Penerapan Algoritma *Bubble Sort* pada Lampu Lalu Lintas Dinamis

http://dx.doi.org/10.28932/jutisi.vXiX.X

Riwayat Artikel
Received: xx Bulan 20xx | *Final Revision*: xx Bulan 20xx | *Accepted*: xx Bulan 20xx

Rivaldo[1], Yohannes, M. Kom.[*2]

[#]*Program Studi Informatika, Universitas Multi Data Palembang*
Jln. Parameswara, Palembang, 30139, Indonesia
[1]Rivaldo@mhs.mdp.ac.id
[2]yohannesmasterous@mdp.ac.id
✉Corresponding author: Rivaldo4403@gmail.com

Abstrak — **Penelitian ini menyelidiki penerapan algoritma *Bubble Sort* pada sistem lampu lalu lintas dinamis sebagai upaya untuk meningkatkan efisiensi pengaturan lalu lintas di persimpangan jalan. Penggunaan *Bubble Sort* menawarkan pendekatan dengan melakukan pengurutan terhadap bobot kendaraan pada setiap jalur dan mengubah warna lampu merah menjadi hijau secara otomatis ketika hasil pengurutan menunjukkan bahwa jalur tersebut memiliki bobot kendaraan paling besar. Pendekatan ini menyediakan cara yang sederhana, efisien, dan efektif dalam mengatur persimpangan lampu lalu lintas untuk menghindari kemacetan di Indonesia. Hasil pengujian menggunakan simulasi menunjukkan bahwa penggunaan *Bubble Sort* pada sistem lampu lalu lintas dinamis berhasil mengatur persimpangan lampu lalu lintas dengan efektif dan efisien, terutama untuk kategori 3 jalur dan 4 jalur. Namun, perlu diperhatikan bahwa pada kategori 5 jalur dan 6 jalur, kendaraan dapat menumpuk di satu jalur karena jumlah jalur yang terlalu banyak, meskipun hal ini tidak mencerminkan kasus dunia nyata.**

Kata kunci— ***Bubble Sort*; efisiensi lalu lintas; lampu lalu lintas dinamis; pengaturan persimpangan; sistem transportasi.**

The Implementation of Bubble Sort Algorithm on Dynamic Traffic Lights

Abstract — ***This research investigates the implementation of the Bubble Sort algorithm in dynamic traffic light systems as an effort to enhance traffic regulation efficiency at road intersections. The use of Bubble Sort offers an approach by sorting the vehicle weights on each lane and automatically changing the red light to green when the sorting results indicate that the lane has the heaviest vehicle weight. This approach provides a simple, efficient, and effective way to regulate traffic light intersections to avoid congestion in Indonesia. Test results using simulations show that the use of Bubble Sort in dynamic traffic light systems successfully regulates traffic light intersections effectively and efficiently, especially for the categories of 3 lanes and 4 lanes. However, it should be noted that in categories of 5 lanes and 6 lanes, vehicles may accumulate in one lane due to the excessive number of lanes, although this does not reflect real-world cases.***

Keywords— ***Bubble Sort; traffic efficiency; dynamic traffic lights; intersection regulation; transportation system.***

I. LATAR BELAKANG

Lampu lalu lintas adalah alat yang digunakan pada persimpangan jalan untuk membantu mengatur lalu lintas kendaraan di persimpangan jalan. Semua peralatan pengatur lalu lintas yang menggunakan tenaga listrik kecuali lampu kedip, rambu, dan marka jalan untuk mengarahkan atau memperingatkan pengemudi kendaraan bermotor, pengendara sepeda atau pejalan kaki [1] . Fungsi utamanya adalah untuk memberikan arahan kepada pengendara tentang kapan mereka harus berhenti, melanjutkan, atau memperlambat laju kendaraan mereka, dengan tujuan utama meningkatkan keamanan dan efisiensi dalam pengaturan lalu lintas. Pengaturan arus lalu lintas pada persimpaangan pada dasarnya dimaksudkan untuk bagaimana pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan dapat bergerak secara pergantian sehingga tidak saling menggangu antar arus yang ada [2].

Lampu lalu lintas menjadi suatu alat yang digunakan untuk mencegah kemacetan di persimpangan, Kemacetan adalah keadaan di mana kendaraan mengalami berbagai jenis kendala yang mengakibatkan turunnya kecepatan kendaraan di bawah keadaan normal. Kemacetan akan sangat merugikan bagi para pengguna jalan, karena akan menghambat waktu perjalanan mereka [3]. Indonesia atau lebih tepatnya di Kota Jakarta termasuk kota peringkat 30 untuk waktu tempuh rata-rata per 10 km dari 387 di seluruh kota di dunia di tahun 2023 [4]. Ini membuktikan bahwa tingkat kemacetan di Indonesia sudah parah.

Namun, sekarang lampu lalu lintas yang merupakan pencegah kemacetan menjadi pusat kemacetan di Indonesia, peristiwa ini bisa terjadi karena beberapa faktor. Salah satunya menurut Herdiansyah, Muhammad Izman dan Atika, Linda Jumlah kendaraan mengalami peningkatan sejalan dengan pertumbuhan ekonomi, sementara disisi lain pertumbuhan infrastruktur lalu lintas tidak setinggi pertumbuhan jumlah kendaraan dan mobilitas masyarakat [5]. Ini menyebabkan jalan dimana lampu lalu lintas berada tidak bisa menampung kendaraan yang sudah semakin banyak karena tidak ada perluasan jalan. Penyebab lainnya disebabkan oleh tidak efektifnya lampu lalu lintas di Indonesia dalam mengatur arus lalu lintas. Menurut Jatmika, Sunu dan Andiko, Indra semakin meningkatnya volume lalu lintas pada suatu persimpangan jalan serta pengaturan nyala lampu lalu lintas yang sudah tidak sesuai lagi dengan tingkat kepadatan kendaraan. Maka menimbulkan berbagai permasalahan yang dapat menganggu kelancaran lalu lintas dan aktifitas masyarakat. Salah satu faktor yang menyebabkan terhambatnya lalu lintas adalah kemacetan [2]. Pernyataan tersebut membuktikan bahwa Pengaturan lampu lalu lintas di Indonesia sudah tidak efektif lagi.

