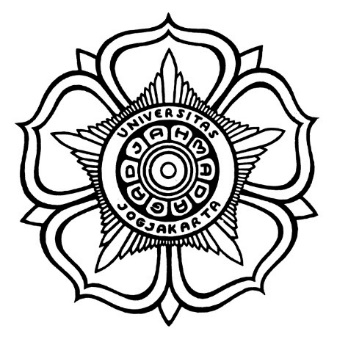
skripsi

*Unscented Kalman Filter* dan *recurrent Neural network* pada peramalan

*Unscented Kalman Filter* and *recurrent Neural network* for forcasting



ruli sastra putri

14/364140/pa/15914

PROGRAM STUDI ilmu komputer

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

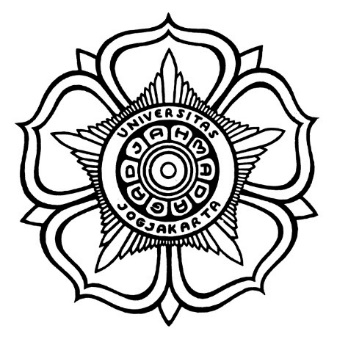
YOGYAKARTA

skripsi

*Unscented Kalman Filter* dan *recurrent Neural network* pada peramalan

*Unscented Kalman Filter* and *recurrent Neural network* for forcasting

Diajukan untuk memenuhi salah satu syarat memperoleh derajat



ruli sastra putri

14/364140/pa/15914

PROGRAM STUDI ilmu komputer

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2019

HALAMAN PENGESAHAN

SKRIPSI

*Unscented Kalman Filter* dan *recurrent Neural network* pada peramalan

Telah dipersiapkan dan disusun oleh

RULI SASTRA PUTRI

14/364140/PA/15914

Telah dipertahankan di depan Tim Penguji

pada tanggal 28 Agustus 2019

Susunan Tim Penguji

|  |  |
| --- | --- |
|  | <Masukan nama penguji> |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | <Masukan nama Penguji> |
|  |  |
|  |  |
|  |  |
|  |  |
|  | <Masukan nama Penguji> |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta,

Ruli Sastra Putri

*Karya ini penulis persembahkan untuk orang tua, kakak-kakak, adik-adik, dan seluruh keluarga besar yang selalu memberikan harapan bagi penulis, juga segenap rekan-rekan serta pembaca sekalian.*

Katakanlah: “Kalau sekiranya lautan menjadi tinta untuk (menulis) kalimat-kalimat Tuhanku, sungguh habislah lautan itu sebelum habis (ditulis) kalimat kalimat Tuhanku, meskipun Kami datangkan tambahan sebanyak itu (pula)”.

(Q.S. Al-Kahfi: 109)

PRAKATA

Puji syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik.

Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapan terima kasih kepada:

1. Allah SWT dan Baginda Rasulullah SAW atas nikmat dan karunia yang telah diberikan pada penulis selama di dunia dan akhirat nanti
2. Prof. Dr. Subanar selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
3. Prof. Dr. Dra. Sri Wahyuni, M.S. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
4. Drs. G.P. Dalijo. Dipl.Comp. selaku Ketua Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
5. Drs. Sri Mulyana, M.Kom selaku Dosen Wali Akademik penulis.
6. Drs. Yohanes Suyanto, M.Kom selaku Dosen Pembimbing yang telah berkenan meluangkan waktu, dan pikiran dalam penyelesaian tugas akhir ini.
7. Nur Rokhman, S.Si, M.Kom dan Aina Musdholifah, S.Si selaku tim penguji.
8. Dr. Pekik Purwantoro, M.Si yang telah menyumbangkan template naskah skripsi FMIPA UGM berbasis LaTeX dan OpenOffice untuk menyusun naskah skripsi ini.
9. Segenap Dosen dan civitas akademik di lingkungan Program Studi Ilmu Komputer, Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.
10. Ibu dan Bapak yang dengan sabarnya menunggu, membimbing, memfasilitasi, dan memberikan kasih saying selama penulis menempuh pendidikan. Tidak lain keja keras ini hanya untuk membahagiakan kalian.
11. Ma Sist yang selalu ada saat ingin bercerita dengan motivasi dan kegigihannya menjadi tolak ukur kami adiknya, serta Kak Hendra kemudian Gendis yang dengan cerdasnya datang kedunia dan menceriakan Palembang dengan pipinya.
12. Mas Kibut dan Yuk Milda yang selalu ada untuk memberikan kepercayaan dan peluang untuk lebih mengeksplore diri. Terima kasih sudah melahirkan Babang yang menjadi sumber energi baru bagi penulis.
13. Saudari saudari dari Indoktang yang dengan teguh, strong menghadapi dunia beserta anak-anaknya yang dengan cerdas menjadi tumpuan pembelajaran baru bagi penulis.
14. Teman-teman kuliah seluruh angkatan yang telah memberikan dukungan tiada habisnya. Menjadi tolak ukur keberhasilan yang kelihatannya tidak berusaha, padahal berjuang mati-matian.
15. Wiwik dan Dyah. Kalianlah yang menjadi pelita yang selalu ada. You are the only one.
16. Teman KKN Kemadang 2018 yang dengan komitmenya menjaga silaturahmi antar sesame.
17. Teman HIMAKOM UGM yang menjadi bagian manis dan pahit dalam berorganisasi.
18. Pihak-pihak lain yang tidak dapat disebutkan satu per satu.
19. Yang terakhir… Kamu. Terima kasih telah menjadi support utama di masa labil penulis. Kenangan manis terukir jelas di seluruh Jogja.

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, terutama bagi perkembangan ilmu pengetahuan serta perkembangan Ilmu Komputer dan Teknologi Informasi.

Yogyakarta, 28 Agustus 2019

Penulis

DAFTAR ISI

PRAKATA iv

INTISARI xi

ABSTRACT xii

BAB I  
PENDAHULUAN 13

1.1 Latar Belakang 13

1.2 Rumusan Masalah 14

1.3 Batasan Masalah 15

1.4 Tujuan Penelitian 15

1.5 Manfaat Penelitian 15

BAB II  
TINJAUAN PUSTAKA 17

BAB III  
LANDASAN TEORI 21

3.1 Bitcoin 21

3.2 Prediksi Data Time Series 21

3.3 Neural Network 22

3.4 Kalman Filter 29

3.5 Normalisasi 32

3.6 Sliding Window 33

3.7 Evaluasi 34

BAB IV  
ANALISIS DAN RANCANGAN SISTEM 37

4.1 Analilis Permasalahan 37

4.2 Rancangan Umum Sistem 37

4.3 Data 38

4.4 Normalisasi Data 39

4.5 Arsitektur NN 40

4.6 Unscented Kalman Filter 42

4.7 Prosedur Pelatihan 45

4.8 Rancangan Pengujian 47

BAB V  
IMPLEMENTASI 48

5.1 Lingkungan Implementasi 48

5.2 Data 48

5.3 Normalisasi Data 48

5.4 Pemisahan Data 49

5.5 Menentukan I/O JST 50

5.6 Implementasi RNN 52

5.7 Pengujian 53

BAB VI  
HASIL PENELITIAN DAN PEMBAHASAN 54

6.1 Pengujian Arsitektur JST 55

6.2 Pengujian Ukuran Sliding Window 55

6.3 Pengujian Jumlah Neuron 55

BAB VII  
KESIMPULAN 56

7.1 Kesimpulan 56

7.2 Saran 56

DAFTAR PUSTAKA 57

LAMPIRAN A

SOURCE CODE 60

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian 17

DAFTAR GAMBAR

Gambar Error! No text of specified style in document..1 Jaringan saraf biologis (Fausett, 1994) 20

Gambar Error! No text of specified style in document..2 *Multilayer perceptron* (Fausett, 1994) 21

Gambar 3.3 Grafik fungsi identitas (Fausett, 1994) 23

Gambar 3.4 Grafik fungsi undak biner (Fausett, 1994) 23

INTISARI

*Unscented Kalman Filter* dan *recurrent Neural network* pada peramalan

Ruli Sastra Putri

14/364140/PA/15914

Contoh untuk Intisari. Pada umumnya sistem perangkat lunak terdiri dari beberapa *concern*, premis dari masalah ini adalah sebaran *concern*, di mana kebutuhan rancangan tertentu cenderung memotong-melintasi grup inti fungsional modul. Teknik orientasi-objek yang menerapkan *concern* tersebut cenderung menghasilkan kode yang tersebar, daya baca yang sulit, serta susah untuk dikembangkan. Metodologi baru, *aspect-oriented programming* (AOP), memberikan fasilitas modularisasi pemotong-lintasan/*cross-cutting concern*. Dengan menggunakan AOP, terdapat cara untuk membuat penerapan sistem yang lebih mudah untuk dirancang, dipahami, dan dipelihara. Lebih jauh lagi, AOP menjanjikan produktivitas yang lebih tinggi, peningkatan kualitas, dan kemampuan lebih baik untuk menambahkan *feature* baru.

Contoh paragraf kedua. AspectJ adalah bahasa pemrograman yang digunakan secara luas untuk menerapkan program-program berorientasi aspek di Java. Namun demikian, AspectJ masih belum memiliki bahasa pemodelan yang dapat memenuhi perancangan program berorientasi aspek. Aspect Oriented Design Model (AODM), sebagai sebuah model perancangan baru pada pengembangan program dalam AspectJ, hanya memperluas konsep-konsep UML (Unified Modeling Language) yang telah ada dengan menggunakan mekanisme perluasan UML untuk memberikan konsep orientasi-aspek yang ada di dalam AspectJ. AODM menyediakan spesikasi model rancangan orientasi-aspek untuk ditransformasikan menjadi model rancangan UML biasa.

