);                      $('#akurasi').html(data.akurasi);                      $('#mse\_test').html(data.mse\_test);                      $('#mape\_test').html(data.mape\_test);                      $('#fitness').html(data.fitness);                      $('#img\_mse').attr('src','{{url\_for('static',filename='images/mse\_pso.png')}}');                      $('#img\_mape').attr('src','{{url\_for('static',filename='images/mape\_pso.png')}}');                      $('#img\_fitness').attr('src','{{url\_for('static',filename='images/fitness.png')}}');                      $('#img\_prediksi').attr('src','{{url\_for('static',filename='images/prediksi\_pso.png')}}');                      $('#result\_loading').hide();                      $('#result\_field').show();                      console.log(data);                  },                  error: function (data) {                      console.log(data);                  }              });          }      });  </script>  {% endblock %} |

*Source Code* hasil-pso.html

|  |
| --- |
| {% extends "layout.html" %}  {% block title %}Hasil BP-PSO{% endblock %}  {% block head %}  {% endblock %}  {% block content %}  <div class="container">      <h2>Hasil JST-BP - PSO</h2>      <hr>      <!-- Nav tabs -->      <ul class="nav nav-tabs" role="tablist">          <li class="nav-item">              <a class="nav-link active" data-toggle="tab" href="#training">Hasil Training</a>          </li>          <li class="nav-item">              <a class="nav-link" data-toggle="tab" href="#testing">Hasil Testing</a>          </li>      </ul>      <!-- Tab panes -->      <div class="tab-content">          <div id="training" class="container tab-pane active"><br>              <div class="row">                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MSE</p>                      <img src="{{url\_for('static',filename='images/mse\_pso.png')}}" width="100%" id="img\_mse"></img>                  </div>                  <div class="col">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik MAPE</p>                      <img src="{{url\_for('static',filename='images/mape\_pso.png')}}" width="100%" id="img\_mape"></img>                  </div>              </div>              <br>              <div class="row">                  <div class="col-sm-6 mx-auto">                      <p align="center" style="font-weight: bold; font-size:130%;">Grafik Fitness</p>                      <img src="{{url\_for('static',filename='images/fitness.png')}}" width="100%" align="center" id="img\_fitness"></img>                  </div>              </div>              <br><br>              <p style="font-size:130%;">Nilai MSE: <strong>1.96737</strong></p>              <p style="font-size:130%;">Nilai MAPE: <strong>1.85356</strong></p>              <p style="font-size:130%;">Nilai Fitness: <strong>0.9818</strong></p>          </div>            <div id="testing" class="container tab-pane fade"><br>              <p align="center" style="font-weight: bold; font-size:130%;">Grafik Perbandingan Harga Prediksi</p>              <img src="{{url\_for('static',filename='images/prediksi\_pso.png')}}" width="100%" id="img\_prediksi"></img>              <br><br>              <p style="font-size:130%;">Tingkat Akurasi: <strong>94,97993%</strong></p>              <p style="font-size:130%;">Nilai MSE: <strong>7.15827</strong></p>              <p style="font-size:130%;">Nilai MAPE: <strong>5.02007</strong></p>          </div>      </div>  </div>  {% endblock %} |

Lampiran . Kode Program Implementasi Algoritma

*Source Code* BPxPSO.py

|  |
| --- |
| import pandas as pd  from numpy import abs, exp, sqrt  from matplotlib import pyplot as plt  plt.style.use('fivethirtyeight')  def un\_normalized(value, min, max):      return value \* (max - min) + min  class Neuron:      def \_\_init\_\_(self, weight):          self.weight = weight          self.output = list()          self.delta = list()    class Backpropagation:      def \_\_init\_\_(self, input, hidden, output, learning\_rate):          self.INPUT\_LAYER = input          self.HIDDEN\_LAYER = hidden          self.OUTPUT\_LAYER = output          self.LEARNING\_RATE = learning\_rate      def initWeight(self, partikel):          layer = list()          partikel\_dimens\_idx = 0          input\_to\_hidden = list()          for i in range(self.HIDDEN\_LAYER):              w = list()              for j in range(self.INPUT\_LAYER):                  w.append(partikel[partikel\_dimens\_idx])                  partikel\_dimens\_idx += 1              w.append(1.0)              input\_to\_hidden.append(Neuron(w))          layer.append(input\_to\_hidden)          hidden\_to\_output = list()          for i in range(self.OUTPUT\_LAYER):              w = list()              for j in range(self.HIDDEN\_LAYER):                  w.append(0.1)                  partikel\_dimens\_idx += 1              w.append(1.0)              hidden\_to\_output.append(Neuron(w))          layer.append(hidden\_to\_output)          self.net = layer          return self.net        def \_\_str\_\_(self):          str\_buff = ""          str\_buff += "Input to Hidden\n"          for neuron in self.net[0]:              str\_buff += str(neuron.weight) + "\n"          str\_buff += "Hidden to Output\n"          for neuron in self.net[1]:              str\_buff += str(neuron.weight) + "\n"          return str\_buff      def summingFunction(self, last\_weight, training\_data):          bias = last\_weight[-1]  # bias on last index          y\_in = 0          for idx in range(0, len(last\_weight) - 1):              y\_in += last\_weight[idx] \* training\_data[idx]          return bias + y\_in      def activation(self, y\_in):          return 1.0 / (1.0 + exp(-1 \* y\_in))      def derivativeFunction(self, y):          return y \* (1.0 - y)      def feedForward(self, row):          processed\_layer = row          for layer in self.net:              # print(layer)              new\_inputs = list()              for neuron in layer:                  y\_in = self.summingFunction(neuron.weight, processed\_layer)                  