Masalah ini bisa diselesaikan dengan menerapkan "Lampu Lalu Lintas Dinamis" menggunakan *Bubble Sort* , dimana menurut KBBI dinamis artinya penuh semangat dan tenaga sehingga cepat bergerak dan mudah menyesuaikan diri dengan keadaan dan sebagainya [6]. Dari pernyataan tersebut dapat digunakan untuk mengartikan Lampu Lalu Lintas Dinamis. Lampu Lalu Lintas Dinamis adalah alat yang digunakan pada persimpangan jalan untuk membantu mengatur lalu lintas kendaraan di persimpangan jalan dengan menyesuaikan diri dengan situasi. Situasi yang dimaksud disini adalah kepadatan suatu jalan yang diukur dari banyaknya kendaraan di jalan tersebut.

II. TUJUAN

Dari penjabaran masalah pada bagian latar belakang maka diperlukan suatu penelitian dengan tujuan untuk mengetahui seberapa efektif penggunaan algoritma pengurutan sederhana yaitu *Bubble Sort* terhadap kualitas pengaturan lampu lalu lintas di Indonesia sekarang dan apakah Penerapan Algoritma *Bubble Sort* pada Lampu Lalu Lintas Dinamis dapat diimplementasikan di persimpangan lampu lampu lalu di Indonesia. Maka penulis ingin mengangkat permasalahan mengenai tingkat efektivitas *Bubble Sort* untuk penerapan Lampu Lalu Lintas Dinamis berikut yang berjudul : "Penerapan Algoritma *Bubble Sort* pada Lampu Lalu Lintas Dinamis".

III. TINJAUAN PUSTAKA

A. *Penelitian terkait*

Pada penelitian ini terdapat beberapa tinjauan pustaka yang dapat menjadi kerangka dalam penelitian ini. Pertama terdapat penelitian yang serupa dengan judul penelitian yang peneliti pilih yaitu penerapan Lampu Lalu Lintas Dinamis sudah pernah dilakukan oleh Manurung, Octavia Yohana menggunakan *Adaptive Neuro-Fuzzy Inference System* (ANFIS) [7]. Penelitian ini berjudul Simulasi Pengaturan Lampu Lalu Lintas Dinamis Menggunakan *Adaptive Neuro-Fuzzy Inference System* (ANFIS). Juga terdapat penelitian yang serupa oleh Leal, Samara Soares dan Almeida, Paulo Eduardo M. de menggunakan *non-dominated sorting genetic algorithm* (NSGA2) dengan judul *Traffic light optimization using non-dominated sorting genetic algorithm* (NSGA2) dimana "Optimisasi rencana lampu lalu lintas ini menghasilkan kelancaran yang lebih besar bagi kendaraan, dan sebagai hasilnya, mengurangi kemacetan serta semua konsekuensi yang ditimbulkannya dalam kehidupan masyarakat." [8].

B. *Algoritma*

Pada penelitian ini digunakan algoritma, Menurut Ramadhon, Rafael Nuansa Algoritma adalah metode atau langkah yang direncanakan secara tersusun dan berurutan untuk menyelesaikan atau memecahkan permasalahan dengan

sebuah intruksi atau kegiatan. Pengertian ini biasanya digunakan untuk kegiatan sehari-hari [9]. Terdapat juga pengertian *Bubble Sort* dimana Menurut Hartanto,

C. *Bubble Sort*

Penelitian ini menggunakan *Bubble Sort*, *Bubble Sort* sendiri adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan/penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya [10].

D. *Flutter*

Pembuatan program penelitian ini menggunakan *Flutter*. *Flutter* adalah kerangka pengembangan untuk membangun aplikasi untuk platform mobile, web, dan desktop dari satu kode sumber. Sejak dirilis secara resmi oleh Google kurang dari beberapa tahun yang lalu, *Flutter* semakin populer di kalangan pengembang aplikasi mobile, bahkan dianggap sebagai perubahan paradigma [11]. Melakukan lebih banyak dengan simbol yang lebih sedikit, bahkan pada sebuah OS tunggal, dengan pemrograman yang mutakhir dan ekspresif, serta metodologi yang definitive [12]. Penggunaan *Flutter* sangatlah efisien karena kita dapat melakukan *coding* sedikit dan menghasilkan sesuatu yang lebih dari framework lain.

E. *Dart*

Bahasa yang digunakan untuk membuat program ini ialah menggunakan dart. *Dart* sendiri merupakan Bahasa program yang digunakan oleh *Flutter*. *Dart* adalah bahasa pemrograman yang digunakan dalam aplikasi *Flutter* untuk membangun antarmuka lokal berkualitas tinggi pada perangkat Android dan iOS [12]. Bahasa pemrograman *Dart* adalah bahasa pemrograman yang produktif, mudah dipelajari, dan efisien dengan semantik yang jelas dan dukungan untuk eksekusi bertahap serta startup aplikasi yang cepat [13].

F. *VSCode (Visual Studio Code)*

VSCode adalah IDE yang digunakan untuk membuat program pada penelitian ini. *Visual Studio Code* adalah lingkungan pengembangan berorientasi kode yang kuat yang menyederhanakan penulisan aplikasi web, mobile, dan cloud menggunakan berbagai bahasa pemrograman di berbagai platform. *Visual Studio Code* juga mendukung siklus pengembangan aplikasi dengan debugger bawaan dan dukungan terintegrasi untuk kontrol versi Git [14].

IV. METODOLOGI

A. *Data*

Data yang digunakan pada penelitian ini merupakan data yang dibuat sendiri untuk melakukan simulasi terhadap persimpangan lampu lalu lintas. Data tersebut disimpan didalam class “Lane” di Gambar 1.

```dart
class Lane {
  int id;
  double cars;
  double trucks;
  double motorcycles;

  Color? color;

  Lane({
    required this.id,
    required this.cars,
    required this.trucks,
    required this.motorcycles,
    this.color,
  });

  double get vehiclesweight => cars * 1.0 + trucks * 2.0 + motorcycles * 0.5;
}
```

Gambar 1. Class Lane

Berikut adalah penjelasan mengenai variabel data yang terdapat di class lane :

1. *Id* : Variabel *Id* digunakan untuk menampilkan jenis jalan atau nomor identifikasi jalur, jika terdapat 4 jalan maka ID lane akan terdapat 4.
2. *Cars* : Variabel *Cars* digunakan untuk menampilkan berapa jumlah mobil yang ada di jalur tersebut.