ABSTRACT

*Unscented Kalman Filter* and *recurrent Neural network* for forcasting

Ruli Sastra Putri

14/364140/PA/15914

Example of Abstract. Most software systems consist several concerns, the premise of such thing is separation of concerns, where certain design requirements tend to cut across group of core functional modules. Object-oriented techniques for implementing such concerns result in systems that are invasive to implement, tough to understand, and difficult to evolve. The new aspect-oriented programming (AOP) methodology facilities modularization of crosscutting concerns. Using AOP, there is a way to create implementations that are easier to design, understand, and maintain. Further, AOP promises higher productivity, improved quality, and better ability to implement newer features.

Second paragraf. AspectJ is a well-established programming language that is widely used to implement aspect-oriented programs in Java. However, there is no modeling language available for the design aspect oriented programs in AspectJ. Aspect Oriented Design Model (AODM), as a new design model for development of AspectJ programs, extends existing UML (Unified Modeling Language) concepts using standard UML extension mechanisms to provide aspect-oriented concepts as in AspectJ. The AODM species how an aspect-oriented design model maybe transformed into an ordinary UML design model.

1. BAB I  
   PENDAHULUAN
   1. Latar Belakang

*Cryptocurrency* merupakan alat transaksi online alternatif yang fenomenal, bersifat desentralisasi untuk mengurangi campur tangan dari pemerintah suatu negara (sistem moneter). *Cryptocurrency* tidak memiiki bentuk fisik, dan diperoleh dengan “mining” oleh beberapa pengguna. Besar, bebasnya penggunaan dan faktornya yang tidak dapat didefinisikan secara pasti. Hal ini menyebabkan flutktuasi harga yang berakibat pada ketidakstabilan. Ketidakstabilan ini juga merupakan salah satu faktor mengapa beberapa negara melarang penggunaaannya. *Cryptocurrency* tidak dapat dinyatakan sebagai mata uang di Indonesia karena tidak memiliki karakteristik sebagai mata uang dibawah UU No. 7 Tahun 2011 (Kenawas & Sulistiowati, 2016). Walaupun kendala legalitas *cryptocurrency* masih dipertanyakan secara hukum di beberapa negara, sampai saat ini *cryptocurrency* masih dimiliki dan digunakan oleh banyak orang.

Bitcoin diklasifikasikan sebagai koin yang paling terkenal dan merupakan salah satu yang tertua dari teknologi *cryptocurrency*. Bitcoin ditemukan pada tahun 2008 oleh Satoshi Nakamoto (Nakamoto, 2009) yang sampai sekarang tidak diketahui identitas aslinya. Pada tahun 2013, Bitcoin sangatlah fenomenal. Pada awal tahun 2013, harga per-Bitcoin adalah $13 dan melambung menjadi $230 pada 9 April 2013 dengan potensial keuntungan mencapai hampir 1700% dalam kurun waktu kurang dari 4 bulan. Pada tahun yang sama, harganya meroket menjadi $395 pada 9 November 2013 dengan keuntungan mencapai 2900% sejak awal tahun 2013 (Kristoufek, 2013).

Pada awal bulan Mei 2018, harga Bitcoin turun hingga $9,674. Fluktuasi yang tinggi tersebut menjadi masalah di dalam dunia *cryptocurrency.* Pemain *cryptocurrency* memiliki satu tujuan, yaitu mendapatkan keuntungan maksimal dengan kerugian seminimal mungkin. Oleh karena itu, permasalahan tersebut menarik banyak peneliti untuk mencoba memprediksi harga menggunakan banyak metode. Akan tetapi, kurangnya referensi sebagai dasar penelitian dan hasil akurasi yang tidak terlalu tinggi dibandingkan dengan data saham menjadi kendala untuk peneliti *cryptocurrency*.

Sebagai upaya peningkatan akurasi, beberapa penelitian lainnya berusaha mencari korelasi dan hubungan kasualitas antara Bitcoin dan faktor-faktor memungkinkan lainnya untuk digunakan sebagai variabel bebas sebagai salah satu upaya peningkatan hasil akurasi*.* Terdapat alah satu penelitian (Kim, et al., 2016) dimana penelitian tersebut menggunakan komentar pengguna dan Twitter sebagai variabel bebas dengan menggunakan algoritma *Machine Learning*. Twitter dan komentar pengguna pada forum setiap koin digunakan karena banyak penelitian sebelumnya yang menganalisis korelasi dan hubungan kasualitas *cryptocurrency* dengan sentimen pengguna media sosial (Twitter) dan perhitungan kuantitatif pencarian *queries* pada mesin pencari (Google). Penelitian tersebut menyatakan bahwa *cryptocurrency* dilakukan secara online dalam skala besar dimana banyak pemain mengambil keputusan *hold*, *sell* dan *buy* bergantung dari informasi dari website*.* Pada hasil penelitian tersebut, akurasi yang didapatkan adalah sebesar ±70%. Peneitian (Martina, et al., 2015) lain menyatakanbahwa Twitter sebagai media jaringan sosial online dan *microblogging* telah menjadi alat yang penting untuk bisnis, komunikasi serta membagikan informasi dengan pertumbuhan yang cepat dan signifikan. Pada akhir penelitian, hasil yang didapatkan adalah *tweets* berpengaruh besar terhadap harga Bitcoin dalam beberapa hari dan Google Trends yang mampu untuk memprediksi Bitcoin dikarenakan nilai *cross-correlation* yang tinggi dan 0 *delay.*

Selain faktor, pemilihan model arsitektur sistem dan algoritma menjadi krusial dalam akurasi prediksi Bitcoin.

* 1. Rumusan Masalah

Fluktuasi *cryptocurrency* yang tidak stabil dikarenakan sifatnya yang desentralisasi menjadi salah satu masalah pada predksi harga Bitcoin. Sedikitnya referensi peneliti *cryptocurrency* dan hasil akurasi prediksi yang tidak terlalu tinggi dibandingkan penelitian menggunakan data saham, menjadi masalah lainnya. Upaya untuk mengatasi masalah ini adalah dengan menggunakan metode dan rancangan model prediksi lainnya yang memungkinkan untuk meningkatkan hasil akurasi prediksi.

Berdasarkan penelitian-penelitian sebelumnya, salah satu metode yang dapat digunakan adalah *Unscented Kalman Filter* dan *Neural Network.* Keduanya telahterbukti dapat menjadi metode untuk prediksi harga mata uang. Oleh karena itu, pada penelitian ini, dilakukan prediksi harga Bitcoin menggunakan metode *Unscented Kalman Filter Neural Network*.

* 1. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Dataset harga Bitcoin diambil dari blockchain.info.
2. Penelitian ini mengggunakan metode *Unscented Kalman Filter* dan *Neural Network.*
3. Data set Bitcoin merupakan harga BTC/USD diambil mulai dari 1 Januari 2011 – 31 Mei 2018 sebanyak 2359 data.
   1. Tujuan Penelitian

Tujuan penelitian ini melakukan prediksi harga Bitcoin menggunakan model *Neural Network* dan persamaan *Unscented Kalman Filter* sebagai metode prediksi. Penelitian ini bertujuan agar pihak-pihak yang terpengaruh atas berubahnya harga Bitcoin dapat memanfaatkannya dan melakukan prediksi. Dengan prediksi tersebut, diharapkan dapat diambil keputusan terbaik untuk mendapatkan keuntungan maksimal dan menghindari serta meminimalisasi terjadinya kerugian.

* 1. Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini, antara lain:

1. Menambah pengetahuan dan wawasan tentang model untuk memprediksi data *time series* harga Bitcoin.
2. Sebagai referensi peneliti lainnya di ruang lingkup masalah yang berhubungan agar menghasilkan metode prediksi yang lebih baik.
3. Dapat digunakan oleh pemain *cryptocurrency* (Bitcoin) dan pihak-pihak lainnnya yang terpengaruh perubahan harga untuk dapat memanfaatkannya untuk melakukan prediksi harga.
4. Sebagai pendukung peneliti lainnya untuk mengembangkan sistem *real-time* yang lebih mudah digunakan oleh pemain lainnya yang bukan berasal dari ranah teknologi.
5. BAB II  
   TINJAUAN PUSTAKA

Berdasarkan penelitian (Kim, et al., 2016), terdapat korelasi dan hubungan kasualitas antara komentar pengguna dan harga serta volume Bitcoin, Ethereum, dan Ripple. Pada penelitian ini, Kim dkk menggunakan metode *Machine Learning* menggunakan persamaan *Average One Dependence Estimator* sebagai validasi model yang digunakan. Komentar pengguna pada setiap forum koin dilakukan analisis sentiment secara manual. Hasilnya, metode tersebut berhasil digunakan dan didapatkan nilai akurasi ±70% untuk semua koin. Besarnya forum dan banyaknya data yang diolah juga mempengaruhi akurasi.

Metode peramalan juga dilakukan pada penelitian lain. Penelitian (Bisoi & Dash, 2014) memprediksi data trend saham menggunakan *Unscented Kalman Filter,* *Differential Evolution* dan DEUKF. Hasilnya didapatkan bahwa model *hybrid* DEUKF menghasilkan nilai MAPE lebih sedikit dibandingkan dengan UKF atau DE sendiri. Metode *hybrid* ini menghasilkan MAPE3.13 dengan 400 data IBM. Sedangkan UKF dan DE sebesar 4.62 dan 3.96.

Penelitian (Amjad & Shah, 2016) tentang perbandingan algoritma untuk prediksi Bitcoin menyatakan bahwa model ARIMA sangat buruk untuk hasil prediksi dan probabilitasnya. Penelitian tersebut mengunakan data dari 2014-2016 dengan perbandingan algoritma antara model *Empirical Conditional,* ARIMA, *Random Forest, Linear Descriminant Analysis* dan *Logistic Regression*. Hasil yang didapatkan algoritma RF (algoritma klasifikasi)menunjukkan performa yang lebih baik dengan akurasi (>60-70%) dan *Sharpe Ratio* (> 2.0)*.*

Prediksi Bitcoin (Albariqi, 2018) menggunakan *Multilayer Perceptron* dan RNN, menghasilkan bahwa MLP mengungguli RNN dengan *accuracy* 81,3%, *precision* 81% dan *recall* 94,7%. Data yang digunakan adalah data *Blockchain* pada tahun 2010-2017 dengan periode 2 hari. Sedangkan pada penelitian lain dilakukan prediksi mata uang USD menggunakan RNN dan *Extended Kalman Filter* (Hazazi, 2018)*.* Hasilnya didapatkan *Dstat* terbaik sebesar 73,06%, MSE 2425,64, RMSE 49,25 dan MAE 35,05. Data yang digunakan adalaj 70% data latih dan 30% data uji dengan model arsitektur 3-6-1.