neuron.output = self.activation(y\_in)                  new\_inputs.append(neuron.output)              processed\_layer = new\_inputs          return processed\_layer  # nilai y1, y2, .... yx      def backPpError(self, target):          for id\_layer in reversed(range(len(self.net))):              layer = self.net[id\_layer]              error\_factor = list()              if id\_layer != len(self.net) - 1:                  for weight\_id in range(len(layer)):                      error\_factor\_sum = 0.0                      prev\_layer = self.net[id\_layer - 1]                      for neuron in prev\_layer:                          error\_factor\_sum += neuron.weight[weight\_id] \* neuron.delta                      error\_factor.append(error\_factor\_sum)              else:                  for weight\_id in range(len(layer)):                      neuron = layer[weight\_id]                      error\_factor.append(target[weight\_id] - neuron.output)              for weight\_id in range(len(layer)):                  neuron = layer[weight\_id]                  neuron.delta = error\_factor[weight\_id] \* self.derivativeFunction(neuron.output)      def updateWeight(self, row):          for i in range(len(self.net)):              inputs = list()              if i == 0:                  inputs = row              else:                  prev\_net = self.net[i - 1]                  inputs = [neuron.output for neuron in prev\_net]              for neuron in self.net[i]:                  for weight\_id in range(len(neuron.weight) - 1):                      neuron.weight[weight\_id] += self.LEARNING\_RATE \* neuron.delta \* inputs[weight\_id]                  neuron.weight[-1] += self.LEARNING\_RATE \* neuron.delta      def training(self, X\_train, Y\_train, max\_value, min\_value, total\_epoch):          X\_list = X\_train.values.tolist()          Y\_list = [[x] for x in Y\_train.values.tolist()]          mselist = list()          mapelist = list()          for epoch in range(total\_epoch):              #print("Iterasi ke-{}".format(epoch))              hasil = list()              mape = 0              acc = 0              mse = 0              rmse = 0              for idx, data\_row in enumerate(Y\_list):                  self.feedForward(X\_list[idx])                  self.backPpError(Y\_list[idx])                  self.updateWeight(X\_list[idx])              for x in range(len(Y\_list)):                  actual\_value = un\_normalized(                      Y\_list[x][0], min\_value, max\_value)                  forecasted\_value = un\_normalized(                      self.predict(X\_list[x])[0], min\_value, max\_value)                  mape += abs(                      (actual\_value - forecasted\_value) / actual\_value)                  acc += (1 - (abs(actual\_value - forecasted\_value) / actual\_value))                  mse += (actual\_value - forecasted\_value)\*\*2                  hasil.append(forecasted\_value)                mape = mape \* 100 / len(Y\_train)              mse = mse / len(Y\_train)              mselist.append(mse)              mapelist.append(mape)          return mape, hasil, mse, mselist, mapelist      def predict(self, row):          output = self.feedForward(row)          return output      def testing(self, X\_test, Y\_test, max\_value, min\_value):          X\_test = X\_test.values.tolist()          mape = 0          acc = 0          mse = 0          hasil = list()          for x, data\_row in enumerate(Y\_test):              actual\_value = un\_normalized(Y\_test.iloc[x], min\_value, max\_value)              forecasted\_value = un\_normalized(                  self.predict(X\_test[x])[0], min\_value, max\_value)              mape += abs((actual\_value - forecasted\_value) / actual\_value)              mse += ((actual\_value - forecasted\_value)\*\*2)              acc += (1 - (abs(actual\_value - forecasted\_value) / actual\_value))              hasil.append(forecasted\_value)          acc = (acc/len(Y\_test)) \* 100          mse = mse / len(Y\_test)          mape = mape \* 100 / len(Y\_test)          return mape, mse, hasil, acc    class BackpropagationParticle():      def \_\_init\_\_(self, particle\_size, position, x\_train, y\_train, x\_test, y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, epoch, input\_layer, hidden\_layer, output\_layer, lr):          self.x\_train, self.y\_train = x\_train, y\_train          self.x\_test, self.y\_test = x\_test, y\_test          self.max\_val, self.min\_val, self.max\_val\_test, self.min\_val\_test = max\_val, min\_val, max\_val\_test, min\_val\_test          self.epoch = epoch          self.position = position[:particle\_size]          self.velocity = [0 for i in range(particle\_size)]          self.input\_layer = input\_layer          self.hidden\_layer = hidden\_layer          self.output\_layer = output\_layer          self.learning\_rate = lr          self.calculate\_fitness()      def calculate\_fitness(self):          particle\_position = self.position          backPro = Backpropagation(self.input\_layer, self.hidden\_layer, self.output\_layer, self.learning\_rate)          layer = backPro.initWeight(particle\_position)          mape, hasil, mse, mselist, mapelist = backPro.training(self.x\_train, self.y\_train, self.max\_val, self.min\_val, self.epoch)          mape\_test, mse\_test, hasil\_test, akurasi = backPro.testing(self.x\_test, self.y\_test, self.max\_val\_test, self.min\_val\_test)          self.mse = mse          self.mselist = mselist          self.layer = layer          self.hasil = hasil          self.mape = mape          self.mapelist = mapelist          self.fitness = 100 / (100 + mape)          self.mse\_test = mse\_test          self.mape\_test = mape\_test          self.hasil\_test = hasil\_test          self.akurasi = akurasi      class PSOxBackpro:      def \_\_init\_\_(self, pop\_size, particle\_size, k=None):          self.k = k          self.pop\_size, self.particle\_size = pop\_size, particle\_size      def