3. *Trucks* : Variabel *Trucks* digunakan untuk menampilkan berapa jumlah truk yang ada di jalur tersebut
4. *Motorcycles* : Variabel *Motorcycles* digunakan untuk menampilkan berapa jumlah motor yang terdapat di jalur tersebut
5. *Color* : Variabel Color digunakan untuk menampilkan warna lampu lalu lintas jalur tersebut apakah berwarna merah atau hijau
6. *Vehicleweight* : Variabel *Vehicleweight* merupakan properti yang dihitung untuk menunjukkan bobot dari suatu jalur, dihitung berdasarkan jumlah mobil dengan bobot 1, truk dengan bobot 2, dan sepeda motor dengan bobot 0,5 yang ada di jalur tersebut. Properti ini digunakan untuk menentukan prioritas pengaturan lampu lalu lintas. Rumus variabel ini terdapat pada Gambar 1.

Data yang terdapat di class tersebut dapat diubah pada program menyebabkan pengaturan data di dalam simulasi persimpangan lampu lalu lintas dapat dilakukan tanpa khawatir. Pengaturan data ditunjukkan pada Gambar 2 dan Gambar 3 berikut.

```
List<Lane>  lanes = [
        Lane(id: 1, cars: 4, trucks: 1, motorcycles: 2.0),
        Lane(id: 2, cars: 2, trucks: 3, motorcycles: 1.0),
        Lane(id: 3, cars: 1, trucks: 2, motorcycles: 7.0),
        Lane(id: 4, cars: 3, trucks: 5, motorcycles: 0.0)
      ];
```

Gambar 2. Tempat pengaturan data dalam class Lane pertama

```
lanes = [
  Lane(id: 1, cars: 4, trucks: 1, motorcycles: 2.0),
  Lane(id: 2, cars: 2, trucks: 3, motorcycles: 1.0),
  Lane(id: 3, cars: 1, trucks: 2, motorcycles: 7.0),
  Lane(id: 4, cars: 3, trucks: 5, motorcycles: 0.0)
];
```

Gambar 3. Tempat pengaturan data dalam class Lane kedua

Dari Gambar 2 dan Gambar 3 kita dapat melihat bahwa terdapat array list dimana kita dapat mengatur data yang akan kita simulasikan di sebuah persimpangan lampu lalu lintas kedua array list ini sama dan harus diganti secara bersamaan jika mau mengganti jumlah data yang terdapat di class Lane. Kode pada Gambar 2 terdapat di class _MyHomePageState sedangkan Kode pada Gambar 3 terdapat di function *_reset*Simulation(). Data-data ini juga tidaklah statis melainkan dinamis, dimana data pada class Lane akan bertambah dan berkurang sendiri tergantung dari kondisi yang nantinya akan dijelaskan di bagian B yaitu Algoritma.

B. *Algoritma*

Adapun cara kerja dari algoritma dari penelitian ini antara lain :

1. Inisialisasi Data Jalur

   Pada awalnya, terdapat inisialisasi data jalur dalam simulasi, yang mewakili jumlah kendaraan di setiap jalur serta bobotnya. Data ini disimpan dalam objek-objek Lane.
2. Sortir Jalur Secara Descending

   Setelah menekan tombol *start*, maka simulasi persimpangan lampu lalu lintas akan dimulai, pertama data-data tersebut diurutkan berdasarkan bobot kendaraan menggunakan algoritma *Bubble Sort* yang telah diimplementasikan dalam fungsi BubbleSortLane(). Proses sortir dilakukan secara terus menerus sampai menekan tombol *reset* dan digunakan untuk memprioritaskan pengaturan lampu lalu lintas berdasarkan bobot kendaraan yang melewati setiap jalur.
3. Pengaturan Lampu Lalu Lintas

   Lampu lalu lintas dinyalakan dan dimatikan berdasarkan kondisi yang ditentukan, yaitu berdasarkan prioritas jalur dengan bobot kendaraan tertinggi. Ketika lampu hijau menyala pada suatu jalur, kendaraan dapat melintasinya, sementara jalur lainnya akan diberi lampu merah. Proses pengaturan lampu lalu lintas ini dilakukan secara dinamis selama simulasi berlangsung.