Berdasarkan penelitian (Torres & Qiu, 2018), prediksi harga Bitcoin menggunakanmodel LSTM dan GRU menghasilkan RSME LSTM dan GRU sebesar 272,96 dan 274,02 dimana hasil tidak terlalu kentara. Tetapi pada performa jaringan, GRU lebih unggul 12% dibandingkan LSTM dikarenakan sistem GRU yang lebih sederhana. Namun pada penelitian yang dilakukan (Saptoro, 2012), digunakan model gabungan *Feedforward Neural Networks* dengan EKF dan UKF. Hasilnya, didapatkan nilai MAE pada UKF lebih unggul dengan peningkatan 2..45-21.48% (data pembelajaran)dan 8.35-29.15% (data uji) dibandingkan dengan EKF.

Salah satu penelitian menunjukkan bahwa EKF memiliki beberapa kekurangan dan melakukan penelitian pada metode alternative, yaitu UKF. Penelitian ini memodifikasi mean dan kovarian pada algoritma EKF. Hasil yang didapatkan adalah jelas, bahwa UKF menunjukkan kompleksitas dan performance yang terus menerus mengungguli EKF (Wan & Merwe, 2002). Sedangkan pada penelitian (Zhan & Wan, 2006), dilakukan perbandingan algoritma EKF, UKF, Particle Filter menggunakan arsitektur NN. Hasilnya, didapatkan bahwa metode gabungan PF-UKF menunjukkan akurasi dengan hanya 4% waktu komputasi yang membuat metode tersebut lebih cocok untuk dilakukan prediksi *real-time.*

Penelitian yang dilakukan pada penelitian ini menggunakan data *time series* harga Bitcoin sebagai *neuron input* model *Neural Network* dan UKF untuk pembobotan sebagai arsitektur model. Kemudian data di*preprocessing* dan normalisasi terlebih dahulu. Evalusi metode dilakukan penghitungan *Dstat,* RMSE, MAE, dan MAPE dalam mengetahui akurasi metode rancangan. Untuk perbandingan penelitian yang sudah ada dirangkum dalam Tabel 2.1

Tabel 2.1 Perbandingan Penelitian

| **No** | **Peneliti** | **Topik** | **Metode** | **Perbedaan** |
| --- | --- | --- | --- | --- |
| 1. | (Kim, et al., 2016) | *Predicting Fluctuations in Cryptocurrency Transaction Based on User Comments and Replies* | *Machine Learning* (persamaan *Average One Dependence Estimator*) | -Metode  -Data Bitcoin, Ethereum, dan Ripple  -Data dari forum web dan Twitter |
| 2. | (Bisoi & Dash, 2014) | *A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter* | **-**UKF  -DE (Differential Evolution)  -DEUKF (hybrid) | **-**Metode  -Data (Data trend saham) |
| 3. | (Amjad & Shah, 2016) | *Trading Bitcoin and Online Time Series Prediction* | -ARIMA  -*Random Forest*  -*Logistic Regression*  -*Linear Discriminant Analysis*  -*Empirical Conditional* | -Perbandingan metode |

**Tabel 2.1 Perbandingan penelitian (lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Peneliti** | **Topik** | **Metode** | **Perbedaan** |
| 4. | (Albariqi, 2018) | *Prediksi Perubahan Harga Bitcoin Menggunakan Jaringan Syaraf Tiruan* | -*Multilayer Perceptron*  *-Recurrent Neural Networks.* | -Metode  -Data |
| 5. | (Saptoro, 2012) | *Extended and unscented kalman filters for artificial neural network modelling of a nonlinear dynamical system* | **-***Extended Kalman Filters*  *-Unscented Kalman Filters*  *-Feedforward Neural Networks* | -Metode  -Data  -Perbandingan |
| 6. | (Hazazi, 2018) | *Reccurent Neural Network dan Extended Kalman Filter untuk Peramalan Nilai Tukar Mata Uang* | -RNN  -EKF | -Metode  -Data |
| 7. | (Wan & Merwe, 2002) | *The Unscented Kalman Filter For Nonlinear Estimation* | -EKF  -UKF | -Data  -Arsitektur |
| 8. | (Zhan & Wan, 2006) | *Neural Network-Aided Adaptive Unscented Kalman Filter For Nonlinear State Estimation* | 1. NN-UKF 2. Particle Filter 3. PF-EKF 4. PF-UKF | -Data  -Metode |

1. BAB III  
   LANDASAN TEORI
   1. Bitcoin

Mata uang digital yang menggantungkan keamanannya dengan kriptografi. Bitcoin bersifat *Peer-to-peer* (P2P) *networking* atau *computing* merupakan aplikasi arsitektur sistem terdistribusi yang membagi-bagi pekerjaan ke setiap titik. Setiap node berfungsi baik sebagai penyedia maupun pengguna layanan. Berbeda dengan sistem *client-server*, *peer-to-peer* bersifat desentralisasi karena setiap titik memiliki hak yang serupa. Bitcoin juga menggunakan *hashcash* *proof-of-work* untuk keamanannya dalam bertransaksi. *Hashcash* adalah jenis *proof-of-work* yang selain diimplementasikan di Bitcoin, diaplikasikan juga untuk menyaring surel yang datang ataupun pesan yang ditujukan ke alamat IP (Internet Protocol). Metode ini dilakukan dengan menambahkan teks pada header pesan yang sudah memiliki bentuk, agar saat dilakukan fungsi *hash*, dapat dihasilkan *message digest* yang sesuai dengan kriteria; Bitcoin menggunakan SHA256 sebagai fungsi hash-nya (Mangan, 2013)

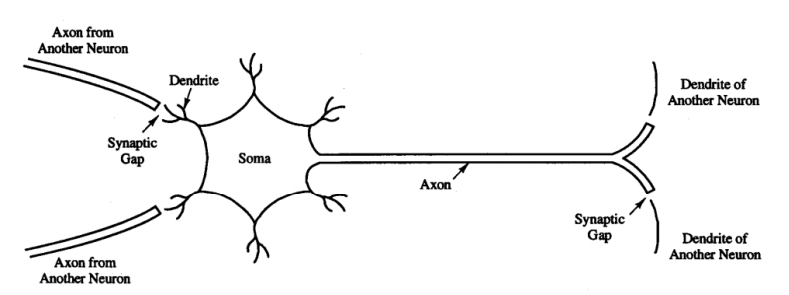
*Proof-of-work* adalah sebuah fungsi atau protokol yang diharapkan mampu mengagalkan *denial of service* ataupun berbagai penggunaan jasa berlebihan seperti *spam* dengan menuntut sebuah pekerjaan dilakukan oleh pengguna/pengaju jasa sebelum menggunakan jasa tersebut, biasanya mengakibatkan waktu proses saat dilakukan komputer. Kunci dari fungsi ini adalah asimetri; pekerjaan tersebut haruslah sulit (tetapi dapat dilakukan) dari pihak pengaju tetapi mudah diperiksa oleh pihak pemberi jasa. *Proof-of-work* berbeda dengan CAPTCHA, yang dimaksudkan untuk diselesaikan oleh manusia (Mangan, 2013).

* 1. Prediksi Data *Time Series*

Data *time series* merupakan nilai-nilai yang berurutan dan biasanya mempunyai atribut waktu. Data *time series* dapat ditemukan di berbagai ranah, contohnya keuangan, energi, klimatologi, dan meteorologi. Terdapat dua macam pengolahan untuk data *time series*, yaitu *time series analysis* dan *time series forecasting*. Analisis data *time series* bertujuan untuk menghitung sifat-sifat statistik data, sedangkan *time series forecasting* untuk mengetahui nilai pada masa mendatang (Gutierrez, et al., 2017)

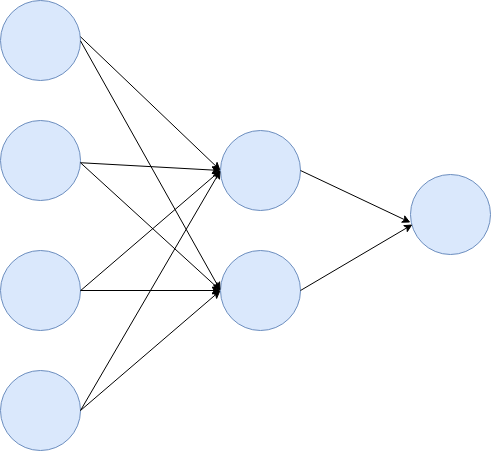
* 1. *Neural Network*

Jaringan saraf tiruan (JST) atau *Artificial Neural Network* (ANN) merupakan sistem pemrosesan informasi yang memiliki karakteristik seperti dengan jaringan saraf biologis (Fausett, 1994). NN dikembangkan sebagai bentuk generalisasi dari model matematika dari sistem pengetahuan manusia.



Gambar Error! No text of specified style in document..1 Jaringan saraf biologis (Fausett, 1994)

Pada jaringan saraf biologis seperti pada Gambar 3.1 neuron mendapat rangsang sebagai *input* ke dalam badan *neuron*, dan disampaikan ke *neuron* lain melalui akson dan dendrit. Pada NN, pola hubungan antar neuron disebit arsitektur jaringan. Sebagai contoh, diilustrasikan sebuah *multilayer perceptron* dengan 3 *neuron* pada lapisan *input*, 2 *neuron* dalam lapisan tersembunyi, dan 1 *neuron* pada lapisan *output* pada Gambar 3.2.



