

# Aplikasi Pemetaan Penjualan Produk Oli Deltalube Berbasis Android

Dedi Arif Rahman<sup>1</sup>, Ahmad Bagus Setiawan<sup>2</sup>, Made Ayu Dusea Widyadara<sup>3</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

<sup>3</sup>Teknik Mesin, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: \*<sup>1</sup>[dediarif.azz3@gmail.com](mailto:dediarif.azz3@gmail.com), <sup>2</sup>[bagus.este@gmail.com](mailto:bagus.este@gmail.com) <sup>3</sup>[madedara@gmail.com](mailto:madedara@gmail.com),

**Abstrak** – Produk oli deltaulbe merupakan produk oli unggulan yang ternama di pasaran. Banyak toko yang menjual produk ini tetapi tidak semua orang tahu lokasinya. Berakibat para pengguna produk sering kesusahan saat mencari lokasi produk oli yang sesuai dengan kebutuhan. permasalahan tersebut akan di gunakan peneliti untuk membuat sebuah aplikasi pemetaan penjualan produk Oli Deltalube yang akan di lengkapi dengan fitur google maps untuk memudahkan pembeli dalam mencari lokasi produk yang di cari. Pengguna akan lebih mudah dalam mengakses informasi lokasi pemasaran produk dan pembelian produk hanya dengan smartphone saja. Sedangkan bagi distributor dapat memantau persebaran penjualan produk oli deltalube. Aplikasi yang akan dibuat dengan judul “APLIKASI PEMETAAN PENJUALAN PRODUK OLI DELTALUBE BERBASIS ANDROID”.

**Kata Kunci** — Android, Algoritma Dijkstra, Website

## 1. PENDAHULUAN

Intensitas pertumbuhan teknologi yang semakin maju membuat semua aktivitas manusia tidak lepas dari peran teknologi, termasuk di sektor keuangan. Mata uang digital atau yang biasa disebut dengan *cryptocurrency* baru-baru ini telah menjadi salah satu topik yang paling banyak dibahas dikalangan masyarakat dan banyak juga masyarakat yang sudah mulai menggunakannya.

*Ethereum* adalah *platform blockchain* terbuka untuk membangun aplikasi yang terdesentralisasi (*Dapps*). *Dapps* juga disebut sebagai kontrak pintar atau kode dari kontrak pintar, yaitu program yang menentukan bagaimana nilai bergerak. Segala sesuatu yang memiliki nilai seperti uang, logam, barang dan jasa dapat diuntungkan dari aplikasi. Beberapa keunggulan *Dapps* yaitu penghapusan server dan skalabilitas besar besaran, *Dapps* sebagai internet tanpa *server* atau internet pasca *server*, yang berarti penggunaan aplikasi di internet tidak lagi berdasarkan ada satu sever atau satu perusahaan, aplikasi didasarkan pada internet itu sendiri [1].

Perubahan harga *ethereum* yang sewaktu-waktu dapat berubah membuat pengguna dan *trader* (orang yang melakukan trading) apalagi yang masih pemula, sering kali mendapatkan *profit/keuntungan* yang kurang maksimal bahkan mengalami kerugian ketika melakukan *trading* (perdagangan). Untuk mengatasi perubahan harga yang sangat tidak menentu maka dibutuhkan sebuah aplikasi prediksi harga *ethereum* untuk membantu memprediksi harga *ethereum* kedepannya supaya bisa

meminimalisir resiko kerugian dan juga bisa mendapatkan profit yang maksimal.

Untuk memprediksi dibutuhkan suatu metode yang tepat agar prediksi yang dilakukan bisa mendapatkan hasil yang seakurat mungkin. Metode *Autoregressive Integrated Moving Average (ARIMA)* adalah salah satu metode yang dapat digunakan dalam memprediksi perubahan harga. Metode ARIMA merupakan suatu metode yang cocok digunakan untuk meramal variabel secara cepat, akurat, dan sederhana karena hanya membutuhkan data variabel yang akan diramal [2].