set\_backpro\_param(self, x\_train, y\_train, x\_test, y\_test, max\_val, min\_val, max\_val\_test, min\_val\_test, epoch, input\_layer, hidden\_layer, output\_layer, lr):          self.x\_train, self.y\_train = x\_train, y\_train          self.x\_test, self.y\_test = x\_test, y\_test          self.max\_val, self.min\_val, self.max\_val\_test, self.min\_val\_test = max\_val, min\_val, max\_val\_test, min\_val\_test          self.input\_layer, self.hidden\_layer, self.output\_layer = input\_layer, hidden\_layer, output\_layer          self.learning\_rate = lr          self.epoch = epoch        def initPops(self):          self.pops = []          self.pops = [BackpropagationParticle(self.particle\_size,                                               [0.3522731340418094, 0.14640481330378685, 0.40211953183863935, 0.8250205036766922, 0.8731580679027796, 0.9375709821301595, 0.46340809666941096, 0.46181264237637687, 0.0696680142964593, 0.7476107133759758, 0.7257548359958343, 0.057108187494075535, 0.3548478004722213, 0.732463316869996, 0.687735063846984],                                               self.x\_train,                                               self.y\_train,                                               self.x\_test,                                               self.y\_test,                                               self.max\_val,                                               self.min\_val,                                               self.max\_val\_test,                                               self.min\_val\_test,                                               self.epoch,                                               self.input\_layer,                                               self.hidden\_layer,                                               self.output\_layer,                                               self.learning\_rate),                       BackpropagationParticle(self.particle\_size,                                               [0.2708677419584947, 0.6450666915240707, 0.15936703861710866, 0.46308287891821864, 0.8720451576365718, 0.794733018297006, 0.6899858986037402, 0.5710455300486151, 0.9534517301676679, 0.11287528772817934, 0.30413850187436364, 0.29544211876056803, 0.9914926713999799, 0.520462476537912, 0.595110313291048],                                               self.x\_train,                                               self.y\_train,                                               self.x\_test,                                               self.y\_test,                                               self.max\_val,                                               self.min\_val,                                               self.max\_val\_test,                                               self.min\_val\_test,                                               self.epoch,                                               self.input\_layer,                                               self.hidden\_layer,                                               self.output\_layer,                                               self.learning\_rate),                       BackpropagationParticle(self.particle\_size,                                               [0.9123195797852695, 0.729751487740457, 0.66721527865428, 0.6943317526053253, 0.725663986687464, 0.31209549507882184, 0.5041401803880817, 0.8478802858108301, 0.17729628989731527, 0.6314889106879424, 0.007502619262353072, 0.6516972327659001, 0.3977139492706012, 0.7162422737613754, 0.3511981242881578],                                               self.x\_train,                                               self.y\_train,                                               self.x\_test,                                               self.y\_test,                                               self.max\_val,                                               self.min\_val,                                               self.max\_val\_test,                                               self.min\_val\_test,                                               self.epoch,                                               self.input\_layer,                                               self.hidden\_layer,                                               self.output\_layer,                                               self.learning\_rate),                       BackpropagationParticle(self.particle\_size,                                               [0.5202128458395567, 0.14826958944021573, 0.40083581910346544, 0.48201114022577707, 0.43361361014923605, 0.7125799411626561, 0.6535930501044525, 0.7471175140617415, 0.5484028429172164, 0.678891317756171, 0.48463003008830596, 0.8111001201500203, 0.8638027031113223, 0.8229387174243393, 0.5941306399864599],                                               self.x\_train,                                               self.y\_train,                                               self.x\_test,                                               self.y\_test,                                               self.max\_val,                                               self.min\_val,                                               self.max\_val\_test,                                               self.min\_val\_test,                                               self.epoch,                                               self.input\_layer,                                               self.hidden\_layer,                                               self.output\_layer,                                               self.learning\_rate),                       BackpropagationParticle(self.particle\_size,                                               [0.7017622890089452, 0.06074386128400722, 0.2649597043914179, 0.8329272218560311, 0.061032875245332074, 0.05882832820432249, 0.020558395778173644, 0.5353163907776635, 0.2649515151700631, 0.439958599587454, 0.8707022023553562, 0.6839902283639202, 0.7868106107838259, 0.21991702513525313, 0.21321731083919848],                                               self.x\_train,                                               self.y\_train,                                               