Gambar Error! No text of specified style in document..2 *Multilayer perceptron* (Fausett, 1994)

Lapisan-lapisan penyusun NN dibagi menjadi tiga (Siang, 2009, p. 24), yaitu;

1. Lapisan *input*

*Neuron* di dalam lapisan *input* disebut neuron *input*. Neuron *input* menerima *input* dari luar, *input* yang diberikan merupakan penggambaran suatu permasalahan.

1. Lapisan *hidden*

*Neuron* di dalam lapisan initersembunyi. Output dari lapisan ini tidak dapat diamati secara langsung.

1. Lapisan *output*

*Neuron* di dalam lapisan *output* disebut neuron *output*. Keluaran dari lapisan ini merupakan hasil dari *Neural Network* terhadap suatu permasalahan.

*Neuron-neuron* dalam NN berfungsi sebagai elemen pemroses yang berfungsi seperti halnya sebuah neuron pada hewan. Sejumlah sinyal *input* dikalikan dengan masing-masing bobot yang bersesuaian (penentuan nilai bobot disebut sebagai metode *training*). Kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan *output* yang dihasilkan digunakan untuk mendapatkan *output* jaringan.

Berdasarkan (Fausett, 1994, pp. 12-15), pelatihan NN ada 2 macam, yaitu pelatihan terbimbing (*supervised training*) dan pelatihan tidak terbimbing (*unsupervised training*).

Secara garis besar *supervised training* merupakan model pelatihan dengan dataset yang telah dilabel. Tipenya adalah klasifikasi dan regresi(misalnya prediksi trend menggunakan data yang telah dilabel sebelumnya). Contoh algoritmanya adalah *Back Propagation.*

*Unsupervised training* tidak diberikan pengetahuan sepanjang pelatihan. Tipenya adalah klasterisasi (menemukan pola dan pengelompokan dari data tanpa pelabelan). Selama proses pelatihan, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Terdapat vektor *input* namun tidak ada vektor target. NN melakukan pembaruan bobot sehingga *input* dengan karakteristik serupa dapat digolongkan menjadi satu klaster. Contohnya adalah *self-organizing NN*.

* + 1. Komponen NN

Seperti pada jaringan saraf biologis, pada NN terdapat *node* yang berperan seperti *neuron*. Pemrosesan informasi berlangsung pada *neuron*-*neuron* tersebut. *Neuron* terhubung dengan *neuron* lainnya menggunakan *connection link* dan memiliki bobot (*weight*) yang terus diperbarui nilainya. Bobot akhir ini yang digunakan apabila ada nilai *input* baru yang diproses untuk mengetahui nilai *output*nya. Ada tahap untuk melakukan 10 pembaruan bobot sampai perulangan yang ditentukan atau sampai nilai pembaruannya tidak berubah secara signifikan. (Fausett, 1994)

Fungsi Aktivasi

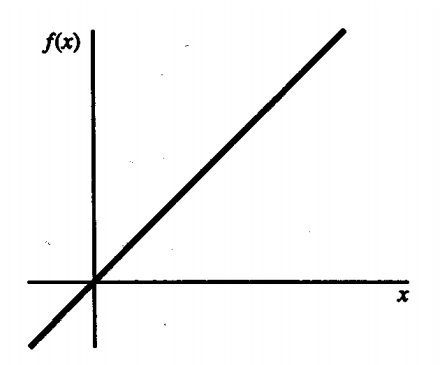
Fungsi aktivasi berfungsi untuk mengubah sinyal *output* menjadi sinyal *input*. Terdapat beberapa macam fungsi aktivasi yang ada pada *Neural Network*.

1. Fungsi linear (identitas)

Fungsi identitas tidak mengubah nilai yang masuk ke *neuron* tersebut. Fungsi linear dirumuskan pada pers.(3.1).

|  |  |
| --- | --- |
| untuk setiap x | (**Error! No text of specified style in document.**.1) |

Kemudian pers.(3.1) dapat digambarkan seperti pada Gambar 3.3

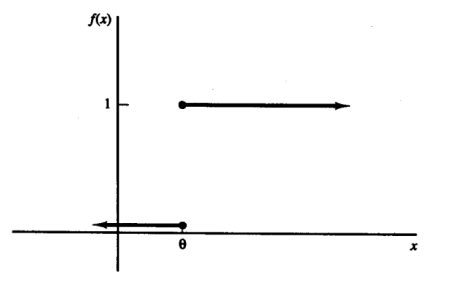


Gambar Error! No text of specified style in document..3 Grafik fungsi identitas (Fausett, 1994)

1. Fungsi undak biner (*binary step function*)

*Neuron* yang mempunyai fungsi undak biner (*binary step function*) menghasilkan keluaran 0 atau 1. Terdapat nilai ambang atau *threshold* (θ) yang menjadi batas apakah nilai yang dihasilkan adalah 0 atau 1. Persamaan untuk fungsi undak biner dituliskan dalam pers.(3.2).

|  |  |
| --- | --- |
|  | (**Error! No text of specified style in document.**.2) |



Gambar Error! No text of specified style in document..4 Grafik fungsi undak biner (Fausett, 1994)

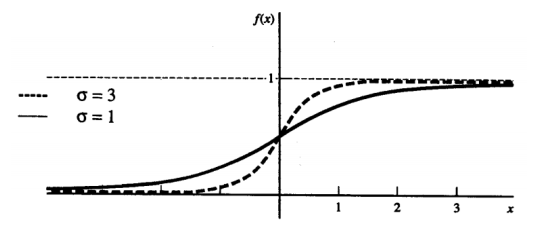
Pada Gambar 3.5, terlihat bahwa grafik yang dibentuk menghasilkan kurva yang lancip dan menunjukkan bahwa nilai yang dihasilkan dari fungsi aktivasi ini adalah 0 atau 1. Nilai *output* bernilai 1 apabila nilai yang masuk lebih besar atau sama dengan nilai ambang yang ditentukan dan akan bernilai 0 apabila nilai yang masuk ke dalam *neuron* lebih kecil dibanding nilai ambang.

1. Sigmoid biner

Fungsi aktivasi sigmoid biner mengubah nilai yang masuk ke dalam *neuron* menjadi antara 0 maupun 1. Sigmoid biner dirumuskan dalam pers.(3.3).

|  |  |
| --- | --- |
|  | (3.3) |

Pada pers.(3.3), semakin besar nilai *x* membuat menjadi 1 dan apabila nilai *x* semakin kecil membuat nilai menjadi 0. Terdapat parameter *σ* yang menentukan nilai yang dihasilkan yang disebut dengan *steepness parameter*. Kurva yang dihasilkan berbentuk seperti huruf S seperti pada Gambar 3.6.



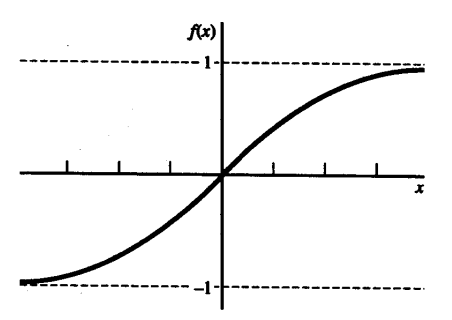
Gambar .5

Grafik sigmoid biner(Fausett, 1994)

Pada Gambar 3.8, kurva dengan garis putus-putus menggunakan *σ* = 3, sedangkan garis lurus menggunakan *σ* = 1.

1. Tanh (*Hyperbolic tangent*)

Sama dengan fungsi sigmoid biner, kurva yang dihasilkan oleh fungsi aktivasi *hyperbolic tangent* juga berbentuk seperti huruf S. Yang membedakan dengan sigmoid biner adalah batas nilai minimal dari *hyperbolic tangent* bukan 0, melainkan -1, seperti pada Gambar 3.7. Oleh karenanya, jangkauan nilai yang dihasilkan dari fungsi aktivasi *hyperbolic tangent* adalah dari -1 sampai 1.



**Gambar 3.7.** Grafik *hyperbolic tangent*(Fausett, 1994)

Fungsi tanh atau dapat dipetakan dari fungsi pada Persamaan (3.4).

|  |  |
| --- | --- |
|  | () |

Persamaan (3.8) dapat membuat batas atas menjadi 1 dan batas bawah menjadi -1. Apabila Persamaan (3.8) disubstitusikan ke dalam Persamaan (3.9), diperoleh Persamaan (3.10).

|  |  |
| --- | --- |
|  | () |
|  | () |
|  | () |
|  | () |

Fungsi bipolar sigmoid erat kaitannya dengan fungsi *hyperbolic tangent* atau *tanh*. Fungsi *tanh* apabila diatur *σ* sebesar 1 dirumuskan dalam Persamaan (3.10). Persamaan (3.11) merupakan persamaan *tanh* yang diturunkan.

***Learning Rate* (α)**

*Learning Rate* adalah salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training.* Dengan kata lain sebagai parameter penentu ketelitian dan kecepatan sebuah arsitektur dalam pembelajaran. Nilai α berada pada range [0,1]. Semakin besarα,maka semakin cepat proses *training* dan *error*nya. Tetapi jika nilai α terlalu besar, maka proses *training* dapat melampaui keadaan optimal (keadaan *error* minimal).

**Bobot**

Bobot (*Weight*) merupakan nilai matematis yang mengekspresikan kekuatan antar *neuron* dari *layer* ke *layer*. Tiap hubungan mempunyai nilai bobot sendiri. Nilai ini menyediakan informasi yang digunakan oleh jaringan untuk memecahkan masalah.

* + 1. *Recurrent Neural Network*

*Recurrent Neural Network* (RNN) merupakan bentuk dari NN di mana pada RNN memungkinkan *output* dari lapisan tersembunyi dapat dijadikan *input* kembali sehingga dapat memperhitungkan riwayat data yang telah dilalui (Fausett, 1994).