Oleh karena itu pada penelitian ini menggunakan metode ARIMA, dan diharapkan dengan adanya aplikasi prediksi harga *ethereum* tersebut dapat mempermudah pengguna dan *trader ethereum* dalam memprediksi harga kedepannya.

## 2. METODE PENELITIAN

Dalam penelitian ini dilakukan proses tahapan pada penelitian, dibawah ini merupakan uraian metode penelitian yang dilakukan.

### 2.1 Prosedur Penelitian

#### a. Studi Literatur

Tahap ini dilakukan untuk melengkapi pengetahuan serta mencari referensi dengan cara pengumpulan data pustaka, artikel penelitian dan pendalaman tentang metode Autoregressive Integrated Moving Average (ARIMA).

#### b. Pengumpulan Data

Dataset yang akan digunakan dalam penelitian ini adalah data historis harian

ethereum dari situs Yahoo Finance selama 3 tahun 4 bulan mulai dari 1 Januari 2019 sampai 30 April 2022. Dataset berjumlah 1216 data dan memiliki 7 buah atribut yaitu date, open, high, low, close, adj close, dan volume.

c. Perancangan Sistem

Pada penelitian ini perancangan sistem berdasarkan hasil analisa sistem yang kemudian dibuat menjadi alur program sistem. Pada tahapan perancangan sistem pada penelitian ini meliputi perancangan sistem prediksi dan perancangan sistem aplikasi berbasis website sebagai (user interface).

d. Pembuatan Sistem

Pada tahap pembuatan system ini merupakan proses untuk membangun aplikasi prediksi berbasis website sesuai dengan perancangan yang telah dibuat pada tahap sebelumnya yaitu pada tahap perancangan system.

e. Implementasi Sistem

Setelah tahap perancangan dan pembuatan sistem, proses mengimplementasikan hasil dari tahapan sebelumnya yaitu desain dan rancangan program diantaranya, proses perancangan desain antar muka program dan proses prediksi sesuai dengan rancangan program yang telah dibuat tersebut.

f. Pengujian

Pada tahap pengujian sistem merupakan tahap implementasi dari hasil program yang telah dibuat berdasarkan proses pengujian program secara menyeluruh baik pengujian fungsional ataupun desain antar muka program dan melakukan perbaikan jika ditemui kesalahan.

g. Perbaikan Sistem

Perbaikan sistem adalah bagian dari tahapan – tahapan pengujian dan evaluasi sudah dilaksanakan serta mengumpulkan data berupa informasi suatu sistem, jika ditemukan error pada program maupun desain interface akan dilakukan perbaikan untuk menghasilkan suatu sistem aplikasi yang berjalan dengan baik.

## 2.2 Metode ARIMA

*Autoregressive Integrated Moving Average* (ARIMA) adalah model yang secara penuh mengabaikan *independent* variabel dalam

membuat peramalan. ARIMA menggunakan nilai masa lalu dan sekarang dari variabel *dependent* untuk menghasilkan peramalan jangka pendek yang akurat. ARIMA cocok jika observasi dari deret waktu (*time series*) secara statistik berhubungan satu sama lain (*dependent*) [3].

## 2.3 Prosedur Pembentukan ARIMA

Secara umum, model ARIMA ditulis dengan ARIMA ( $p, d, q$ ) yang artinya model ARIMA dengan derajat AR ( $p$ ), *differencing* ( $d$ ), dan derajat MA ( $q$ ). Berikut langkah-langkah pembentukan model secara *iterative* [4] :