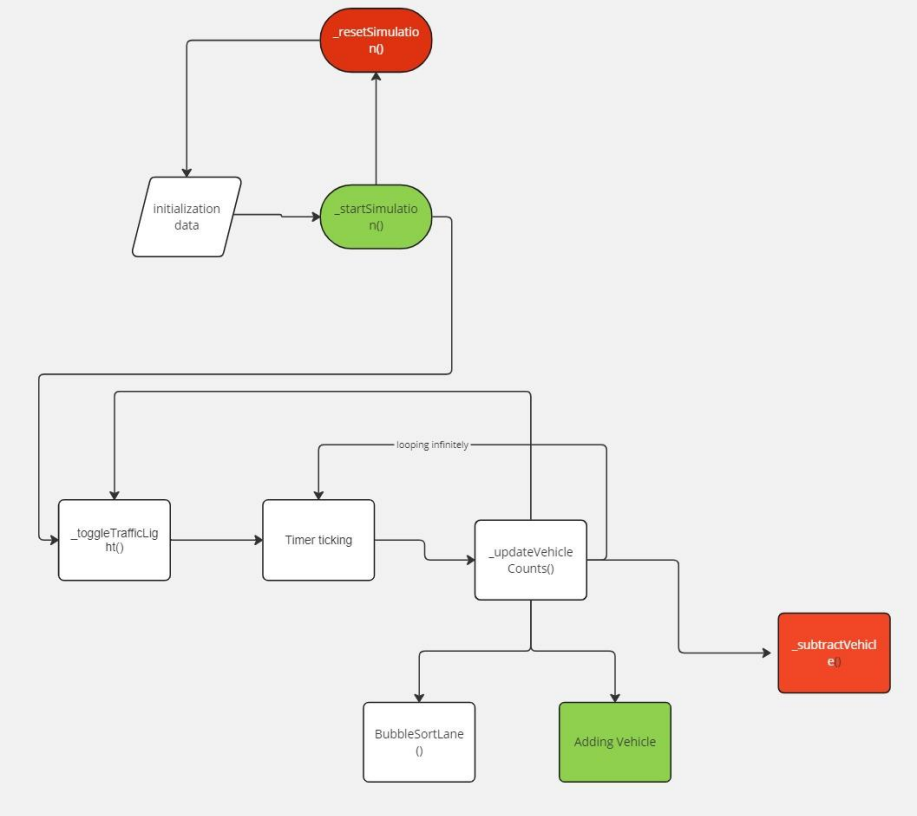
4. Pembaruan Jumlah Kendaraan

Selama simulasi berlangsung, jumlah kendaraan di setiap jalur akan terus diperbarui. Kendaraan baru ditambahkan ke setiap jalur secara acak per detiknya, sementara kendaraan yang melewati jalur dengan lampu hijau akan dikurangi berdasarkan aturan yang mencerminkan dunia nyata, dimana jika bobot kendaraan atau *vehicleweight* kurang dari sama dengan 10 maka jumlah kendaraan akan berkurang 1, jika bobot kendaraan lebih dari 10 maka kendaraan akan berkurang 2, dan terakhir jika bobot kendaraan lebih dari 30 maka jumlah kendaraan akan berkurang 3. Pengurangan dan penambahan akan dilakukan secara acak sehingga program akan mengambil salah satu dari 3 jenis kendaraan yaitu mobil, truk, atau motor. Pengurangan dan penambahan kendaraan akan mempengaruhi bobot kendaraan yang dihitung dengan rumus vehicleweight pada Gambar 1. Pengurangan jumlah kendaraan ini akan dilakukan tiap detik sampai waktu lampu hijau yaitu 10 detik habis atau bobot kendaraan yang terdapat di jalur adalah 0 atau kurang dari 0.

5. Data ditampilkan ke UI

Simulasi lampu lalu lintas dan pembaruan jumlah kendaraan ditampilkan secara grafis dalam antarmuka pengguna. Pengguna dapat memulai atau menghentikan simulasi, serta melihat status warna lampu lalu lintas, bobot kendaraan di setiap jalur dan jumlah setiap jenis kendaraan di setiap jalur.

C. *Rancangan Program*



Gambar 4 Rancangan program dalam bentuk flowchart yang disederhanakan

Proses pengembangan kode dilakukan menggunakan Integrated Development Environment (IDE) *Visual Studio Code (VSCode)*. *VSCode* memberikan lingkungan pengembangan yang kuat dan fleksibel untuk menulis, mengedit, dan mengelola kode program secara efisien. Framework yang digunakan untuk membangun antarmuka pengguna program pada penelitian ini adalah *Flutter*. *Flutter* adalah framework UI open-source yang dikembangkan oleh Google untuk membangun antarmuka pengguna yang kaya dan dinamis secara lintas platform. Dengan *Flutter*, pengembang dapat membuat aplikasi yang indah dan responsif untuk berbagai platform, termasuk Android, iOS, dan web. Terakhir, Bahasa pemrograman yang digunakan dalam pengembangan program ini adalah *Dart*. *Dart* adalah bahasa pemrograman yang dioptimalkan untuk pengembangan aplikasi dengan *Flutter*. *Dart* menawarkan sintaksis yang bersih, kinerja yang tinggi, serta dukungan untuk pemrograman berorientasi objek dan fungsional. Program yang dibuat terdapat di (link github)

V. IMPLEMENTASI

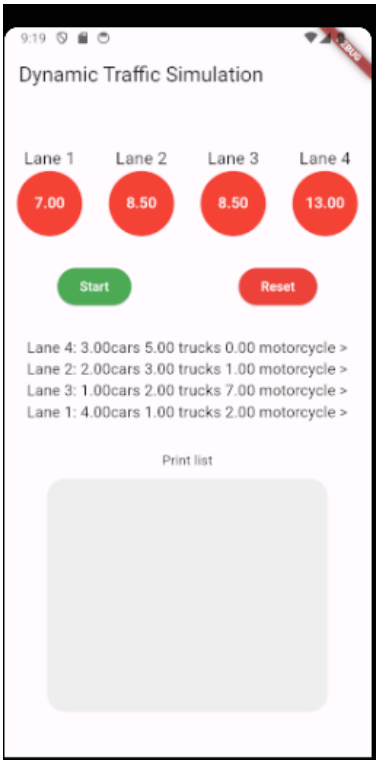
Untuk Implementasi dari Program digunakan Algoritma *Bubble Sort* secara *descending* yang menjadi suatu fokus utama dalam penelitian ini, Selain *Bubble Sort* terdapat juga Algoritma untuk menjalankan simulasinya menggunakan *timer*, penambahan, dan pengurangan kendaraan. Dimana kode-kode tersebut menggunakan sintaks *if* dan *for*.

A. *Input dan Output Program*

Untuk input dalam program dilakukan bukan dengan memasukkan input di UI program melainkan dilakukan dengan mengatur inisialisasi input pada Gambar 2, Gambar 3, dan juga greenLightDuration dimana kedua nilai pada Gambar 2 dan Gambar 3 harus sama sedangkan untuk greenLightDuration dapat diatur mau berapa detik lampu hijaunya akan menyala jika kondisi terpenuhi. Jika nilai pada Gambar 2 dan Gambar 3 berbeda maka nilai inisialisasi pada saat menjalankan program dan menekan tombol *reset* pada program akan berbeda, tetapi fungsionalitas dari program masih tetap berfungsi dengan baik.

Sedangkan output pada program adalah macam-macam UI yang menampilkan status lampu lalu lintas (hijau atau merah) dengan bobot kendaraan setiap jalur yang menampilkan pengurangan dan penambahan bobot tiap detik. Juga terdapat tampilan UI yang menampilkan jumlah kendaraan pada setiap jalur, penambahan dan pengurangan kendaraan serta hasil sorting jalur tiap detik. Terdapat pula daftar pernyataan yang mencatat perubahan status warna lampu lalu lintas.