* 1. Kalman Filter

*Kalman filter* merupakan persamaan matematis yang termasuk kedalam salah satu penemuan penting dalam prinsip teori kontrol, diterbitkan pada artikel E. Kalman pada tahun 1960. Paling banyak diaplikasikan pada sistem dinamis yang kompleks seperti proses *manufacturing*, penerbangan, perkapalan, dan pesawat luar angkasa (digunakan di sistem pemandu). Sekarang, *Kalman filter* tidak hanya digunakan pada mesin, tetapi juga pada bidang ekonomi, grafis dan lainnya. Bagaimanapun, *Extended Kalman Filter* mulai digunakan pada metode *training* di *Neural Network* karena berkembangnya sistem komputer.

* + 1. *Linear Kalman filter*

State vector (xk) didefinisikan sebagai himpunan data minumum yang secara unik mendeskripsikan tingkah laku sistem yang dinamis, dimana t menunjukkan waktu diskrit. Sederhananya, state vector adalah vektor yang berisi semua variabel sistem yang relevan dari sistem (Haykin, 2001).

***Process Equation***

*Process equation* ditunjukkan pada persamaan (3.11)

|  |  |  |
| --- | --- | --- |
|  | | () |
|  | | () |
| Keterangan: | |  |
| : matriks transisi | : *noise* dari *process* | |
| : Kovarian *noise* dari *process* | : *state*dari t-1 sampai *t.* | |

***Measurement Equation***

|  |  |  |
| --- | --- | --- |
|  | | () |
|  | | () |
| Keterangan: | |  |
| : *observable* pada waktu t | :matriks *measurement* | |
| : Kovarian *noise* dari *measurement* | : *noise* dari *measurement* | |

*Noise* dari dan diasumsikan aditif dan *Gaussian* dengan *zero mean* dan memiliki kovarian dan . dan tidak saling berkorelasi.

Permasalahan Kalman *filter* terletak pada penggabungan solusi *Process* dan *Measurement equation* untuk *state* yang tidak diketahui dengan optimal. Semua data yang didapatkan harus digunakan, terdiri dati vektor untuk menentuan MSE dari *state* untuk semua *t ≥ 1.* Kalman *filter* bekerja dalam dua tahap berulang:

***Time Update***

Disebut juga sebagai tahap prediksi. Persamaan seperti pada (3.15) dan (3.16) dengan menghitung prediksi apriori dari *state* dan *error covariance Superscript* “–“ menandakan prediksi priori.

|  |  |  |
| --- | --- | --- |
|  | | () |
|  | | () |
| Keterangan: | |  |
| :prediksi apriori *state* | : matriks transisi | |
| :Kovarian *noise* dari *process* | : *error covariance* | |

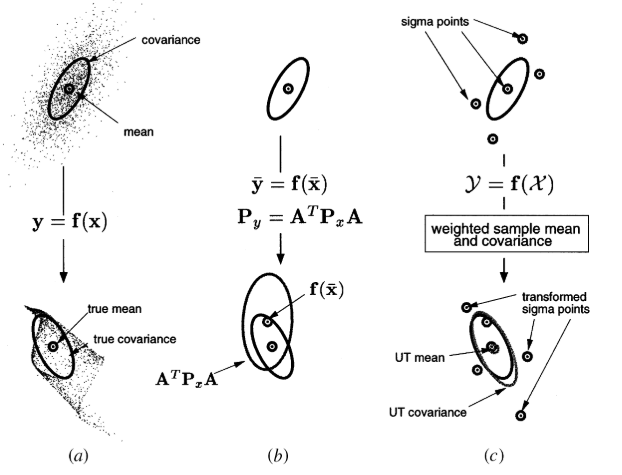
***Prediction Update***

Dikenal juga dengan tahap koreksi, dimana dihitung *Kalman gain* yang digunakan untuk mengkoreksi prediksi dari *state* yang dihasilkan pada tahap *time update* berdasarkan *measurement* dari seperti pada persamaan (3.17).

|  |  |  |
| --- | --- | --- |
|  | | () |
|  | | () |
|  | | () |
| Keterangan: | |  |
| :kovarian *noise* dari *measurement* | : matriks *measurement* | |
| : Kalman *gain* | :prediksi *state* | |
| :kovarian *error* | :observable | |

* + 1. *Unscented kalman filter*

UKF merupakan salah satu metode pengembangan kalman filter yang bertujuan untuk mengupdate bobot. Metode yang digunakan mirip dengan tipe metode Monte Carlo, dimana sejumlah sample dipilih secara acak yang juga diperbanyak melalui transformasi nonlinear. Di sisi lain, sigma points tidak dipilih secara acak, melainkan secara deterministik dan jumlah sigma points yang rendah. Ilustrasi perbedaan UKF dan pengembangan kalman filter linear dan Extended Kalman Filter dapat dilihat pada Gambar 3.8.



**Gambar 3.8** Perbandingan a)Nilai sebenarnya, b)EKF, c)UKF

Himpunan *sigma points* dipilih dan digunakan sebagai *sample* untuk *mean* dan kovarian. Ketika himpunan tersebut diperbanyak melalui sistem *nonlinear* sesungguhnya, maka dapat menangkap mean posterior dan kovarian ke *second-order* dari pengembangan Taylor *series* untuk *nonlinearitas* manapun. Persamaan UKF (Wan & Merwe, 2002) dapat dijabarkan sebagai berikut.

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | () |
|  | |  | () |
|  | |  | () |
| Keterangan: | | |  |
| :titik-titik *Sigma* | :ukuran dimensi data | | |
| : state k-1 |  | | |
|  | |  | () |
|  | |  | () |
|  | |  | () |
|  | |  | () |
|  | |  | () |
| Kemudian vektor sigma diperbanyak melalui fungsi nonlinear | | |  |
|  | |  | () |
|  | |  | () |
|  | |  | () |

* 1. Normalisasi

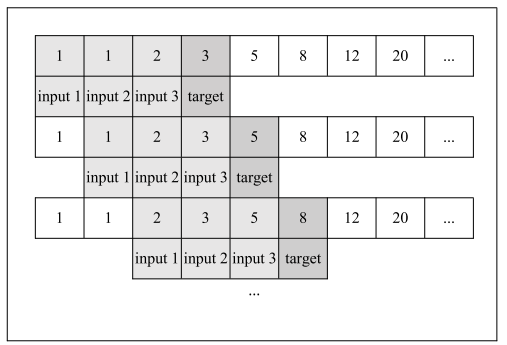
Cs Normalisasi merupakan tahapan dimana data yang didapatkan disamakan jangkauannya. Hal ini dilakukan agar hasil yang didapatkan optimal dengan sistem yang dirancang. Biasanya, nilai data disamakan dengan jangkauan niai real [0,1] atau [-1,1]. Hal ini sangat diperlukan dan berguna apabila data berupa data mentah dengan jangkauan (perbedaan) yang sangat jauh. Dengan normalisasi, dapat dihitung jauhnya perbedaan dan hubungan nilai antar satu data dengan yang lainnya.

Perlu dilakukan beberapa proses sebelum dan sesudah memasukkan data ke dalam model arsitektur. Agar jangkauannya sama, perlu dilakukan normalisasi. Normalisasi data dinyatakan seperti pada Persamaan (3.41). Pada tahap akhir sebelum evaluasi, data dikembalikan seperti data sebenarnya dengan denormalisasi Persamaan (3.42).

|  |  |  |
| --- | --- | --- |
|  | | () |
|  | | () |
| Keterangan: | |  |
| : Data saat dinormalisasi dan didenormalisasi | : Batas bawah jangkauan | |
| : Nilai minimum dari data sebenarnya | : Batas atas jangkauan | |
| : Nilai maksimum dari data sebenarnya |  | |

* 1. *Sliding Window*

Digunakan untuk menentukan besarnya *sliding window* data model *time series*. *Sliding window* menentukan besarnya jumlah hari yang digunakan untuk menentukan nilai prediksi keluaran. Pada pengujian analisis univariat data latih, didapatkan hasil MAPE terkecil dan Dstat terbesar untuk menentukan ukuran *sliding window* yang diimplementasikan pada model.



**Gambar 3.9** Ilustrasi *sliding window*

Sebagai contoh dipilih sliding window dengan ukuran 3 untuk sebuah data time-series maka digunakan data t-3, t-2, dan t-1 untuk meramalkan nilai ke-t. Begitu seterusnya untuk meramalkan nilai t+1, t+2, hingga nilai ke n. Metode sliding window diilustrasikan seperti pada Gambar 3.9.

* 1. Evaluasi

Evaluasi dilakukan untuk mengetahui performa model yang dirancang. Dalam penelitian ini, dilakukan penghitunan MAE, RMSE dan MAPE serta Dstat. seperti pada Persamaan (3.25-48).

* + 1. *Mean absolute error*

Salah satu kriteria pengukuran akurasi prediksi adalah menggunakan mean absolute error (MAE). MAE menentukan rata-rata selisih antara nilai prediksi (〖x̂〗\_i) terhadap nilai sebenarnya (x\_i) untuk n buah data seperti dirumuskan pada Persamaan (3.25).

|  |  |
| --- | --- |
|  | () |

* + 1. *Mean squared error*

MSE adalah suatu ukuran akurasi prediksi dengan mengkuadratkan masing-masing error untuk masing-masing pengamatan dalam sebuah susunan data dan kemudian memperoleh rata-rata jumlah kuadrat tersebut. MSE memberikan bobot yang lebih besar terhadap error dibandingkan dengan error yang kecil, sebab error dikudratkan sebelum dijumlahkan. MSE dapat dihitung dengan persamaan (3.26).

|  |  |
| --- | --- |
|  | () |

* + 1. *Mean absolute percentage error*

Nilai MAE relatif tergantung jarak jangkauan nilai minimal dan maksimal data sehingga agar nilai akurasi dapat dijadikan ke dalam bentuk persen. Untuk mendapatkan nilai dalam bentuk persen, mean absolute percentage error (MAPE) digunkan, yaitu dengan membagi selisih dengan nilai sebenarnya agar diperoleh perbandingan terhadap nilai sebenarnya. MAPE dirumuskan seperti pada Persamaan (3.27).

|  |  |
| --- | --- |
|  | () |

* + 1. *Root mean squared error*

Root-mean-squared error (RMSE) adalah nilai akar pangkat 2 dari rata-rata eror kuadratnya. RMSE dirumuskan seperti pada Persamaan (3.28).

|  |  |
| --- | --- |
|  | () |

* + 1. *Directional statistic*

Directional statistics atau Dstat merupakan tingkat akurasi yang lebih umum digunakan dalam bidang ekonomi khususnya finansial (Pangayom, 2017). Berbeda dengan perhitungan error, Dstat mengukur tingkat kebenaran naik atau turunnya nilai hasil prediksi terhadap nilai naik atau turun yang sebenarnya sehingga nilai Dstat yang lebih tinggi yang diharapkan. Nilai Dstat dirumuskan seperti pada Persamaan (3.29).

|  |  |
| --- | --- |
|  | () |

Dimana 𝑎𝑡 bernilai 1 apabila dan bernilai 0 apabila .