self.x\_test,                                               self.y\_test,                                               self.max\_val,                                               self.min\_val,                                               self.max\_val\_test,                                               self.min\_val\_test,                                               self.epoch,                                               self.input\_layer,                                               self.hidden\_layer,                                               self.output\_layer,                                               self.learning\_rate),                      ]          self.p\_best = self.pops          self.g\_best = self.get\_g\_best()        def get\_g\_best(self):          p\_best\_sorted = self.p\_best          p\_best\_sorted.sort(key=lambda x: x.fitness, reverse=True)          return p\_best\_sorted[0]        def velocity\_clamping(self, vnew):          if self.k is None:              return vnew          else:              vmax = self.k \* (1 - 0) / 2              vmin = -1 \* vmax              if(vnew > vmax):                  return vmax              elif(vnew < vmin):                  return vmin              else:                  return vnew      def update\_velocity\_and\_position(self, w, c1, c2):          updated\_pops = self.pops          listr1 = [0.02339881415026357, 0.8879191269632195, 0.9632800020094943, 0.10266577517415809, 0.14650702407980176, 0.376193776215101, 0.147132558767758, 0.1342779888808232, 0.13411897929647665, 0.8461189179113258, 0.25804798627339987, 0.24051380102908615, 0.4722685089703986, 0.11261238313660848, 0.5088911829181822, 0.29883594538296787, 0.4676827463042148, 0.5046365961994549, 0.8868468116671602, 0.6005925483617235, 0.1008507273083532, 0.8285463187443176, 0.3286678002125677, 0.8367576299430859, 0.1097144812275479, 0.914346530489158, 0.6340722645649579, 0.32087892851179645, 0.7142394674465194, 0.03070040923848194, 0.0968714235736674, 0.5704022244950279, 0.6443860096435277, 0.5575185820227897, 0.06184318712576731, 0.5736601375392951, 0.8349003745093522, 0.4133828574725995, 0.4404562206366577, 0.07141411349151872, 0.30844888586009267, 0.6178487307985877, 0.13794809910784422, 0.5199215596734195, 0.6232494264721372, 0.8457063108838302, 0.6553404462259736, 0.1762173473711347, 0.40419473510085036, 0.08706844564405358, 0.2288559189350382, 0.2830835075855691, 0.9743614795082921, 0.7882494625948815, 0.23994350286280808, 0.7667594810591023, 0.14483508290958536, 0.4600730878096103, 0.7353707618804668, 0.8244126129491081, 0.1998603152818086, 0.9363790071622874, 0.32314371844272305, 0.23068230379695598, 0.6653784076564732, 0.05913161047878479, 0.37820309351670056, 0.31820996815622815, 0.33533437992176107, 0.5532915526367381, 0.5637908500399601, 0.6212728470001214, 0.1113109800686688, 0.1700294893391222, 0.9557995894745693]          listr2 = [0.35488007726168247, 0.7862765993455152, 0.4336464325856312, 0.45751702992693666, 0.33425441322266836, 0.021979474682622957, 0.827935680082634, 0.7135118081147601, 0.342098777964703, 0.935421664386896, 0.35699587336373795, 0.3939418710369257, 0.3607558921924392, 0.6203271190102304, 0.137934628721607, 0.7495263784416939, 0.19813155247606817, 0.0005630382490691144, 0.03977523693548679, 0.7056785668387272, 0.6898211796561455, 0.24772995808981002, 0.5210302609905166, 0.7893929141665976, 0.3703090922024497, 0.08304268437737283, 0.9202784099909919, 0.18778560867010707, 0.9858330941850937, 0.5972863851512671, 0.22370428544387588, 0.26377571320439885, 0.4082395212528338, 0.38751726114268625, 0.8253286867199064, 0.508709374859329, 0.8918508838198235, 0.754983057923608, 0.6883624944823127, 0.7744024488290014, 0.6027504602113605, 0.9397056025729577, 0.4423582945812111, 0.4228976156034633, 0.6775876149598414, 0.49922825225527834, 0.46168049452776694, 0.5329284493212804, 0.5573610905306373, 0.15691150038301394, 0.6980758854177472, 0.6993896705901846, 0.5212858484961309, 0.6029356312185978, 0.2170450438480448, 0.015853652083170577, 0.00703525571341812, 0.38936753634310406, 0.30146553549101596, 0.7307195341106, 0.19216705239885623, 0.8178247310491898, 0.9850762554696315, 0.15459025725962872, 0.7549575365372728, 0.46736890223348626, 0.15980005369699346, 0.6282187339765569, 0.7259758495980768, 0.5228849057175144, 0.04087416369928265, 0.9332240255204498, 0.06306324868276736, 0.810002479096422, 0.9221782820590523]          x = 0          for partc\_id, particle in enumerate(self.pops):              for partc\_id\_dimen in range(len(particle.velocity)):                  vnew = (                      (w \* particle.velocity[partc\_id\_dimen]) +                      (c1 \* listr1[x] \*                      (self.p\_best[partc\_id].position[partc\_id\_dimen] -                          particle.position[partc\_id\_dimen])) +                          (c2 \* listr2[x] \*                      (self.g\_best.position[partc\_id\_dimen] -                          particle.position[partc\_id\_dimen]))                  )                  if self.k is not None:                      vnew = self.velocity\_clamping(vnew)                  updated\_pops[partc\_id].velocity[partc\_id\_dimen] = vnew                  xnew = particle.position[partc\_id\_dimen] + vnew                  updated\_pops[partc\_id].position[partc\_id\_dimen] = xnew                    x += 1              updated\_pops[partc\_id].calculate\_fitness()              #if (updated\_pops[partc\_id].fitness < self.p\_best[partc\_id].fitness):              #    updated\_pops[partc\_id] = self.p\_best[partc\_id]          # mengganti partikel yang lama dengan partikel yang telah diupdate          self.pops = updated\_pops      def update\_p\_best(self):          for partc\_id in range(len(self.pops)):              if(self.pops[partc\_id].fitness > self.p\_best[partc\_id].fitness):                  self.p\_best[partc\_id] = self.pops[partc\_id]      def get\_average\_fitness(self):          sum\_fitness = 0          for partc in self.pops:              sum\_fitness += partc.fitness          return sum\_fitness / len(self.pops)      