a. Tahapan Identifikasi

Hal pertama yang dilakukan pada tahap ini adalah apakah *time series* bersifat *stasioner* atau *nonstasioner* dan bahwa aspek-aspek AR dan MA dari model ARIMA hanya berkenaan dengan *time series* yang *stasioner*. Tahap identifikasi merupakan suatu tahap yang digunakan untuk mencari atau menentukan nilai  $p$ ,  $d$  dan  $q$  dengan hasil *autocorrelation function* (ACF) dan *partial autocorrelation function* (PACF). Hasil perhitungan ini diperlakukan untuk menentukan model ARIMA yang sesuai, apakah ARIMA( $p,0,0$ ) atau AR( $p$ ), ARIMA( $0,0,q$ ) atau MA( $q$ ), ARIMA( $p,0,q$ ) atau ARMA( $p,q$ ), dan ARIMA( $p,d,q$ ).

b. Tahap Estimasi

Tahap berikutnya setelah  $p$  dan  $q$  ditentukan adalah dengan estimasi parameter AR dan MA yang ada pada model. Pada tahap ini, teknik perhitungan secara matematis relatif kompleks, sehingga pada umumnya para peneliti menggunakan bantuan *software* yang menyediakan fasilitas perhitungan seperti *Rstudio*, *SPSS*, *Eviews*, *Minitab* dan lain-lain.

c. Tahap Tes Diagnosa

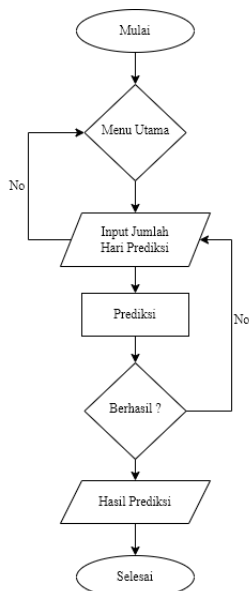
Setelah model ditentukan, kemudian dicek apakah model cocok dengan data dan memenuhi persyaratan model peramalan yang baik. Jika estimasi *residual*-nya *white noise* maka model cocok, namun jika tidak maka harus dilakukan pengecekan kembali. Model dikatakan memadai jika asumsi dari *error* memenuhi proses *white noise* dan berdistribusi normal. Jika *residual*-nya ternyata tidak *white noise* maka modelnya dapat dikatakan tidak tepat dan perlu dicari spesifikasi yang lebih baik.

## 2.4 Desain Sistem Arsitektur

Desain sistem arsitektur merupakan gambaran sketsa suatu alur sistem yang akan berjalan [5]. Berikut desain sistem arsitektur pada sistem yang dibangun :

### a. Flowchart

*Flowchart* adalah penggambaran secara grafik asal langkah-langkah serta urutan mekanisme suatu program. umumnya menghipnotis penyelesaian problem yang khususnya perlu dipelajari serta dievaluasi lebih lanjut [6]. *Flowchart* aplikasi ini menggambarkan alur kerja sistem aplikasi mulai dari awal sampai akhir. Ketika *user* membuka aplikasi user akan berada pada halaman utama. Pada halaman prediksi *user* bisa melakukan prediksi dengan memasukkan jumlah hari yang ingin di prediksi, jika *user* tidak ingin memprediksi bisa kembali ke menu yang lainnya. Setelah itu sistem akan melakukan proses prediksi, jika gagal *user* bisa menginputkan jumlah hari kembali. Jika proses prediksi berhasil sistem akan menampilkan hasil prediksi sesuai dengan jumlah hari yang diinginkan oleh *user*. Gambar *flowchart* aplikasi dapat dilihat pada gambar 1.