B. *Tampilan Program (GUI)*



Gambar 5 Tampilan GUI program

Untuk tampilan GUI program dapat dilihat pada Gambar 5 dimana di paling atas tampilan UI terdapat bobot kendaraan pada setiap jalur beserta UI warna lampu pada jalur tersebut. Setelah itu terdapat tombol *start* dan *reset* untuk menjalankan dan menghentikan output dari program. Lalu terdapat UI untuk melihat nilai setiap jenis kendaraan pada jalur tersebut serta urutan ke berapa jalur tersebut dari yang terbesar sampai terkecil. Dimana jika lampu jalur tersebut berwarna hijau maka Tulisan di jalur tersebut akan berubah ke warna merah karena terjadinya pengurangan kendaraan. Terakhir terdapat list daftar untuk menampilkan status program apakah menekan tombol *start*, menampilkan peristiwa pergantian warna pada lampu lalu lintas di salah satu jalur, dan apakah menekan tombol *reset*.

C. *Penjelasan Cara Penggunaan*

Cara menggunakan programnya sangatlah mudah, pertama pastikan inisialisasi jumlah kendaraan pada setiap jalur sudah benar pada Gambar 2 dan Gambar 3. Jika sudah jalankan program dengan menekan run pada IDE *VSCode*. Setelah UI muncul maka tekan tombol *start* untuk menjalankan program. Setelah itu tinggal lihat hasil outputnya yang akan berlangsung sampai menekan tombol *reset*. Jika sudah selesai melihat proses simulasi lampu lalu lintas dinamis menggunakan *Bubble Sort* maka tekan tombol *reset* untuk mengulang prosesnya. Jika mau mengganti jumlah kendaraan tinggal mengaturnya di kode program tepatnya seperti pada Gambar 2 dan Gambar 3.

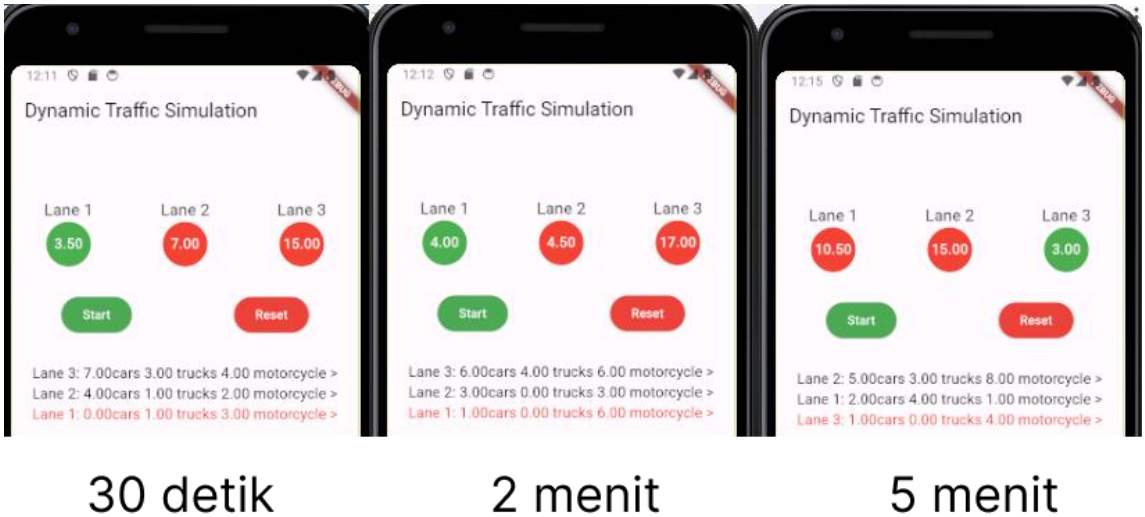
VI. PENGUJIAN

Pengujian dalam penelitian ini dilakukan dengan menyimulasikan pengaturan persimpangan lampu lalu lintas dengan menetapkan jumlah jalur yang akan disimulasikan dari 3 hingga 6 jalur dan nilai setiap jalur akan dibuat sama-sama 0. Efektivitas penerapan *Bubble Sort* pada lampu lalu lintas dinamis diukur dari rata-rata kecepatan penyortiran dari 10 kali penyortiran yang diukur dalam *microseconds* yang seperjuta dari satu detik dan kemampuan *Bubble Sort* untuk mengatur persimpangan pada detik 30, menit 2, dan menit 5. Jika terdapat 3 jalur yang memiliki jumlah kendaraan melebihi 50, maka *Bubble Sort* dianggap tidak efektif untuk mengatur jumlah jalur tersebut dan persimpangan lalu lintas tersebut akan dianggap sudah macet dan gagal untuk diatur. Berikut adalah ringkasan hasil pengujian untuk setiap jumlah jalur yang akan disajikan dalam bentuk tabel.

Tabel 1 hasil pengujian dari 3 jalur hingga 6 jalur

No	Jumlah jalur	Rata-rata kecepatan sorting	Berapa jalur yang melewati 50 bobot kendaraan		
			30 detik	2 menit	5 menit
1	3 jalur	40,3 microseconds	0	0	0
2	4 jalur	51,8 microseconds	0	0	0
3	5 jalur	104,2 microseconds	0	1	0
4	6 jalur	131,2 microseconds	0	1	2

Dari pengujian pada 3 jalur sampai 6 jalur, didapat data-data pada Tabel 1 yang dapat digunakan untuk analisis dan menyimpulkan hasil dari penerapan *Bubble Sort* terhadap lampu lalu lintas dinamis. Hasil yang didapatkan sangat memuaskan karena dengan data tersebut pertanyaan dari penelitian ini dapat dijawab.