Dalam prediksi, nilai error yang diharapkan adalah nilai yang seminimal mungkin karena menunjukkan bahwa perbedaan nilai yang diramalkan dengan nilai sebenarnya cukup kecil atau hasil prediksi mendekati nilai yang sebenarnya. Untuk Dstat diharapkan nilai yang maksimal karena mendeskripsikan ketepatan arah prediksi.

1. BAB IV  
   ANALISIS DAN RANCANGAN SISTEM
   1. Analilis Permasalahan

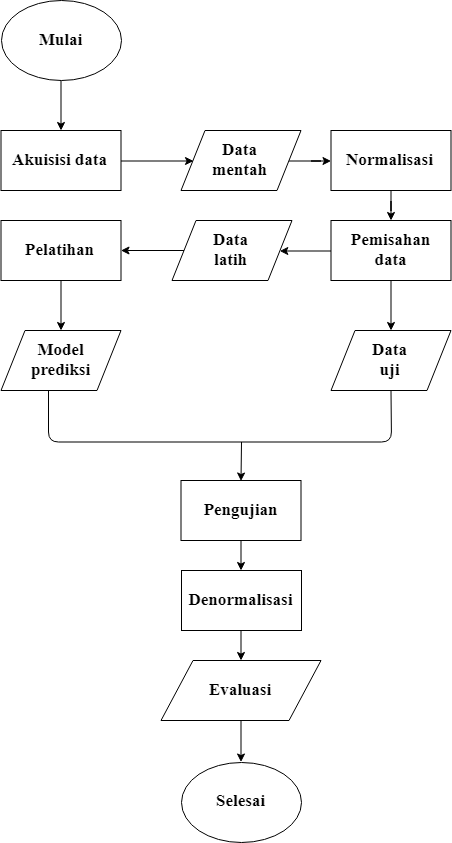
Pada penelitian ini dilakukan penghitungan untuk mencari akurasi dengan performa algoritma RNN-UKF dengan data time series. Data time series harga Bitcoin diambil dari 1 Januari 2011 – 31 Mei 2018. Data Bitcoin kemudian dilakukan normalisasi menggunakan persamaan (3.4). Setelah dilakukan normalisasi data, data ternormalisasi dibagi menjadi data latih dan data uji dengan perbandingan 7:3. Untuk membangun model rancangan diperlukan data latih yang dari sini, data latih dibagi lagi menjadi data latih dan validasi untuk optimasi model perancangan. Data latih dan uji dengan perbandingan 7:3 yang kemudian dilakukan pengujian di model RNN-UKF. Setelah dilakukan pengujian dan implementasi model rancangan, dilakukan denormalisasi data dengan menggunakan kebalikan dari persamaan normalisasi pada awal data mentah. Setelah implementasi, evaluasi performa metode dilakukan dengan perhitungan Dstat, RMSE, MAE, MSE dan MAPEnya. Langkah-langkah penelitian yang dilakukan dialurkan pada Gambar

* 1. Rancangan Umum Sistem

Rancangan umum sistem memiliki langkah-langkah yang dilakukan seperti:

1. Mengumpulkan data *time series* harga Bitcoin.
2. Melakukan praproses data dengan normalisasi data. Jangkauan data diubah menjadi skala yang lebih sesuai jangkuan nilai fungsi aktivasi pada *neuron*.
3. Memisahkan data menjadi dua bagian, yaitu data pelatihan dan data uji.
4. Melakukan pelatihan data pada data pelatihan untuk membuat model prediksi.
5. Melakukan pengujian pada data uji dengan menggunakan model yang telah dilatih sebelumnya.
6. Menghitung akurasi dari model yang sudah dilatih terhadap data.

Rancangan dilakukan dengan tahapan tahapan seperti yang dilustrasikan pada diagram Gambar 4.1.

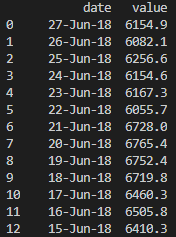


**Gambar 4.1** Diagram alur penelitian

Analisis data juga dilakukan untuk menemukan performa terbaik dari sistem *neural network* yang dirancang dan algoritmanya dalam memprediksi *time-series* data. Penelitian ini menggunakan rancangan sistem dengan pengembangan algoritma *Neural Network.* Pengembangan tersebut adalah dengan menggunakan metode *Recurrent Neural Network* dan *Unscented Kalman Filter* (RNN-UKF).

* 1. Data

Data time series Bitcoin diperoleh dari blockchain.info. Data terdiri dari data harga Bitcoin terhadap USD. Data terdiri dari dua variabel, yaitu waktu dan harga pada saat itu. Data yang diambil dan simpan dalam format .csv dibagi menjadi data latih dan uji.



**Gambar 4.2** Contoh data Bitcoin *sort by newest date*

* 1. Normalisasi Data

Sebelum dilakukan pelatihan, dilakukan tahap praproses terlebih dahulu. Tahap praproses yang dilakukan adalah normalisasi jangkauan nilai atau melakukan normalisasi minimum-maksimum. Normalisasi dengan membuat skala jangkauan nilai menjadi lebih kecil diberlakukan pada variabel harga Bitcoin. Nilai yang besar diperkecil dengan normalisasi agar jangkauannya menjadi [-1,1]. Nilai maksimum, minimum, dan jangkauan pada data digunakan untuk menormalisasi data sesuai skala yang ditentukan menggunakan Persamaan (3.4).

**Tabel 4.1** Cuplikan data yang sudah di normalisasi

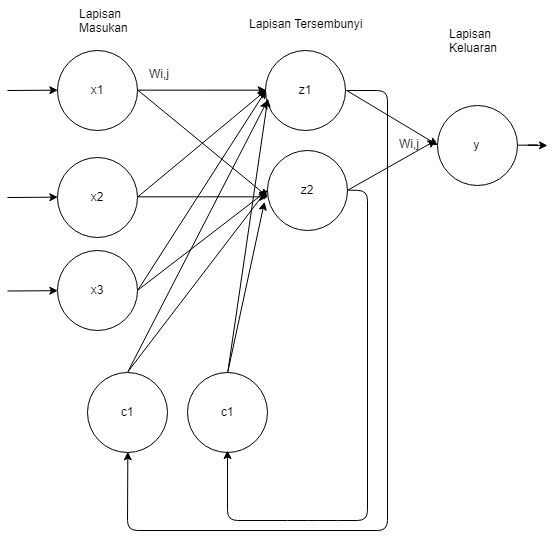
|  |  |  |
| --- | --- | --- |
| Date | Price (USD) | Normalisasi |
| 1/12/2012 | 6.8 | -0.999814 |
| 1/13/2012 | 6.4 | -0.999835 |
| 1/14/2012 | 6.8 | -0.999814 |
| 1/15/2012 | 7 | -0.999804 |
| 1/16/2012 | 6.7 | -0.999819 |
| 1/17/2012 | 5.6 | -0.999876 |
| 1/18/2012 | 5.9 | -0.99986 |
| 1/19/2012 | 6.4 | -0.999835 |
| 1/20/2012 | 6.5 | -0.999829 |
| 1/21/2012 | 6.2 | -0.999845 |
| 1/22/2012 | 6.3 | -0.99984 |
| … | … | … |
| 6/27/2018 | 6,154.90 | -0.68194 |

* 1. Arsitektur NN

Pada *Neural Network*, terdapat bermacam jenis dan arsitektur. Salah satunya adalah *Recurrent Neural Network*. *Neural Network* sendiri biasanya mempunyai 3 layer (*input, hidden, output*). Pada dasarnya, RNN menggunakan 3 layer tersebut. Pada beberapa arsitektur RNN, lebih dari 2 *hidden layer* digunakan. Tetapi penambahan *hidden layer* tidak menjamin akurasi pada prediksi. Terdapat banyak faktor untuk menjamin didapatkan nilai akurasi terbaik. Salah satunya adalah analisis data, performa algoritma dan rancangan sistem. Menurut (McNally, 2016) terlalu banyak *node* (50-100) dapat mempunyai performa, tetapi dapat meningkatkan kemungkinan *overfitting*. Maka karena itu, pada penelitian digunakan antara 10, 20, 30, 40 *neuron* pada *hidden layer*.

Pada sistem ini, digunakan RNN dengan *neuron input* merupakan nilai dari variabel masukan yang digunakan pada *timestep t* yaitu nilai *float* dari Bitcoin. Jika menggunakan satu *neuron* masukan, maka data yang dimasukkan adalah data (kemarin) untuk memprediksi data (hari ini). Jika menggunakan tiga *neuron*, maka digunakan (tiga hari kebelakang).