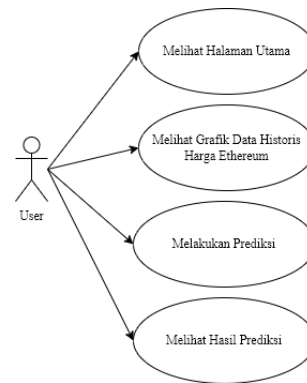


Gambar 1. Flowchart Aplikasi

### b. Use Case Diagram

*use case diagram* merupakan pemodelan buat kelakuan (*behavior*) sistem isu yg akan dibuat, *use case diagram* digunakan buat mengetahui fungsi apa saja yang terdapat di dalam sistem dan yang berhak menggunakan fungsi-fungsi tersebut [7]. *Use case diagram* ini mendeskripsikan interaksi *user* dalam sebuah sistem. Pertama

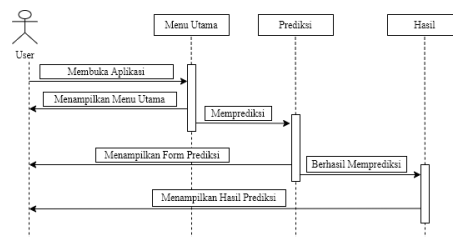
*user* dapat melihat halaman utama berisikan deskripsi tentang *ethereum*. Selanjutnya *user* dapat melihat grafik dan data historis harga *ethereum*. Berikutnya *user* dapat melakukan proses prediksi untuk mengetahui harga *ethereum* kedepannya. Dan terakhir *user* dapat melihat hasil prediksi yang telah dilakukan. Gambar *use case diagram* dapat dilihat pada gambar 2.



Gambar 2. Use Case Diagram

### c. Sequence diagram

*Sequence Diagram* adalah *tool* yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek [8]. *Sequence diagram* dapat diketahui bahwa *user* dapat mengakses menu utama dan sistem menampilkan menu utama kepada *user*. Selanjutnya *user* memilih halaman prediksi dan sistem menampilkan *form* untuk melakukan prediksi. Setelah *user* melakukan prediksi, sistem akan menampilkan informasi berupa hasil prediksi yang telah dilakukan oleh *user*. Gambar *sequence diagram* dapat dilihat pada gambar 3.

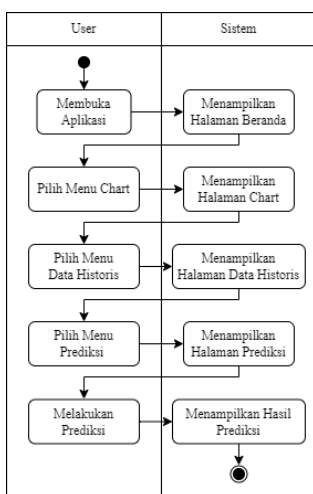


Gambar 3. Sequence Diagram

### d. Activity Diagram

*Activity diagram* menggambarkan aliran fungsionalitas dalam suatu sistem informasi. Secara lengkap, *activity diagram* mendefinisikan dimana *workflow* dimulai, dimana berhentinya, aktifitas apa yang terjadi selama *workflow*, dan bagaimana urutan kejadian aktifitas tersebut [9]. Pada

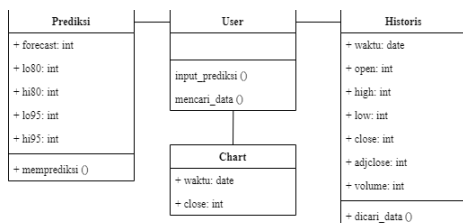
*activity diagram* ini terlihat ada beberapa tahap yang perlu di jalankan. Dimulai dari *user* membuka aplikasi sistem akan menampilkan halaman beranda kepada *user*. Selanjutnya ketika *user* memilih menu *chart* sistem akan menampilkan halaman *chart*. Berikutnya ketika *user* memilih menu data historis sistem akan menampilkan halaman data historis. Selanjutnya *user* memilih menu prediksi sistem akan menampilkan halaman prediksi, dan ketika *user* melakukan prediksi sistem akan menampilkan hasilnya. Gambar *activity diagram* dapat dilihat pada gambar 4.



Gambar 4. Activity Diagram

e. Class Diagram

*Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun *system* [10]. Pada *class diagram* ini terlihat ada beberapa *class* beserta atributnya yang telah dibentuk untuk membangun sistem. Bisa dilihat bahwa *user* dapat terhubung dengan *class* yang lainnya mulai dari *class* prediksi, historis, dan chart. Gambar *class diagram* dapat dilihat pada gambar 5.