def optimize(self, tmax, w, c1, c2):          t = 0   # t awal diset ke 0          #print("Popsize: ", len(self.pops), ", Itermax: ", tmax)          #print("w: ", w, ", c1: ", c1, ", c2: ", c2)          #print("k: ", self.k)          #for p in self.pops:          #    print(p.fitness)          self.listmse = list()          self.listmape = list()          self.listfitness = list()          while(t < tmax):              print("--------")              print("|t = ", (t + 1), "|")              self.update\_velocity\_and\_position(w, c1, c2)              self.update\_p\_best()              self.g\_best = self.get\_g\_best()              #print("Global Best Position: ", self.g\_best.position)              #print("Global Best Particle Velocity: ", self.g\_best.velocity)              #print("Global Best Fitness: ", self.g\_best.fitness)              #print("Average Fitness: ", self.get\_average\_fitness())              #print("-------------------------------------------------------")              t += 1              self.listfitness.append(self.g\_best.fitness)              self.listmse.append(self.g\_best.mse)              self.listmape.append(self.g\_best.mape)          return self.g\_best.position, self.g\_best.fitness, self.get\_average\_fitness(), self.listfitness, self.listmse, self.listmape |

Lampiran . Crude Oil WTI (CL=F) Dataset

*Datasets* wti.csv

|  |  |
| --- | --- |
| Date | Close |
| 1/3/2017 | 52.33 |
| 1/4/2017 | 53.26 |
| 1/5/2017 | 53.76 |
| 1/6/2017 | 53.99 |
| 1/9/2017 | 51.96 |
| 1/10/2017 | 50.82 |
| 1/11/2017 | 52.25 |
| 1/12/2017 | 53.01 |
| 1/13/2017 | 52.37 |
| 1/17/2017 | 52.48 |
| 1/18/2017 | 51.08 |
| 1/19/2017 | 51.37 |
| 1/20/2017 | 52.42 |
| 1/23/2017 | 52.75 |
| 1/24/2017 | 53.18 |
| 1/25/2017 | 52.75 |
| 1/26/2017 | 53.78 |
| 1/27/2017 | 53.17 |
| 1/30/2017 | 52.63 |
| 1/31/2017 | 52.81 |
| 2/1/2017 | 53.88 |
| 2/2/2017 | 53.54 |
| 2/3/2017 | 53.83 |
| 2/6/2017 | 53.01 |
| 2/7/2017 | 52.17 |
| 2/8/2017 | 52.34 |
| 2/9/2017 | 53 |
| 2/10/2017 | 53.86 |
| 2/13/2017 | 52.93 |
| 2/14/2017 | 53.2 |
| 2/15/2017 | 53.11 |
| 2/16/2017 | 53.36 |
| 2/17/2017 | 53.4 |
| 2/21/2017 | 54.06 |
| 2/22/2017 | 53.59 |
| 2/23/2017 | 54.45 |
| 2/24/2017 | 53.99 |
| 2/27/2017 | 54.05 |
| 2/28/2017 | 54.01 |
| 3/1/2017 | 53.83 |
| 3/2/2017 | 52.61 |
| 3/3/2017 | 53.33 |
| 3/6/2017 | 53.2 |
| 3/7/2017 | 53.14 |
| 3/8/2017 | 50.28 |
| 3/9/2017 | 49.28 |
| 3/10/2017 | 48.49 |
| 3/13/2017 | 48.4 |
| 3/14/2017 | 47.72 |
| 3/15/2017 | 48.86 |
| 3/16/2017 | 48.75 |
| 3/17/2017 | 48.78 |
| 3/20/2017 | 48.22 |
| 3/21/2017 | 47.34 |
| 3/22/2017 | 48.04 |
| 3/23/2017 | 47.7 |
| 3/24/2017 | 47.97 |
| 3/27/2017 | 47.73 |
| 3/28/2017 | 48.37 |
| 3/29/2017 | 49.51 |
| 3/30/2017 | 50.35 |
| 3/31/2017 | 50.6 |
| 4/3/2017 | 50.24 |
| 4/4/2017 | 51.03 |
| 4/5/2017 | 51.15 |
| 4/6/2017 | 51.7 |
| 4/7/2017 | 52.24 |
| 4/10/2017 | 53.08 |
| 4/11/2017 | 53.4 |
| 4/12/2017 | 53.11 |
| 4/13/2017 | 53.18 |
| 4/17/2017 | 52.65 |
| 4/18/2017 | 52.41 |
| 4/19/2017 | 50.44 |
| 4/20/2017 | 50.27 |
| 4/21/2017 | 49.62 |
| 4/24/2017 | 49.23 |
| 4/25/2017 | 49.56 |
| 4/26/2017 | 49.62 |
| 4/27/2017 | 48.97 |
| 4/28/2017 | 49.33 |
| 5/1/2017 | 48.84 |
| 5/2/2017 | 47.66 |
| 5/3/2017 | 47.82 |
| 5/4/2017 | 45.52 |
| 5/5/2017 | 46.22 |
| 5/8/2017 | 46.43 |
| 5/9/2017 | 45.88 |
| 5/10/2017 | 47.33 |
| 5/11/2017 | 47.83 |
| 5/12/2017 | 47.84 |
| 5/15/2017 | 48.85 |
| 5/16/2017 | 48.66 |
| 5/17/2017 | 49.07 |
| 5/18/2017 | 49.35 |
| 5/19/2017 | 50.33 |
| 5/22/2017 | 50.73 |
| 5/23/2017 | 51.47 |
| 5/24/2017 | 51.36 |
| 5/25/2017 | 48.9 |
| 5/26/2017 | 49.8 |
| 5/30/2017 | 49.66 |
| 5/31/2017 | 48.32 |
| 6/1/2017 | 48.36 |
| 6/2/2017 | 47.66 |
| 6/5/2017 | 47.4 |
| 6/6/2017 | 48.19 |
| 6/7/2017 | 45.72 |
| 6/8/2017 | 45.64 |
| 6/9/2017 | 45.83 |
| 6/12/2017 | 46.08 |
| 6/13/2017 | 46.46 |
| 6/14/2017 | 44.73 |
| 6/15/2017 | 44.46 |
| 6/16/2017 | 44.74 |
| 6/19/2017 | 44.2 |
| 6/20/2017 | 43.23 |
| 6/21/2017 | 42.53 |
| 6/22/2017 | 42.74 |
| 6/23/2017 | 43.01 |
| 6/26/2017 | 43.38 |
| 6/27/2017 | 44.24 |
| 6/28/2017 | 44.74 |
| 6/29/2017 | 44.93 |
| 6/30/2017 | 46.04 |
| 7/5/2017 | 45.13 |
| 7/6/2017 | 45.52 |
| 7/7/2017 | 44.23 |
| 7/10/2017 | 44.4 |
| 7/11/2017 | 45.04 |
| 7/12/2017 | 45.49 |
| 7/13/2017 | 46.08 |
| 7/14/2017 | 46.54 |
| 7/17/2017 | 46.02 |
| 7/18/2017 | 46.4 |
| 7/19/2017 | 47.12 |
| 7/20/2017 | 46.79 |
| 7/21/2017 | 45.77 |
| 7/24/2017 | 46.34 |
| 7/25/2017 | 47.89 |
| 7/26/2017 | 48.75 |
| 7/27/2017 | 49.04 |
| 7/28/2017 | 49.71 |
| 7/31/2017 | 50.17 |
| 8/1/2017 | 49.16 |
| 8/2/2017 | 49.59 |
| 8/3/2017 | 49.03 |
| 8/4/2017 | 49.58 |
| 8/7/2017 | 49.39 |
| 8/8/2017 | 49.17 |
| 8/9/2017 | 49.56 |
| 8/10/2017 | 48.59 |
| 8/11/2017 | 48.82 |
| 8/14/2017 | 47.59 |
| 8/15/2017 | 47.55 |
| 8/16/2017 | 46.78 |
| 8/17/2017 | 47.09 |
| 8/18/2017 | 48.51 |
| 8/21/2017 | 47.37 |
| 8/22/2017 | 47.64 |
| 8/23/2017 | 48.41 |
| 8/24/2017 | 47.43 |
| 8/25/2017 | 47.87 |
| 8/28/2017 | 46.57 |
| 8/29/2017 | 46.44 |
| 8/30/2017 | 45.96 |
| 8/31/2017 | 47.23 |
| 9/1/2017 | 47.29 |
| 9/5/2017 | 48.66 |
| 9/6/2017 | 49.16 |
| 9/7/2017 | 49.09 |
| 9/8/2017 | 47.48 |
| 9/11/2017 | 48.07 |
| 9/12/2017 | 48.23 |
| 9/13/2017 | 49.3 |
| 9/14/2017 | 49.89 |
| 9/15/2017 | 49.89 |
| 9/18/2017 | 49.91 |
| 9/19/2017 | 49.48 |
| 9/20/2017 | 50.41 |
| 9/21/2017 | 50.55 |
| 9/22/2017 | 50.66 |
| 9/25/2017 | 52.22 |
| 9/26/2017 | 51.88 |
| 9/27/2017 | 52.14 |
| 9/28/2017 | 51.56 |
| 9/29/2017 | 51.67 |
| 10/2/2017 | 50.58 |
| 10/3/2017 | 50.42 |
| 10/4/2017 | 49.98 |
| 10/5/2017 | 50.79 |
| 10/6/2017 | 49.29 |
| 10/9/2017 | 49.58 |
| 10/10/2017 | 50.92 |
| 10/11/2017 | 51.3 |
| 10/12/2017 | 50.6 |
| 10/13/2017 | 51.45 |
| 10/16/2017 | 51.87 |
| 10/17/2017 | 51.88 |
| 10/18/2017 | 52.04 |
| 10/19/2017 | 51.29 |
| 10/20/2017 | 51.47 |
| 10/23/2017 | 51.9 |
| 10/24/2017 | 52.47 |
| 10/25/2017 | 52.18 |
| 10/26/2017 | 52.64 |
| 10/27/2017 | 53.9 |
| 10/30/2017 | 54.15 |
| 10/31/2017 | 54.38 |
| 11/1/2017 | 54.3 |
| 11/2/2017 | 54.54 |
| 11/3/2017 | 55.64 |
| 11/6/2017 | 57.35 |
| 11/7/2017 | 57.2 |
| 11/8/2017 | 56.81 |
| 11/9/2017 | 57.17 |