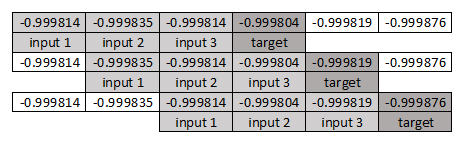
Pada hidden layer, digunakan *neuron* dengan fungsi aktivasi *hyperbolic tangent*. Pengaruh jumlah *hidden layer*, diuji dengan variasi dari satu unit *neuron* hingga 10 unit *neuron*. Pada output layer, digunakan *many-to-one*. Yaitu satu *neuron* sebagai *output* yang ditetapkan oleh *hyperbolic tangent* karena merupakan masalah regresi.



**Gambar 4.3** Arsitektur RNN yang digunakan

Selain tiga layer tersebut, digunakan juga *context layer* untuk menyimpan nilai dari *hidden layer* pada *timestep t-1* untuk digunakan kembali sebagai masukan pada *hidden layer* pada *timestep t*. Nilai pada *context layer* terus diupdate hingga pelatihan selesai dengan jumlah *neuron* yang sama dengan *neuron* pada *hidden layer* seperti pada Gambar 4.3.

Pada *neuron input*, input yang digunakan diilustrasikan seperti pada Gambar 4.4. Pada gambar, diilustrasikan *Sliding Window* yang digunakan adalah 3 data, dengan data ke-4 menjadi target. Kemudian begitu seterusnya untuk *neuron* masukan.



**Gambar 4.4** Ilustrasi *Sliding Window*

* 1. *Unscented Kalman Filter*

Algoritma pembelajaran yang digunakan pada penelitian ini merupakan algoritma gabungan dari *Recurrent Neural Network* dan update bobot *Unscented Kalman Filter*. Berdasarkan persamaan matematis (Wan & Merwe, 2002), *Unscented Kalman Filter* memiliki persamaan dan tahapan seperti seperti berikut:

|  |  |  |
| --- | --- | --- |
| **Inisialisasi dan transformasi** | |  |
|  | | () |
|  | | () |
|  | | () |
|  | | () |
| For seaharusnyaw ada penjelasannya lagi Tambahin blab la bla | |  |
| **Menghitung titik sigma:** | |  |
|  | | () |
| Keterangan: | |  |
| : matriks transisi | : *noise* dari *process* | |
| : Kovarian *noise* dari *process* | : *state*dari t-1 sampai *t.* | |

Pada Persamaan 4.5, menghitung sigma points dengan sebagai kolom ke-i dari matriks akar kuadrat dari dengan digunakan untuk mengontrol kovarian matriks dengan .

|  |  |
| --- | --- |
| Tambahin blab la bla |  |
| ***Time Update:*** |  |
|  | () |
|  | () |
|  | () |
|  | () |
|  | () |

|  |  |
| --- | --- |
| Tambahin blab la bla |  |
| ***Measurement Update:*** |  |
|  | () |
|  | () |
|  | () |
| ) | () |
|  | () |
|  | () |

Paramerer scalar dari dan menentukan penyebaran dari *sigma points* di sekitar . Dimana juga mempengaruhi pengurangan order error dari pendekatan mean dan kovarian. Konstan biasanya ditetapkan antara [-4,1], dan atau 0 (Bisoi & Dash, 2014).

|  |  |
| --- | --- |
| Tambahin blab la bla |  |
|  | () |
|  | () |
| dengan |  |
| dan | () |
|  | () |
|  | () |
|  | () |
| Dengan *priori* dari error kovarian dijabarkan dengan |  |
|  | () |
| , | () |
|  | () |

*Sigma point* kemudian diperbanyak melallui *output* model prediksi data historis untuk mengestimasi *mean* dan kovarian dari

|  |  |
| --- | --- |
|  |  |
|  |  |
| Step2 : Measurement update???????????????? |  |
|  |  |

* 1. Prosedur Pelatihan

Pada penelitian, dilakukan langkah-langkah yang terdiri dari beberapa bagian, yakni:

* + 1. Data latih

Data yang digunakan adalah 70% data dari data keseluruhan yang telah dinormalisasi. Data yang terhitung adalah data sebanyak 1647 data.

* + 1. *Preprocessing*

Dilakukan normalisasi agar jangkauan nilai harga Bitcoin tidak terlalu jauh. Normalisasi dilakukan agar hasil prediksi yang diharapkan dapat maksimal. Normalisasi data akan menghasilkan nilai [-1,1] yang kemudian akan dilakukan pemilihan data dengan fungsi aktivasi hyperbolic tangent. Penghitungan dilakukan pada data dengan jumlah input yang telah ditentukan dengan metode *sliding window*.

* + 1. Inisialisasi bobot awal

Terdapat tiga bobot awal yang ditentukan secara random [-1,1] untuk digunakan dalam aristektur RNN yaitu sebagai betikut:

1. sebagai bobot dari *input* ke *context layer.*

2. sebagai bobot dari *context* ke *hidden layer.*

3. sebagai bobot dari *hidden* ke *output layer.*

* + 1. Inisialisasi parameter pelatihan

Parameter pembelajaran ditentukan pada awal sistem. Dalam sistem ini, parameter pembelajarannya adalah sebagai berikut:

Tabel 4.3 Parameter pelatihan

|  |  |  |
| --- | --- | --- |
| No. | Parameter | Penjelasan |
| 1. | *Process noise covariance Q* | Matriks berukuran , sebagai kuantitas dari bobot. |
| 2. | *Measurement noise covariance R* | Matriks berukuran , sebagai jumlah neuron pada *output layer* |
| 3. | *Error covariance P* | Matriks berukuran , diinisialisasikan dengan nilai |
| 4. | Jumlah *epoch* | Diterapkan sebanyak 2000 epoch |
| 5. | Fungsi aktivasi | Fungsi aktivasi linear dan *hyperbolic tangent* |

*Q* dan *R* memiliki nilai dimana merupakan skalar dan I merupakan matriks identitas. Akan dilakukan pengujian terhadap nilai untuk mengetahui pengaruh nilai *Q* dan *R* terhadap hasil prediksi.

* + 1. Proses update bobot dengan UKF

Algoritma UKF terdapat di dalamnya 2 bagian penting, yaitu tahap prediksi dan tahap *update*.

1. Hitung kumpulan dari titik sigma (*Sigma Points*)

2. Tambahkan bobot awal ke setiap titik sigma

3. Transformasi titik-titik menjadi fungsi non-linear

4. Huting *Gaussian* dari bobot dan transirasi titik-titik tersebut

5. Hitung *mean* dan varian dari Gausian terbaru

* + 1. Perhitungan Prediksi

Prediksi dilakukan dengan pembagian data menjadi data latih dan data uji yang sebelumnya telah dilakukan normalisasi data. Bobot UKF yang didapatkan dari pelatihan, diterapkan pada data uji. Setelah pemrosesan prediksi and data telah didapatkan, data didenormalisasi ke awal lagi seperti nilai sebenarnya (harga Bitcoin dalam USD). Setelah didenormalisasi, dilakukan pengujian akurasi MAE, MSE, MAPE, RMSE, Dstat untuk mengetahui kekuatan model rancangan dan bsesarnya error terhadap data sebenarnya.

* 1. Rancangan Pengujian

Membangun model prediksi *neural network* membutuhkan beberapa parameter pelatihan karena setiap arsitektur NN yang berbeda, menghasilkan akurasi yang berbeda juga. Maka dilakukan pengujian yang meliputi:

1. Pengujian laju pembelajaran.

2. Pengujian arsitektur NN.

3. Pengujian ukuran *sliding window*.

4. Pengujian akurasi MAE, MSE, MAPE, RMSE, Dstat untuk mengetahui kekuatan model rancangan.

Tujuan dari pengujian ini adalah untuk mendapatkan akurasi dengan *error* seminimal mungkin dengan melakukan persiapan dan rancangan sistem sebaik mungkin dengan teori dan penelitian sebelumnya

1. BAB V  
   IMPLEMENTASI
   1. Lingkungan Implementasi

Penelitian ini menggunakan sistem seperti berikut:

1. Laptop Asus VivoBook AMD Rayzen 5, 8 GB RAM

2. Sistem operasi Windows 10 64bit

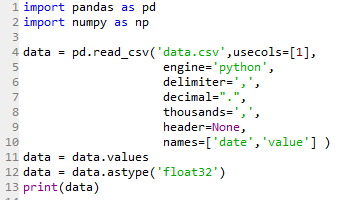
3. Python 3.7.1 64bit

4. IDE Spyder

5. Library Pandas, Numpy, Matplotlib

* 1. Data

Data diambil dari blockchain.info dan disimpan kedalam file .csv. Kemudian data dilakukan perubahan jenis data *value* sebagai float32.



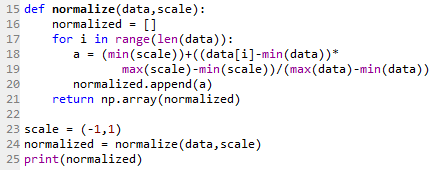
**Gambar 5.1** Kode untuk membaca data.csv

data

**Gambar 5.2** Data dibaca sebanyak 2359 *rows* x 1 *columns*

* 1. Normalisasi Data

Dilakukan normalisasi jangkauan data [-1,1] dengan menggunakan Persamaan (3.4) dan menyimpannya ke dalam variabel scale. Fungsi normalize menggunakan variabel data yang dimuat dan scale sebagai skala yang diinginkan [-1,1].



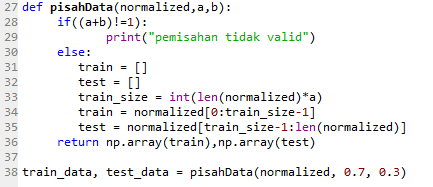
**Gambar 5.3** Kode untuk menormalisasi data

tabel_normalisasidata

**Gambar 5.4** Tabel normalisasi data dan contoh nilai [-1,1]

* 1. Pemisahan Data

Pemisahan data dilakukan sebagai salah satu bentuk persiapan data sebelum dimasukkan ke sistem. Pada penelitian ini, data dipisahkan menjadi dua bagian utama, yaitu data latih dan data uji. Kode pemisahan data dapat dilihat pada Gambar 5.4. Data dipisah dengan rasio 7:3 untuk data latih dan data uji. Model yang dirancang kemudian menggunakan 70% dari keseluruhan data untuk pelatihan dan sisanya untuk diuji.