Gambar 5. Class Diagram

### 3. HASIL DAN PEMBAHASAN

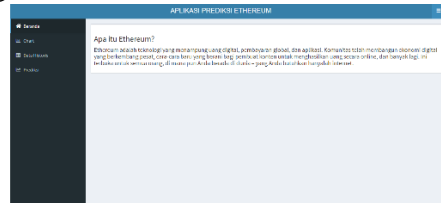
#### 3.1 Tampilan Antarmuka

Berdasarkan hasil implementasi yang telah dilakukan maka dibawah ini adalah hasil tampilan antarmuka dari aplikasi yang telah

dibuat. Terdapat beberapa halaman pada aplikasi tersebut antara lain:

a. Halaman Beranda

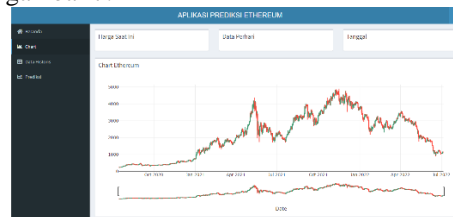
Halaman beranda ini *user* dapat melihat informasi berupa deskripsi atau penjelasan tentang *ethereum* mulai dari pengertian, pembuat, dan lain-lainnya. Gambar halaman beranda dapat dilihat pada gambar 6.



Gambar 6. Halaman Beranda

b. Halaman Chart

Halaman *chart* ini *user* dapat melihat informasi berupa grafik pergerakan harga *ethereum* beberapa periode sebelumnya dan diatas grafik ada keterangan harga *ethereum* saat ini, data per hari, dan tanggal saat ini. Gambar halaman *chart* dapat dilihat pada gambar 7.



Gambar 7. Halaman Chart

c. Halaman Data Historis

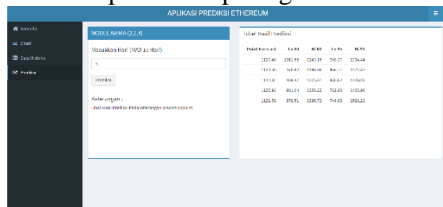
Halaman data historis ini *user* dapat melihat informasi berupa tabel historis harga *ethereum* saat ini dan sebelumnya. *User* juga dapat mencari data harga *ethereum* sesuai dengan tanggal yang diinginkan. Gambar halaman data historis dapat dilihat pada gambar 8.

Gambar 8. Halaman Data Historis

d. Halaman Prediksi

Halaman prediksi merupakan menu paling utama dari aplikasi ini karena disini adalah menu untuk memprediksi dan untuk melihat hasil prediksi harga *ethereum* kedepannya. Pada halaman ini *user* dapat melakukan prediksi dengan cara

memasukkan jumlah hari yang diinginkan pada kolom input yang tersedia dan mengklik tombol prediksi untuk melakukan proses prediksi. *User* dapat langsung melihat hasil prediksi berupa tabel. Gambar halaman prediksi dapat dilihat pada gambar 9.



Gambar 9. Halaman Prediksi

### 3.2 Hasil Pengujian

Pada pengujian fungsional aplikasi ini dilakukan dengan menggunakan metode *black box testing* dimana dilakukan pengujian fungsi atau fitur dan aspek-aspek penting yang ada pada aplikasi. Hasil pengujian dapat dilihat pada table 1.