| 11/10/2017 | 56.74 |
| 11/13/2017 | 56.76 |
| 11/14/2017 | 55.7 |
| 11/15/2017 | 55.33 |
| 11/16/2017 | 55.14 |
| 11/17/2017 | 56.55 |
| 11/20/2017 | 56.09 |
| 11/21/2017 | 56.83 |
| 11/22/2017 | 58.02 |
| 11/27/2017 | 58.11 |
| 11/28/2017 | 57.99 |
| 11/29/2017 | 57.3 |
| 11/30/2017 | 57.4 |
| 12/1/2017 | 58.36 |
| 12/4/2017 | 57.47 |
| 12/5/2017 | 57.62 |
| 12/6/2017 | 55.96 |
| 12/7/2017 | 56.69 |
| 12/8/2017 | 57.36 |
| 12/11/2017 | 57.99 |
| 12/12/2017 | 57.14 |
| 12/13/2017 | 56.6 |
| 12/14/2017 | 57.04 |
| 12/15/2017 | 57.3 |
| 12/18/2017 | 57.16 |
| 12/19/2017 | 57.46 |
| 12/20/2017 | 58.09 |
| 12/21/2017 | 58.36 |
| 12/22/2017 | 58.47 |
| 12/26/2017 | 59.97 |
| 12/27/2017 | 59.64 |
| 12/28/2017 | 59.84 |
| 12/29/2017 | 60.42 |
| 1/2/2018 | 60.37 |
| 1/3/2018 | 61.63 |
| 1/4/2018 | 62.01 |
| 1/5/2018 | 61.44 |
| 1/8/2018 | 61.73 |
| 1/9/2018 | 62.96 |
| 1/10/2018 | 63.57 |
| 1/11/2018 | 63.8 |
| 1/12/2018 | 64.3 |
| 1/16/2018 | 63.73 |
| 1/17/2018 | 63.97 |
| 1/18/2018 | 63.95 |
| 1/19/2018 | 63.37 |
| 1/22/2018 | 63.49 |
| 1/23/2018 | 64.47 |
| 1/24/2018 | 65.61 |
| 1/25/2018 | 65.51 |
| 1/26/2018 | 66.14 |
| 1/29/2018 | 65.56 |
| 1/30/2018 | 64.5 |
| 1/31/2018 | 64.73 |
| 2/1/2018 | 65.8 |
| 2/2/2018 | 65.45 |
| 2/5/2018 | 64.15 |
| 2/6/2018 | 63.39 |
| 2/7/2018 | 61.79 |
| 2/8/2018 | 61.15 |
| 2/9/2018 | 59.2 |
| 2/12/2018 | 59.29 |
| 2/13/2018 | 59.19 |
| 2/14/2018 | 60.6 |
| 2/15/2018 | 61.34 |
| 2/16/2018 | 61.68 |
| 2/20/2018 | 61.9 |
| 2/21/2018 | 61.68 |
| 2/22/2018 | 62.77 |
| 2/23/2018 | 63.55 |
| 2/26/2018 | 63.91 |
| 2/27/2018 | 63.01 |
| 2/28/2018 | 61.64 |
| 3/1/2018 | 60.99 |
| 3/2/2018 | 61.25 |
| 3/5/2018 | 62.57 |
| 3/6/2018 | 62.6 |
| 3/7/2018 | 61.15 |
| 3/8/2018 | 60.12 |
| 3/9/2018 | 62.04 |
| 3/12/2018 | 61.36 |
| 3/13/2018 | 60.71 |
| 3/14/2018 | 60.96 |
| 3/15/2018 | 61.19 |
| 3/16/2018 | 62.34 |
| 3/19/2018 | 62.06 |
| 3/20/2018 | 63.4 |
| 3/21/2018 | 65.17 |
| 3/22/2018 | 64.3 |
| 3/23/2018 | 65.88 |
| 3/26/2018 | 65.55 |
| 3/27/2018 | 65.25 |
| 3/28/2018 | 64.38 |
| 3/29/2018 | 64.94 |
| 4/2/2018 | 63.01 |
| 4/3/2018 | 63.51 |
| 4/4/2018 | 63.37 |
| 4/5/2018 | 63.54 |
| 4/6/2018 | 62.06 |
| 4/9/2018 | 63.42 |
| 4/10/2018 | 65.51 |
| 4/11/2018 | 66.82 |
| 4/12/2018 | 67.07 |
| 4/13/2018 | 67.39 |
| 4/16/2018 | 66.22 |
| 4/17/2018 | 66.52 |
| 4/18/2018 | 68.47 |
| 4/19/2018 | 68.29 |
| 4/20/2018 | 68.38 |
| 4/23/2018 | 68.64 |
| 4/24/2018 | 67.7 |
| 4/25/2018 | 68.05 |
| 4/26/2018 | 68.19 |
| 4/27/2018 | 68.1 |
| 4/30/2018 | 68.57 |
| 5/1/2018 | 67.25 |
| 5/2/2018 | 67.93 |
| 5/3/2018 | 68.43 |
| 5/4/2018 | 69.72 |
| 5/7/2018 | 70.73 |
| 5/8/2018 | 69.06 |
| 5/9/2018 | 71.14 |
| 5/10/2018 | 71.36 |
| 5/11/2018 | 70.7 |
| 5/14/2018 | 70.96 |
| 5/15/2018 | 71.31 |
| 5/16/2018 | 71.49 |
| 5/17/2018 | 71.49 |
| 5/18/2018 | 71.28 |
| 5/21/2018 | 72.24 |
| 5/22/2018 | 72.13 |
| 5/23/2018 | 71.84 |
| 5/24/2018 | 70.71 |
| 5/25/2018 | 67.88 |
| 5/29/2018 | 66.73 |
| 5/30/2018 | 68.21 |
| 5/31/2018 | 67.04 |
| 6/1/2018 | 65.81 |
| 6/4/2018 | 64.75 |
| 6/5/2018 | 65.52 |
| 6/6/2018 | 64.73 |
| 6/7/2018 | 65.95 |
| 6/8/2018 | 65.74 |
| 6/11/2018 | 66.1 |
| 6/12/2018 | 66.36 |
| 6/13/2018 | 66.64 |
| 6/14/2018 | 66.89 |
| 6/15/2018 | 65.06 |
| 6/18/2018 | 65.85 |
| 6/19/2018 | 65.07 |
| 6/20/2018 | 66.22 |
| 6/21/2018 | 65.54 |
| 6/22/2018 | 68.58 |
| 6/25/2018 | 68.08 |
| 6/26/2018 | 70.53 |
| 6/27/2018 | 72.76 |
| 6/28/2018 | 73.45 |
| 6/29/2018 | 74.15 |
| 7/2/2018 | 73.94 |
| 7/5/2018 | 72.94 |
| 7/6/2018 | 73.8 |
| 7/9/2018 | 73.85 |
| 7/10/2018 | 74.11 |
| 7/11/2018 | 70.38 |
| 7/12/2018 | 70.33 |
| 7/13/2018 | 71.01 |
| 7/16/2018 | 68.06 |
| 7/17/2018 | 68.08 |
| 7/18/2018 | 68.76 |
| 7/19/2018 | 69.46 |
| 7/20/2018 | 70.46 |
| 7/23/2018 | 67.89 |
| 7/24/2018 | 68.52 |
| 7/25/2018 | 69.3 |
| 7/26/2018 | 69.61 |
| 7/27/2018 | 68.69 |
| 7/30/2018 | 70.13 |
| 7/31/2018 | 68.76 |
| 8/1/2018 | 67.66 |
| 8/2/2018 | 68.96 |
| 8/3/2018 | 68.49 |
| 8/6/2018 | 69.01 |
| 8/7/2018 | 69.17 |
| 8/8/2018 | 66.94 |
| 8/9/2018 | 66.81 |
| 8/10/2018 | 67.63 |
| 8/13/2018 | 67.2 |
| 8/14/2018 | 67.04 |
| 8/15/2018 | 65.01 |
| 8/16/2018 | 65.46 |
| 8/17/2018 | 65.91 |
| 8/20/2018 | 66.43 |
| 8/21/2018 | 67.35 |
| 8/22/2018 | 67.86 |
| 8/23/2018 | 67.83 |
| 8/24/2018 | 68.72 |
| 8/27/2018 | 68.87 |
| 8/28/2018 | 68.53 |
| 8/29/2018 | 69.51 |
| 8/30/2018 | 70.25 |
| 8/31/2018 | 69.8 |
| 9/4/2018 | 69.87 |
| 9/5/2018 | 68.72 |
| 9/6/2018 | 67.77 |
| 9/7/2018 | 67.75 |
| 9/10/2018 | 67.54 |
| 9/11/2018 | 69.25 |
| 9/12/2018 | 70.37 |
| 9/13/2018 | 68.59 |
| 9/14/2018 | 68.99 |
| 9/17/2018 | 68.91 |
| 9/18/2018 | 69.85 |
| 9/19/2018 | 71.12 |
| 9/20/2018 | 70.8 |
| 9/21/2018 | 70.78 |
| 9/24/2018 | 72.08 |
| 9/25/2018 | 72.28 |
| 9/26/2018 | 71.57 |
| 9/27/2018 | 72.12 |
| 9/28/2018 | 73.25 |
| 10/1/2018 | 75.3 |
| 10/2/2018 | 75.23 |
| 10/3/2018 | 76.41 |
| 10/4/2018 | 74.33 |
| 10/5/2018 | 74.34 |
| 10/8/2018 | 74.29 |
| 10/9/2018 | 74.96 |
| 10/10/2018 | 73.17 |
| 10/11/2018 | 70.97 |
| 10/12/2018 | 71.34 |
| 10/15/2018 | 71.78 |
| 10/16/2018 | 71.92 |
| 10/17/2018 | 69.75 |
| 10/18/2018 | 68.65 |
| 10/19/2018 | 69.12 |
| 10/22/2018 | 69.17 |
| 10/23/2018 | 66.43 |
| 10/24/2018 | 66.82 |
| 10/25/2018 | 67.33 |
| 10/26/2018 | 67.59 |
| 10/29/2018 | 67.04 |
| 10/30/2018 | 66.18 |
| 10/31/2018 | 65.31 |
| 11/1/2018 | 63.69 |
| 11/2/2018 | 63.14 |
| 11/5/2018 | 63.1 |
| 11/6/2018 | 62.21 |
| 11/7/2018 | 61.67 |
| 11/8/2018 | 60.67 |
| 11/9/2018 | 60.19 |
| 11/12/2018 | 59.93 |
| 11/13/2018 | 55.69 |
| 11/14/2018 | 56.25 |
| 11/15/2018 | 56.46 |
| 11/16/2018 | 56.46 |
| 11/19/2018 | 56.76 |
| 11/20/2018 | 53.43 |
| 11/21/2018 | 54.63 |
| 11/26/2018 | 51.63 |
| 11/27/2018 | 51.56 |
| 11/28/2018 | 50.29 |
| 11/29/2018 | 51.45 |
| 11/30/2018 | 50.93 |
| 12/3/2018 | 52.95 |
| 12/4/2018 | 53.25 |
| 12/6/2018 | 51.49 |
| 12/7/2018 | 52.61 |
| 12/10/2018 | 51 |
| 12/11/2018 | 51.65 |
| 12/12/2018 | 51.15 |
| 12/13/2018 | 52.58 |
| 12/14/2018 | 51.2 |
| 12/17/2018 | 49.88 |
| 12/18/2018 | 46.24 |
| 12/19/2018 | 47.2 |
| 12/20/2018 | 45.88 |
| 12/21/2018 | 45.59 |
| 12/26/2018 | 46.22 |
| 12/27/2018 | 44.61 |