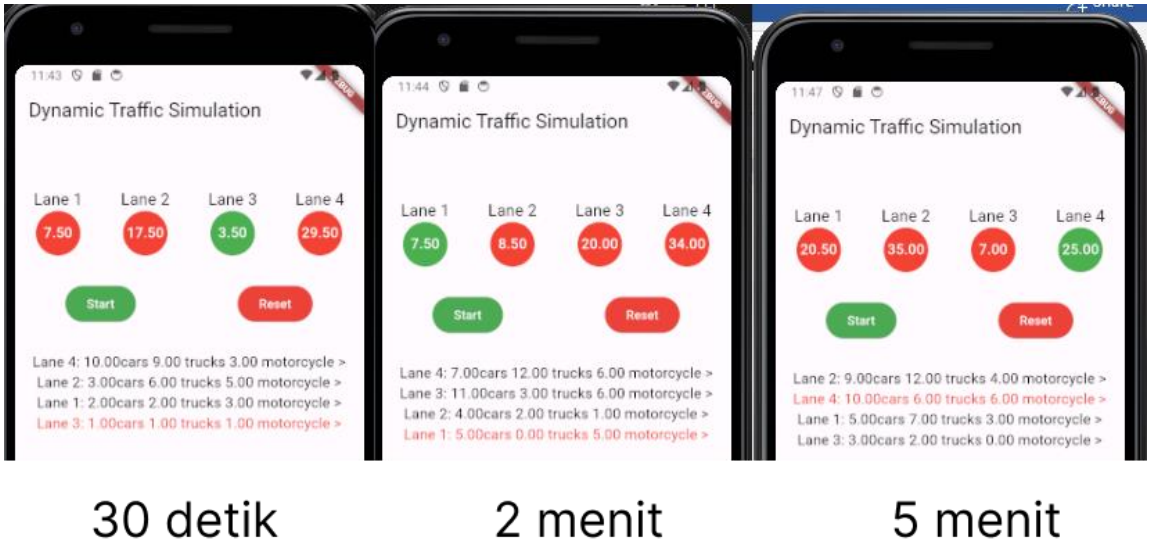


Gambar 6 hasil dari pengujian 3 jalur

Dari hasil pengujian menggunakan simulasi persimpangan lampu lalu lintas dengan 3 jalur pada Gambar 6 didapatkan hasil sebagai berikut yang akan disajikan dalam bentuk Tabel 2.

No	Waktu	Jalur	Jumlah Mobil	Jumlah Truk	Jumlah Motor	Bobot Kendaraan
1	30 detik	Ke-1	0	1	3	3,5
		Ke-2	4	1	2	7
		Ke-3	7	3	4	15
2	2 menit	Ke-1	1	0	6	4
		Ke-2	3	0	3	4,5
		Ke-3	6	4	6	17
3	5 menit	Ke-1	2	4	1	10,5
		Ke-2	5	3	8	15
		Ke-3	1	0	4	3

Tabel 2 hasil pengujian pada persimpangan lampu lalu lintas 3 jalur

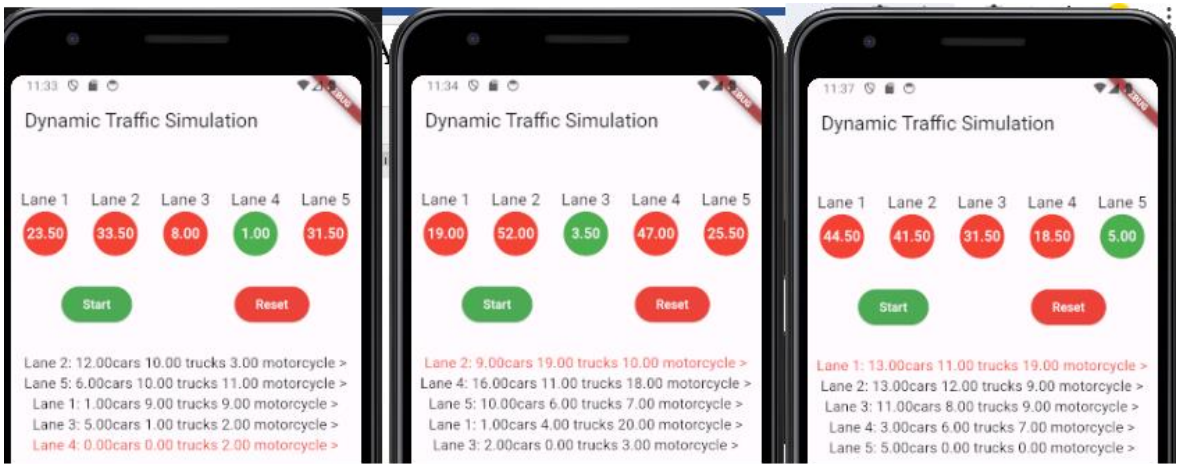


Gambar 7 hasil dari pengujian 4 jalur

Pada simulasi persimpangan lampu lalu lintas dengan 4 jalur pada Gambar 7 didapatkan hasil sebagai berikut yang akan disajikan dalam bentuk tabel pada Tabel 3.

Tabel 3 hasil pengujian pada persimpangan lampu lalu lintas 4 jalur

No	Waktu	Jalur	Jumlah Mobil	Jumlah Truk	Jumlah Motor	Bobot Kendaraan
1	30 detik	Ke-1	2	2	3	7,5
		Ke-2	3	6	5	17.5
		Ke-3	1	1	1	3.5
		Ke-4	10	9	3	29.5
2	2 menit	Ke-1	5	0	5	7.5
		Ke-2	4	2	1	8,5
		Ke-3	11	3	6	20
		Ke-4	7	12	6	34
3	5 menit	Ke-1	5	7	3	20.5
		Ke-2	9	12	4	35
		Ke-3	3	2	0	7
		Ke-4	10	6	6	25