Tabel 1. Hasil Pengujian

No	Layout	Skenario Uji	Hasil Yang Diharapkan	Hasil
1	Tampilan Menu Utama	Memilih halaman <i>chart</i>	Dapat berpindah ke halaman <i>Chart</i> setelah menekan button halaman <i>Chart</i>	Berhasil
		Memilih halaman data historis	Dapat berpindah ke halaman data historis setelah menekan button halaman data historis	Berhasil
		Memilih halaman prediksi	Dapat berpindah ke halaman prediksi setelah menekan button halaman prediksi	Berhasil
		Memilih halaman beranda	Dapat kembali ke halaman menu utama atau beranda setelah menekan button halaman beranda	Berhasil
2	Tampilan Halaman Beranda	Melihat deskripsi <i>ethereum</i>	Dapat menampilkan deskripsi <i>ethereum</i>	Berhasil

3	Tampilan Halaman <i>Chart</i>	Melihat <i>chart ethereum</i>	Dapat menampilkan <i>chart</i> pergerakan harga <i>ethereum</i>	Berhasil
4	Tampilan Halaman Data Historis	Melihat dan mencari data historis <i>ethereum</i>	Dapat menampilkan dan mencari data historis harga <i>ethereum</i> sebelumnya	Berhasil
5	Tampilan Halaman Prediksi	Melakukan prediksi dan melihat hasilnya	Dapat melakukan input dan menampilkan hasil prediksi sesuai jumlah hari yang diinputkan oleh <i>user</i>	Berhasil

## 4. SIMPULAN

Dari hasil penelitian yang dilakukan dapat diambil kesimpulan sebagai berikut:

1. Metode Autoregressive Integrated Moving Average (ARIMA) ini dapat diimplementasikan pada sistem aplikasi untuk memprediksi harga ethereum kedepan.
2. Aplikasi ini dapat memprediksi harga ethereum sesuai dengan periode atau hari yang diinginkan oleh user.

## 5. SARAN

Diperoleh beberapa hal yang dapat dilakukan untuk pengembangan penelitian selanjutnya antara lain:

1. Penelitian selanjutnya dapat menambahkan fitur cetak data historis dan hasil prediksi yang telah dilakukan.
2. Penelitian selanjutnya dapat menggunakan perubahan data dengan frekuensi yang lebih cepat, seperti data perubahan ethereum dalam hitungan jam.

## DAFTAR PUSTAKA

- [1] Macedo, L. (2018). Blockchain for trade facilitation: Ethereum, eWTP, COs and regulatory issues. *World Custom Journal*, 87.
- [2] Hartati. 2017. Penggunaan Metode ARIMA dalam Meramal Pergerakan Inflasi. *Jurnal Matematika, Saint, dan Teknologi*. Vol.18, No.1, 1-10.
- [3] Hendrawan, B. 2012. Penerapan Model ARIMA Dalam Memprediksi IHSG. *Jurnal Integrasi*. Vol 4, No 2.
- [4] Makridakis, S., Wheelwright, S., & McGee, V. E. (1999). *Metode dan Aplikasi Peramalan*. Jakarta: Erlangga.

- [5] S. Utari and S. Setiawan “Sistem Penentuan Penerimaan Beasiswa Di SMA PGRI 4 Jakarta Timur” J. Infortech, vol. 3, no. 2, pp. 129 –135, 2021, doi: 10.31294/infortech.v3i2.11589.
- [6] Indrajani. 2011. Perancangan Basis Data dalam All in 1. Jakarta: PT. Elex Media Komputindo.
- [7] Sukamto, Rosa. A.S, & Shalahuddin, M. (2014). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung.
- [8] Nurdam, N. (2014). Sequence Diagram sebagai perkakas perancangan antarmuka pemakai. *Ultimatics: Jurnal Teknik Informatika*, 6(1), 21-25.
- [9] Dewi, L. P., Indahyanti, U., & Hari, Y. (2012). *Pemodelan proses bisnis menggunakan activity diagram uml dan bpmn (studi kasus frs online)* (Doctoral dissertation, Petra Christian University).
- [10] Putra, D. W. T., & Andriani, R. (2019). Unified modelling language (uml) dalam perancangan sistem informasi permohonan pembayaran restitusi sppd. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 7(1), 32-39.