| 12/28/2018 | 45.33 |
| 12/31/2018 | 45.41 |
| 1/2/2019 | 46.54 |
| 1/3/2019 | 47.09 |
| 1/4/2019 | 47.96 |
| 1/7/2019 | 48.52 |
| 1/8/2019 | 49.78 |
| 1/9/2019 | 52.36 |
| 1/10/2019 | 52.59 |
| 1/11/2019 | 51.59 |
| 1/14/2019 | 50.51 |
| 1/15/2019 | 52.11 |
| 1/16/2019 | 52.31 |
| 1/17/2019 | 52.07 |
| 1/18/2019 | 53.8 |
| 1/22/2019 | 52.57 |
| 1/23/2019 | 52.62 |
| 1/24/2019 | 53.13 |
| 1/25/2019 | 53.69 |
| 1/28/2019 | 51.99 |
| 1/29/2019 | 53.31 |
| 1/30/2019 | 54.23 |
| 1/31/2019 | 53.79 |
| 2/1/2019 | 55.26 |
| 2/4/2019 | 54.56 |
| 2/5/2019 | 53.66 |
| 2/6/2019 | 54.01 |
| 2/7/2019 | 52.64 |
| 2/8/2019 | 52.72 |
| 2/11/2019 | 52.41 |
| 2/12/2019 | 53.1 |
| 2/13/2019 | 53.9 |
| 2/14/2019 | 54.41 |
| 2/15/2019 | 55.59 |
| 2/19/2019 | 56.09 |
| 2/20/2019 | 56.92 |
| 2/21/2019 | 56.96 |
| 2/22/2019 | 57.26 |
| 2/25/2019 | 55.48 |
| 2/26/2019 | 55.5 |
| 2/27/2019 | 56.94 |
| 2/28/2019 | 57.22 |
| 3/1/2019 | 55.8 |
| 3/4/2019 | 56.59 |
| 3/5/2019 | 56.56 |
| 3/6/2019 | 56.22 |
| 3/7/2019 | 56.66 |
| 3/8/2019 | 56.07 |
| 3/11/2019 | 56.79 |
| 3/12/2019 | 56.87 |
| 3/13/2019 | 58.26 |
| 3/14/2019 | 58.61 |
| 3/15/2019 | 58.52 |
| 3/18/2019 | 59.09 |
| 3/19/2019 | 59.03 |
| 3/20/2019 | 59.83 |
| 3/21/2019 | 59.98 |
| 3/22/2019 | 59.04 |
| 3/25/2019 | 58.82 |
| 3/26/2019 | 59.94 |
| 3/27/2019 | 59.41 |
| 3/28/2019 | 59.3 |
| 3/29/2019 | 60.14 |
| 4/1/2019 | 61.59 |
| 4/2/2019 | 62.58 |
| 4/3/2019 | 62.46 |
| 4/4/2019 | 62.1 |
| 4/5/2019 | 63.08 |
| 4/8/2019 | 64.4 |
| 4/9/2019 | 63.98 |
| 4/10/2019 | 64.61 |
| 4/11/2019 | 63.58 |
| 4/12/2019 | 63.89 |
| 4/15/2019 | 63.4 |
| 4/16/2019 | 64.05 |
| 4/17/2019 | 63.76 |
| 4/18/2019 | 64 |
| 4/22/2019 | 65.7 |
| 4/23/2019 | 66.3 |
| 4/24/2019 | 65.89 |
| 4/25/2019 | 65.21 |
| 4/26/2019 | 63.3 |
| 4/29/2019 | 63.5 |
| 4/30/2019 | 63.91 |
| 5/1/2019 | 63.6 |
| 5/2/2019 | 61.81 |
| 5/3/2019 | 61.94 |
| 5/6/2019 | 62.25 |
| 5/7/2019 | 61.4 |
| 5/8/2019 | 62.12 |
| 5/9/2019 | 61.7 |
| 5/10/2019 | 61.66 |
| 5/13/2019 | 61.04 |
| 5/14/2019 | 61.78 |
| 5/15/2019 | 62.02 |
| 5/16/2019 | 62.87 |
| 5/17/2019 | 62.76 |
| 5/20/2019 | 63.1 |
| 5/21/2019 | 62.99 |
| 5/22/2019 | 61.42 |
| 5/23/2019 | 57.91 |
| 5/24/2019 | 58.63 |
| 5/28/2019 | 59.14 |
| 5/29/2019 | 58.81 |
| 5/30/2019 | 56.59 |
| 5/31/2019 | 53.5 |
| 6/3/2019 | 53.25 |
| 6/4/2019 | 53.48 |
| 6/5/2019 | 51.68 |
| 6/6/2019 | 52.59 |
| 6/7/2019 | 53.99 |
| 6/10/2019 | 53.26 |
| 6/11/2019 | 53.27 |
| 6/12/2019 | 51.14 |
| 6/13/2019 | 52.28 |
| 6/14/2019 | 52.51 |
| 6/17/2019 | 51.93 |
| 6/18/2019 | 53.9 |
| 6/19/2019 | 53.76 |
| 6/20/2019 | 56.65 |
| 6/21/2019 | 57.43 |
| 6/24/2019 | 57.9 |
| 6/25/2019 | 57.83 |
| 6/26/2019 | 59.38 |
| 6/27/2019 | 59.43 |
| 6/28/2019 | 58.47 |
| 7/1/2019 | 59.09 |
| 7/2/2019 | 56.25 |
| 7/5/2019 | 57.51 |
| 7/8/2019 | 57.66 |
| 7/9/2019 | 57.83 |
| 7/10/2019 | 60.43 |
| 7/11/2019 | 60.2 |
| 7/12/2019 | 60.21 |
| 7/15/2019 | 59.58 |
| 7/16/2019 | 57.62 |
| 7/17/2019 | 56.78 |
| 7/18/2019 | 55.3 |
| 7/19/2019 | 55.63 |
| 7/22/2019 | 56.22 |
| 7/23/2019 | 56.77 |
| 7/24/2019 | 55.88 |
| 7/25/2019 | 56.02 |
| 7/26/2019 | 56.2 |
| 7/29/2019 | 56.87 |
| 7/30/2019 | 58.05 |
| 7/31/2019 | 58.58 |
| 8/1/2019 | 53.95 |
| 8/2/2019 | 55.66 |
| 8/5/2019 | 54.69 |
| 8/6/2019 | 53.63 |
| 8/7/2019 | 51.09 |
| 8/8/2019 | 52.54 |
| 8/9/2019 | 54.5 |
| 8/12/2019 | 54.93 |
| 8/13/2019 | 57.1 |
| 8/14/2019 | 55.23 |
| 8/15/2019 | 54.47 |
| 8/16/2019 | 54.87 |
| 8/19/2019 | 56.21 |
| 8/20/2019 | 56.34 |
| 8/21/2019 | 55.68 |
| 8/22/2019 | 55.35 |
| 8/23/2019 | 54.17 |
| 8/26/2019 | 53.64 |
| 8/27/2019 | 54.93 |
| 8/28/2019 | 55.78 |
| 8/29/2019 | 56.71 |
| 8/30/2019 | 55.1 |
| 9/3/2019 | 53.94 |
| 9/4/2019 | 56.26 |
| 9/5/2019 | 56.3 |
| 9/6/2019 | 56.52 |
| 9/9/2019 | 57.85 |
| 9/10/2019 | 57.4 |
| 9/11/2019 | 55.75 |
| 9/12/2019 | 55.09 |
| 9/13/2019 | 54.85 |
| 9/16/2019 | 62.9 |
| 9/17/2019 | 59.34 |
| 9/18/2019 | 58.11 |
| 9/19/2019 | 58.13 |
| 9/20/2019 | 58.09 |
| 9/23/2019 | 58.64 |
| 9/24/2019 | 57.29 |
| 9/25/2019 | 56.49 |
| 9/26/2019 | 56.41 |
| 9/27/2019 | 55.91 |
| 9/30/2019 | 54.07 |
| 10/1/2019 | 53.62 |
| 10/2/2019 | 52.64 |
| 10/3/2019 | 52.45 |
| 10/4/2019 | 52.81 |
| 10/7/2019 | 52.75 |
| 10/8/2019 | 52.63 |
| 10/9/2019 | 52.59 |
| 10/10/2019 | 53.55 |
| 10/11/2019 | 54.7 |
| 10/14/2019 | 53.59 |
| 10/15/2019 | 52.81 |
| 10/16/2019 | 53.36 |
| 10/17/2019 | 53.93 |
| 10/18/2019 | 53.78 |
| 10/21/2019 | 53.31 |
| 10/22/2019 | 54.16 |
| 10/23/2019 | 55.97 |
| 10/24/2019 | 56.23 |
| 10/25/2019 | 56.66 |
| 10/28/2019 | 55.81 |
| 10/29/2019 | 55.54 |
| 10/30/2019 | 55.06 |
| 10/31/2019 | 54.18 |
| 11/1/2019 | 56.2 |
| 11/4/2019 | 56.54 |
| 11/5/2019 | 57.23 |
| 11/6/2019 | 56.35 |
| 11/7/2019 | 57.15 |
| 11/8/2019 | 57.24 |
| 11/11/2019 | 56.86 |
| 11/12/2019 | 56.8 |
| 11/13/2019 | 57.12 |
| 11/14/2019 | 56.77 |
| 11/15/2019 | 57.72 |
| 11/18/2019 | 57.05 |
| 11/19/2019 | 55.21 |
| 11/20/2019 | 57.11 |
| 11/21/2019 | 58.58 |
| 11/22/2019 | 57.77 |
| 11/25/2019 | 58.01 |
| 11/26/2019 | 58.41 |
| 11/27/2019 | 58.11 |
| 12/2/2019 | 55.96 |
| 12/3/2019 | 56.1 |
| 12/4/2019 | 58.43 |
| 12/5/2019 | 58.43 |
| 12/6/2019 | 59.2 |
| 12/9/2019 | 59.02 |
| 12/10/2019 | 59.24 |
| 12/11/2019 | 58.76 |
| 12/12/2019 | 59.18 |
| 12/13/2019 | 60.07 |
| 12/16/2019 | 60.21 |
| 12/17/2019 | 60.94 |
| 12/18/2019 | 60.93 |
| 12/19/2019 | 61.22 |
| 12/20/2019 | 60.44 |
| 12/23/2019 | 60.52 |
| 12/26/2019 | 61.68 |
| 12/27/2019 | 61.72 |
| 12/30/2019 | 61.68 |
| 12/31/2019 | 61.06 |
| 1/2/2020 | 61.18 |
| 1/3/2020 | 63.05 |
| 1/6/2020 | 63.27 |
| 1/7/2020 | 62.7 |
| 1/8/2020 | 59.61 |
| 1/9/2020 | 59.56 |
| 1/10/2020 | 59.04 |
| 1/13/2020 | 58.08 |
| 1/14/2020 | 58.23 |
| 1/15/2020 | 57.81 |
| 1/16/2020 | 58.52 |
| 1/17/2020 | 58.54 |
| 1/21/2020 | 58.34 |
| 1/22/2020 | 56.74 |
| 1/23/2020 | 55.59 |
| 1/24/2020 | 54.19 |
| 1/27/2020 | 53.14 |
| 1/28/2020 | 53.48 |
| 1/29/2020 | 53.33 |
| 1/30/2020 | 52.14 |
| 1/31/2020 | 51.56 |
| 2/3/2020 | 50.11 |
| 2/4/2020 | 49.61 |
| 2/5/2020 | 50.75 |
| 2/6/2020 | 50.95 |
| 2/7/2020 | 50.32 |
| 2/10/2020 | 49.57 |
| 2/11/2020 | 49.94 |
| 2/12/2020 | 51.17 |
| 2/13/2020 | 51.42 |