**Gambar 5.5** Kode pemisahan data

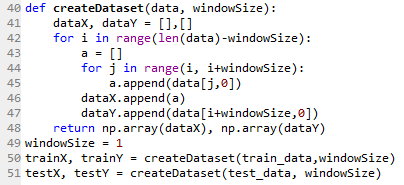
tabel_pemisahandata

**Gambar 5.6** Tabel pemisahan data

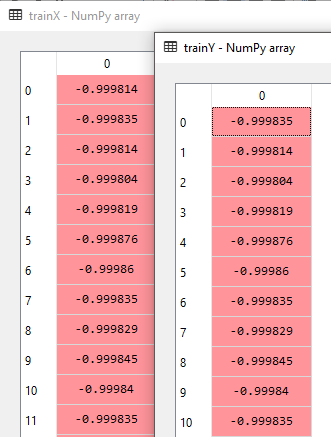
Pada Gambar 5.4 dapat dilihat bahwa kode pemisahan data diawali dengan membuat fungsi pisahData dengan parameter (data), persentase data train (a) dan test (b). Kemudian kode hanya berjalan jika data awal sama dengan 100% yang kemudian data latih disimpan dalam variabel train\_data dan data uji dalam test\_data. Setelah program dijalankan, didapatkan bahwa keseluruhan data berjumlah 2359 data yang kemudian dipisah menjadi 1650 data latih dan 709 data uji yang terlihat pada Gambar 5.6.

* 1. Menentukan I/O JST

Pada arsitektur NN penelitian ini, digunakan *many-to-one* arsitektur dimana *input* berupa banyak *nodes* atau *neuron*. Sedangkan output berupa satu keluaran saja untuk memprediksi harga Bitcoin. Menentukan masukan dan keluaran RNN dapat dianalisis menggunakan fungsi windowSize seperti pada Gambar 5.7. Awalnya, dibuat fungsi createDataset dengan variabel (data dan windowSize). Kemudian masukan dimuat dalam dataX dan target ke dalam dataY. Masing-masing masukan data latih dan uji dideklarasikan pada trainX dan trainY.

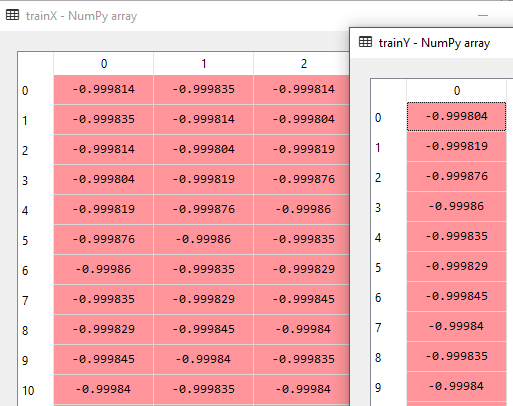
****

**Gambar 5.7** Kode input data



**Gambar 5.8** Tabel input data dengan windowSize=1

Jika windowSize=3, maka pada data ke 4 digunakan sebagai target. Sedangkan data ke-1, ke-2, dan ke-3 menjadi masukan pertama seperti pada Gambar 5.9. Begitu pula seterusnya sampai data ke-n.



**Gambar 5.9** Tabel input data dengan windowSize=3

* 1. Implementasi RNN

Arsitektur NN yang digunakan terbagi menjadi *input layer*, *hidden layer*, dan *output layer*. Pada penelitian ini, digunakan *Recurrent Neural Network* dengan konsep *many-to-one*. Banyaknya *neuron* pada *input layer* ditentukan oleh besarnya *sliding window* yang dipilih. *Neuron* pada *hidden layer* menjadi bahan pengujian antara 1 unit sampai 10 unit. Sedangkan *output layer* berjumlah satu buah *neuron*. Variabel savedLoss digunakan untuk menyimpan nilai *loss* selama pelatihan.

Fungsi aktivasi *hyperbolic tangent* digunakan untuk memfilter nilai yang masuk pada RNN dengan Persamaan (3.11) dengan kode seperti pada Gambar 5.10.

**Gambar 5.10** Fungsi hyperbolic tangent

Pada akhir sistem, semua data dikembalikan nilainya seperti awal lagi yang dalam hal ini seperti data sebelum di normalisasi ke dalam jangkauan [-1,1]. Maka data dilak ukan proses pengembalian dengan persamaan 3.5. Kode dari fungsi denormalisasi dijabarkan seperti pada gambar 5.11.

**Gambar 5.11** Fungsi denormalisasi

Fungsi denormalisasi dengan tiva variabel, yaitu data(x), source sebagai data asal dan scale sebagai skala. Kemudian dilakukan perhitungan dengan persamaan matematika dimana hasil denormalisasi untuk valPredict disimpan dalam denormValPredict, dan hasil dari denormalisasi untuk testPredict disimpan dalam denormTestPredict.

* 1. Pengujian

Tahap akhir dalam penelitian adalah menguji sistem yang telah dirancang dan telah diimplementasikan datanya. Untuk menguji sistem rancangam, maka dihitung performa model dengan menghitung MAE, MSE, MAPE, RMAE dan Dstat.

* + 1. MAE

Mean Absolute Error (MAE) merepresentasikan rata-rata kesalahan (error) absolut antara nilai prediksi dan sebenarnya.

* + 1. MSE

Mean Squared Error (MSE) merepresentasikan sebagai error dengan varian ditambah dengan kuadrat bias suatu model. Biasanya nilai MSE lebih besar dibandingkan dengan MAE karena dikuadratkan. Akibatnya nilai yang kecil semakin kecil dan yang besar semakin besar.

* + 1. MAPE
    2. RMSE
    3. Dstat

1. BAB VI  
   HASIL PENELITIAN DAN PEMBAHASAN

Pada Bab ini, dipaparkan hasil pengujian implementasi NN-UKF dengan arsitektur *Recurrent Neural Network* untuk memprediksi nilai *cryptocurrency* Bitcoin. Data historis yang dipakai adalah harga bitcoin dengan nilai tukar USD dari 12 Januari 2012 sampai 27 Juni 2018 sebanyak 2359 data.

**Gambar 6.1** Grafik pemisahan data

Data diambil dari blockchain.info dan disimpan dalam file .csv. Kemudian data di *load* dengan dua kolom berbeda, yaitu date dan value. Kemudian kolom value dinormalisasi dengan persamaan 3.4 agar jangkauan data tidak terlalu jauh [-1,1]. Setelah dinormalisasi, data dipisah menjadi 2 bagian dengan rasio 7:3 antara data latih dan data uji. Grafik data untuk tiap bagiannya disajikan pada Gambar 6.1 untuk data sebelum dinormalisasi dan Gambar 6.2 yang telah dinormalisasi.

**Gambar 6.2** Grafik pemisahan data ternormalisasi

* 1. Pengujian Arsitektur JST
  2. Pengujian Ukuran *Sliding Window*
  3. Pengujian Jumlah *Neuron*

1. BAB VII  
   KESIMPULAN
   1. Kesimpulan
   2. Saran

1. DAFTAR PUSTAKA

Albariqi, R., 2018. Prediksi Perubahan Harga Bitcoin Menggunakan Jaringan Syaraf Tiruan. *Skripsi.*

Amjad, M. J. & Shah, D., 2016. Trading Bitcoin and Online Time Series Prediction. *Proceedings of the Time Series Workshop at NIPS,* pp. 1-15.

Bisoi, R. & Dash, P., 2014. A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. *Applied Soft Computing Journal,* Volume 19, pp. 41-56.

Bisoi, R. & Dash, P. K., 2014. A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. *Applied Soft Computing Journal,* Volume 19, pp. 41-56.

Fausett, L. V., 1994. *Fundamentals of neural networks: Architectures, algorithms, and applications.* Englewood Cliffs(NJ): Prentice-Hall.

Haykin, S., 2001. *Kalman Filtering and Neural Networks.* New York, USA: John Wiley & Sons, Inc..

Hazazi, M. A., 2018. Reccurent Neural Network dan Extended Kalman Filter untuk Peramalan Nilai Tukar Mata Uang. *Skripsi.*

Kenawas, M. R. & Sulistiowati, 2016. Legalitas Bitcoin sebagai mata uang dan metode pembayaran di Indonesia. *Skripsi.*

Kim, Y. B. et al., 2016. Predicting Virtual World User Population Fluctuations with Deep Learning. *PLoS ONE,* 11(12), pp. 1-12.

Kristoufek, L., 2013. *BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era,* s.l.: s.n.

Mangan, D. M., 2013. Bitcoin: Cara Kerja dan Perbandingannya dengan Mata Uang Konvensional. *Makalah Kriptografi,* pp. 1-5.

Martina, M., Lunesu, I. & Marchesi, M., 2015. Bitcoin Spread Prediction Using Social And Web Search Media. *UMAP Workshops.*

McNally, S., 2016. Predicting the price of Bitcoin using Machine Learning. *Doctoral dissertation, Dublin, National College of Ireland.*

Nakamoto, S., 2009. Bitcoin: A Peer-to-Peer Electronic Cash System.

Saptoro, A., 2012. Extended and unscented kalman filters for artificial neural network modelling of a nonlinear dynamical system. *Theoretical Foundations of Chemical Engineering,* 46(3), pp. 274-278.

Siang, J. J., 2009. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB.* Yogyakarta: Andi.

Torres, D. G. & Qiu, H., 2018. Applying Recurrent Neural Networks for Multivariate Time Series Forecasting of Volatile Financial Data. pp. 1-10.

Wan, E. A. & Merwe, R. v. d., 2002. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000,* Issue 3, pp. 153-158.

Zhan, R. & Wan, J., 2006. Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation. *IEEE Signal Processing Letters,* 13(7), pp. 445-448.

1. LAMPIRAN A  
   SOURCE CODE

Perhitungan Data