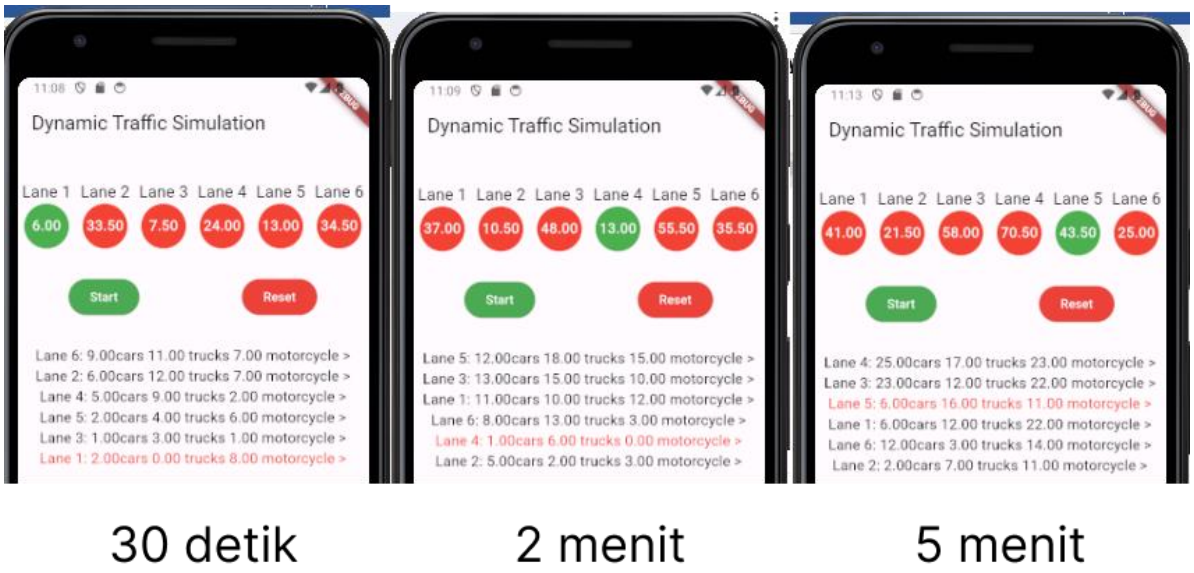


Gambar 8 hasil dari pengujian 5 jalur

Setelah itu dilanjutkan dengan simulasi persimpangan lampu lalu lintas dengan 5 jalur pada Gambar 8 didapatkan hasil sebagai berikut yang akan disajikan pada Tabel 4.

Tabel 4 hasil pengujian pada persimpangan lampu lalu lintas 5 jalur

No	Waktu	Jalur	Jumlah Mobil	Jumlah Truk	Jumlah Motor	Bobot Kendaraan
1	30 detik	Ke-1	1	9	9	23,5
		Ke-2	12	10	3	33,5
		Ke-3	5	1	2	8
		Ke-4	0	0	2	1
		Ke-5	6	10	11	31,5
2	2 menit	Ke-1	1	4	20	19
		Ke-2	9	19	10	52
		Ke-3	2	0	3	3,5
		Ke-4	16	11	18	47
		Ke-5	10	6	7	25
3	5 menit	Ke-1	13	11	19	44,5
		Ke-2	13	12	9	41,5
		Ke-3	11	8	9	31,5
		Ke-4	3	6	7	18,5
		Ke-5	5	0	0	5



Gambar 9 hasil dari pengujian 6 jalur

Pengujian terakhir menggunakan simulasi persimpangan lampu lalu lintas dengan 6 jalur pada Gambar 9 didapatkan hasil sebagai berikut yang akan disajikan di Tabel 5.

Tabel 5 hasil pengujian pada persimpangan lampu lalu lintas 6 jalur

No	Waktu	Jalur	Jumlah Mobil	Jumlah Truk	Jumlah Motor	Bobot Kendaraan
1	30 detik	Ke-1	2	0	8	6
		Ke-2	6	12	7	33,5
		Ke-3	1	3	1	7,5
		Ke-4	5	9	2	24
		Ke-5	2	4	6	13
		Ke-6	9	11	7	6
2	2 menit	Ke-1	11	10	12	37
		Ke-2	5	2	3	10.5
		Ke-3	13	15	10	48
		Ke-4	1	6	0	13
		Ke-5	12	18	15	55,5
		Ke-6	8	13	3	35,5
3	5 menit	Ke-1	6	12	22	41
		Ke-2	2	7	11	21,5
		Ke-3	23	12	22	58
		Ke-4	25	17	23	70,5
		Ke-5	6	16	11	43,5
		Ke-6	12	3	14	25

VII. ANALISIS

Dari pengujian yang sudah dilakukan, didapatkan hasil yang memuaskan untuk simulasi persimpangan lampu lalu lintas dengan 3 jalur sampai 6 jalur. Pada Tabel 1 dapat dilihat bahwa rata-rata kecepatan penyortiran sangatlah cepat berkisar dari 40,3 *microseconds* pada kategori 3 jalur, 51,8 *microseconds* pada kategori 4 jalur, 104,2 *microseconds* pada kategori 5 jalur, dan 131,2 *microseconds* pada kategori 6 jalur. Kecepatan sorting yang dihasilkan menggunakan *Bubble Sort* sangat cepat, ini menyebabkan *program* berjalan dengan lancar dan cepat. Ini juga membuktikan bahwa *Bubble Sort* dapat diterapkan di dunia nyata pada lampu lalu lintas di persimpangan.

Hasil pengujian juga membuktikan bahwa *Bubble Sort* dapat mengatur semua kategori jalur dengan baik dari 3 jalur hingga 6 jalur. Ini didapatkan dari Tabel 1 dimana tidak ada satupun kategori jalur yang 3 jalurnya melewati 50 bobot kendaraan. Namun, pada kategori 6 jalur pada Gambar 9 bisa dilihat bahwa pada menit ke-5 bobot kendaraan jalur ke44

sampai melewati 70 bobot kendaraan dimana dari bobot tersebut mencerminkan bahwa jalur tersebut terdapat banyak kendaraan yang bisa dibilang sudah termasuk macet. Dari situlah *Bubble Sort* langsung menyatakan bahwa jalur ke-4 adalah jalur dengan bobot kendaraan yang terbesar dan mengubah warna pada jalur tersebut menjadi hijau sehingga mengurangi jumlah kendaraan di jalan tersebut. Ini adalah tujuan dari penerapan *Bubble Sort* pada lampu lalu lintas menjadikan lampu lalu lintas tersebut dari statis dimana pengaturan dilakukan secara berurutan menjadi dinamis dimana pengurutan dilakukan berdasarkan besarnya nilai bobot kendaraan. Sehingga pengaturan persimpangan lalu lintas dapat dilakukan secara efisien dan efektif serta menghindari kemacetan di lampu lalu lintas.