| 2/14/2020 | 52.05 |
| 2/18/2020 | 52.05 |
| 2/19/2020 | 53.29 |
| 2/20/2020 | 53.78 |
| 2/21/2020 | 53.38 |
| 2/24/2020 | 51.43 |
| 2/25/2020 | 49.9 |
| 2/26/2020 | 48.73 |
| 2/27/2020 | 47.09 |
| 2/28/2020 | 44.76 |
| 3/2/2020 | 46.75 |
| 3/3/2020 | 47.18 |
| 3/4/2020 | 46.78 |
| 3/5/2020 | 45.9 |
| 3/6/2020 | 41.28 |
| 3/9/2020 | 31.13 |
| 3/10/2020 | 34.36 |
| 3/11/2020 | 32.98 |
| 3/12/2020 | 31.5 |
| 3/13/2020 | 31.73 |
| 3/16/2020 | 28.7 |
| 3/17/2020 | 26.95 |
| 3/18/2020 | 20.37 |
| 3/19/2020 | 25.22 |
| 3/20/2020 | 22.43 |
| 3/23/2020 | 23.36 |
| 3/24/2020 | 24.01 |
| 3/25/2020 | 24.49 |
| 3/26/2020 | 22.6 |
| 3/27/2020 | 21.51 |
| 3/30/2020 | 20.09 |
| 3/31/2020 | 20.48 |
| 4/1/2020 | 20.31 |
| 4/2/2020 | 25.32 |
| 4/3/2020 | 28.34 |
| 4/6/2020 | 26.08 |
| 4/7/2020 | 23.63 |
| 4/8/2020 | 25.09 |
| 4/9/2020 | 22.76 |
| 4/13/2020 | 22.41 |
| 4/14/2020 | 20.11 |
| 4/15/2020 | 19.87 |
| 4/16/2020 | 19.87 |
| 4/17/2020 | 18.27 |
| 4/21/2020 | 10.01 |
| 4/22/2020 | 13.78 |
| 4/23/2020 | 16.5 |
| 4/24/2020 | 16.94 |
| 4/27/2020 | 12.78 |
| 4/28/2020 | 12.34 |
| 4/29/2020 | 15.06 |
| 4/30/2020 | 18.84 |
| 5/1/2020 | 19.78 |
| 5/4/2020 | 20.39 |
| 5/5/2020 | 24.56 |
| 5/6/2020 | 23.99 |
| 5/7/2020 | 23.55 |
| 5/8/2020 | 24.74 |
| 5/11/2020 | 24.14 |
| 5/12/2020 | 25.78 |
| 5/13/2020 | 25.29 |
| 5/14/2020 | 27.56 |
| 5/15/2020 | 29.43 |
| 5/18/2020 | 31.82 |
| 5/19/2020 | 32.5 |
| 5/20/2020 | 33.49 |
| 5/21/2020 | 33.92 |
| 5/22/2020 | 33.25 |
| 5/26/2020 | 34.35 |
| 5/27/2020 | 32.81 |
| 5/28/2020 | 33.71 |
| 5/29/2020 | 35.49 |
| 6/1/2020 | 35.44 |
| 6/2/2020 | 36.81 |
| 6/3/2020 | 37.29 |
| 6/4/2020 | 37.41 |
| 6/5/2020 | 39.55 |
| 6/8/2020 | 38.19 |
| 6/9/2020 | 38.94 |
| 6/10/2020 | 39.6 |
| 6/11/2020 | 36.34 |
| 6/12/2020 | 36.26 |
| 6/15/2020 | 37.12 |
| 6/16/2020 | 38.38 |
| 6/17/2020 | 37.96 |
| 6/18/2020 | 38.84 |
| 6/19/2020 | 39.75 |
| 6/22/2020 | 40.46 |
| 6/23/2020 | 40.37 |
| 6/24/2020 | 38.01 |
| 6/25/2020 | 38.72 |
| 6/26/2020 | 38.49 |
| 6/29/2020 | 39.7 |
| 6/30/2020 | 39.27 |
| 7/1/2020 | 39.82 |
| 7/2/2020 | 40.65 |
| 7/6/2020 | 40.63 |
| 7/7/2020 | 40.62 |
| 7/8/2020 | 40.9 |
| 7/9/2020 | 39.62 |
| 7/10/2020 | 40.55 |
| 7/13/2020 | 40.1 |
| 7/14/2020 | 40.29 |
| 7/15/2020 | 41.2 |
| 7/16/2020 | 40.75 |
| 7/17/2020 | 40.59 |
| 7/20/2020 | 40.81 |
| 7/21/2020 | 41.96 |
| 7/22/2020 | 41.9 |
| 7/23/2020 | 41.07 |
| 7/24/2020 | 41.29 |
| 7/27/2020 | 41.6 |
| 7/28/2020 | 41.04 |
| 7/29/2020 | 41.27 |
| 7/30/2020 | 39.92 |
| 7/31/2020 | 40.27 |
| 8/3/2020 | 41.01 |
| 8/4/2020 | 41.7 |
| 8/5/2020 | 42.19 |
| 8/6/2020 | 41.95 |
| 8/7/2020 | 41.22 |
| 8/10/2020 | 41.94 |
| 8/11/2020 | 41.61 |
| 8/12/2020 | 42.67 |
| 8/13/2020 | 42.24 |
| 8/14/2020 | 42.01 |
| 8/17/2020 | 42.89 |
| 8/18/2020 | 42.89 |
| 8/19/2020 | 42.93 |
| 8/20/2020 | 42.58 |
| 8/21/2020 | 42.34 |
| 8/24/2020 | 42.62 |
| 8/25/2020 | 43.35 |
| 8/26/2020 | 43.39 |
| 8/27/2020 | 43.04 |
| 8/28/2020 | 42.97 |
| 8/31/2020 | 42.61 |
| 9/1/2020 | 42.76 |
| 9/2/2020 | 41.51 |
| 9/3/2020 | 41.37 |
| 9/4/2020 | 39.77 |
| 9/8/2020 | 36.76 |
| 9/9/2020 | 38.05 |
| 9/10/2020 | 37.3 |
| 9/11/2020 | 37.33 |
| 9/14/2020 | 37.26 |
| 9/15/2020 | 38.28 |
| 9/16/2020 | 40.16 |
| 9/17/2020 | 40.97 |
| 9/18/2020 | 41.11 |
| 9/21/2020 | 39.31 |
| 9/22/2020 | 39.6 |
| 9/23/2020 | 39.93 |
| 9/24/2020 | 40.31 |
| 9/25/2020 | 40.25 |
| 9/28/2020 | 40.6 |
| 9/29/2020 | 39.29 |
| 9/30/2020 | 40.22 |
| 10/1/2020 | 38.72 |
| 10/2/2020 | 37.05 |
| 10/5/2020 | 39.22 |
| 10/6/2020 | 40.67 |
| 10/7/2020 | 39.95 |
| 10/8/2020 | 41.19 |
| 10/9/2020 | 40.6 |
| 10/12/2020 | 39.43 |
| 10/13/2020 | 40.2 |
| 10/14/2020 | 41.04 |
| 10/15/2020 | 40.96 |
| 10/16/2020 | 40.88 |
| 10/19/2020 | 40.83 |
| 10/20/2020 | 41.46 |
| 10/21/2020 | 40.03 |
| 10/22/2020 | 40.64 |
| 10/23/2020 | 39.85 |
| 10/26/2020 | 38.56 |
| 10/27/2020 | 39.57 |
| 10/28/2020 | 37.39 |
| 10/29/2020 | 36.17 |
| 10/30/2020 | 35.79 |
| 11/2/2020 | 36.81 |
| 11/3/2020 | 37.66 |
| 11/4/2020 | 39.15 |
| 11/5/2020 | 38.79 |
| 11/6/2020 | 37.14 |
| 11/9/2020 | 40.29 |
| 11/10/2020 | 41.36 |
| 11/11/2020 | 41.45 |
| 11/12/2020 | 41.12 |
| 11/13/2020 | 40.13 |
| 11/16/2020 | 41.34 |
| 11/17/2020 | 41.43 |
| 11/18/2020 | 41.82 |
| 11/19/2020 | 41.74 |
| 11/20/2020 | 42.15 |
| 11/23/2020 | 43.06 |
| 11/24/2020 | 44.91 |
| 11/25/2020 | 45.71 |
| 11/30/2020 | 45.34 |
| 12/1/2020 | 44.55 |
| 12/2/2020 | 45.28 |
| 12/3/2020 | 45.64 |
| 12/4/2020 | 46.26 |
| 12/7/2020 | 45.76 |
| 12/8/2020 | 45.6 |
| 12/9/2020 | 45.52 |
| 12/10/2020 | 46.78 |
| 12/11/2020 | 46.57 |
| 12/14/2020 | 46.99 |
| 12/15/2020 | 47.62 |
| 12/16/2020 | 47.82 |
| 12/17/2020 | 48.36 |
| 12/18/2020 | 49.1 |
| 12/21/2020 | 47.74 |
| 12/22/2020 | 47.02 |
| 12/23/2020 | 48.12 |
| 12/28/2020 | 47.62 |
| 12/29/2020 | 48 |
| 12/30/2020 | 48.4 |
| 12/31/2020 | 48.52 |
| 1/4/2021 | 47.62 |
| 1/5/2021 | 49.93 |
| 1/6/2021 | 50.63 |
| 1/7/2021 | 50.83 |
| 1/8/2021 | 52.24 |
| 1/11/2021 | 52.25 |
| 1/12/2021 | 53.21 |
| 1/13/2021 | 52.91 |
| 1/14/2021 | 53.57 |
| 1/15/2021 | 52.36 |
| 1/19/2021 | 52.98 |
| 1/20/2021 | 53.24 |
| 1/21/2021 | 53.13 |
| 1/22/2021 | 52.27 |
| 1/25/2021 | 52.77 |
| 1/26/2021 | 52.61 |
| 1/27/2021 | 52.85 |
| 1/28/2021 | 52.34 |
| 1/29/2021 | 52.2 |
| 2/1/2021 | 53.55 |
| 2/2/2021 | 54.76 |
| 2/3/2021 | 55.69 |
| 2/4/2021 | 56.23 |
| 2/5/2021 | 56.85 |
| 2/8/2021 | 57.97 |
| 2/9/2021 | 58.36 |
| 2/10/2021 | 58.68 |
| 2/11/2021 | 58.24 |
| 2/12/2021 | 59.47 |
| 2/16/2021 | 60.05 |
| 2/17/2021 | 61.14 |
| 2/18/2021 | 60.52 |
| 2/19/2021 | 59.24 |
| 2/22/2021 | 61.49 |
| 2/23/2021 | 61.67 |
| 2/24/2021 | 63.22 |
| 2/25/2021 | 63.53 |
| 2/26/2021 | 61.5 |
| 3/1/2021 | 60.64 |
| 3/2/2021 | 59.75 |
| 3/3/2021 | 61.28 |
| 3/4/2021 | 63.83 |
| 3/5/2021 | 66.09 |
| 3/8/2021 | 65.05 |
| 3/9/2021 | 64.01 |
| 3/10/2021 | 64.44 |
| 3/11/2021 | 66.02 |
| 3/12/2021 | 65.61 |
| 3/15/2021 | 65.39 |
| 3/16/2021 | 64.8 |
| 3/17/2021 | 64.6 |
| 3/18/2021 | 60 |
| 3/19/2021 | 61.42 |
| 3/22/2021 | 61.55 |
| 3/23/2021 | 57.76 |
| 3/24/2021 | 61.18 |
| 3/25/2021 | 58.56 |
| 3/26/2021 | 60.97 |
| 3/29/2021 | 61.56 |
| 3/30/2021 | 60.55 |
| 3/31/2021 | 59.16 |