Namun, bisa dilihat pada kategori 5 jalur dan 6 jalur bisa dilihat bahwa program lampu lalu lintas dinamis sudah kewalahan untuk mengatur kategori-kategori jalur tersebut pada menit ke-5, pada jalur ke-5 terdapat dua lane dengan nilai yang mendekati 50 yaitu 44,5 dalam jalur ke-1 dan 41,5 pada jalur ke-2, sedangkan jalur 6 terdapat 2 jalur yang melewati 40 dan sudah melewati 50 yaitu 41 pada jalur ke-1, 43,5 pada jalur ke-2, 58 pada jalur ke-3, dan 70,5 pada jalur ke 4. Ini sudah bisa dibilang bahwa persimpangan lampu lalu lintas tersebut sudah mau mengalami kemacetan, penyebab masalahnya bukan terdapat pada Algoritma *Bubble Sort* melainkan terdapat pada program lampu hijau yang masih belum cukup untuk mengurangi jumlah kendaraan yang terdapat di suatu jalur, ini karena semakin bertambahnya jalur maka semakin banyak kendaraan yang masuk ke jalur tersebut tiap detiknya sehingga jika pada menit ke-5 pada kategori jalur ke-5 dan ke-6 jumlah kendaraan akan menumpuk dan 10 detik tidak cukup untuk mengurangi semua kendaraan pada lampu merah.

Itulah salah satu kelemahan dari program yang dibuat untuk penelitian ini, tetapi kasus persimpangan lampu lalu lintas yang terdapat 5 jalur dan 6 jalur tidak mencerminkan keadaan di dunia nyata. Sehingga kita dapat menyimpulkan bahwa sebenarnya program untuk penelitian penerapan algoritma *Bubble Sort* pada lampu lalu lintas dinamis sudah berhasil karena tidak ada persimpangan lampu lalu lintas yang terdapat 5 jalur dan 6 jalur apalagi selebihnya. Biasanya jika terdapat 5 jalur dan 6 jalur maka akan dibuat persimpangan dengan bentuk melingkar seperti pada Gambar 10 berikut ini.

Gambar 10. Bundaran HI di kawasan Menteng Jakarta Pusat
Sumber : Diadaptasi dari [15]

Bisa dilihat salah satu contoh pada Gambar 10, terbukti bahwa jika terdapat persimpangan dengan 5 jalur, 6 jalur, atau selebihnya maka tidak diimplementasikan lampu lalu lintas. Sehingga tidak diperlukan lampu lalu lintas dinamis, artinya penerapan Algoritma *Bubble Sort* pada lampu lalu lintas dinamis sangat efektif untuk menggantikan implemetasi lampu lalu lintas di persimpangan negara Indonesia.

VIII. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Pada penelitian ini program lampu lalu lintas dinamis menggunakan *Bubble Sort* menghasilkan kecepatan yang sangat baik berkisar antara 40 sampai 132 microseconds yang seperjuta dari 1. Ini sangatlah cepat dan sangat cocok untuk kasus pada persimpangan lalu lintas dimana kecepatan adalah kunci untuk mengatasi kemacetan. *Bubble Sort* sukses untuk mengatur persimpangan dengan kategori 3 jalur dan 4 jalur, sedangkan pada kategori 5 jalur dan 6 jalur mendapatkan hasil yang kurang memuaskan, tetapi jangan khawatir karena persimpangan lampu lalu lintas pada 5 jalur dan 6 jalur tidak mencerminkan kasus dunia nyata. Sehingga bisa dikatakan bahwa Penerapan Algoritma *Bubble Sort* pada Lampu Lalu Lintas Dinamis dapat diterapkan di persimpangan lampu lalu lintas di Indonesia, artinya tujuan penelitian ini sudah tercapai.

B. *Saran*

Penelitian ini belum sempurna karena masih menggunakan algoritma sederhana yaitu *Bubble Sort*. Selanjutnya peneliti akan mengembangkan aplikasi dengan campuran Algoritma Sorting yang lain atau dengan Algoritma yang lebih Advance. Sebaiknya juga program pada penelitian ini lebih membuat function lampu hijau menjadi lebih dinamis sehingga program dapat lebih efektif. Juga sebaiknya penulisan jurnal dapat ditingkatkan menjadi lebih baik lagi.

IX. DAFTAR PUSTAKA

[1] "BAB II TINJAUAN PUSTAKA Lampu lalu lintas." Accessed: May 26, 2024. [Online]. Available: https://e-journal.uajy.ac.id/3387/3/2TS09707.pdf

[2] S. Jatmika and I. Andiko, "SIMULASI PENGATURAN LAMPU LALU LINTAS BERDASARKAN DATA IMAGE PROCESSING KEPADATAN KENDARAAN BERBASIS MIKROKONTROLER ATMEGA16," *J. Tek. ITS*, vol. 7, no. 2, pp. 81–96, 2019, doi: 10.12962/j23373539.v7i2.35059.

[3] "BAB 2 LANDASAN TEORI Kemacetan." Accessed: May 26, 2024. [Online]. Available: http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2011-2-00100-TI Bab2001.pdf

[4] TomTom, "TOMTOM TRAFFIC INDEX Ranking 2023." Accessed: May 26, 2024. [Online]. Available: https://www.tomtom.com/traffic-index/ranking/

[5] M. I. Herdiansyah and L. Atika, "PENGATURAN LAMPU LALU LINTAS MENGGUNAKAN PENDEKATAN SISTEM PAKAR," *J. Ilm. Matrik*, vol. 18, no. 3, pp. 241–250, 2016, [Online]. Available: https://media.neliti.com/media/publications/224979-pengaturan-lampu-lalu-lintas-menggunakan-c5ea028e.pdf

[6] Kamus Besar Bahasa Indonesia (KBBI), "Pengertian kata dinamis." Accessed: May 26, 2024. [Online]. Available: https://kbbi.web.id/dinamis#google_vignette

[7] O. Y. Manurung, "Simulasi Pengaturan Lampu Lalu Lintas Dinamis Menggunakan Adaptive Neuro-Fuzzy Inference System (ANFIS)." Accessed: May 26, 2024. [Online]. Available: https://repositori.usu.ac.id/handle/123456789/4332

[8] S. S. Leal and P. E. M. de Almeida, "Traffic light optimization using non-dominated sorting genetic algorithm (NSGA2)," *Sci. Rep.*, vol. 13, no. 1, pp. 1–10, 2023, doi: 10.1038/s41598-023-38884-2.

[9] W. Hartanto, "Implementasi Algoritma Bubble Sort dengan Bahasa Pemrograman Python." Accessed: May 26, 2024. [Online]. Available: https://binus.ac.id/bandung/2023/10/implementasi-algoritma-bubble-sort-dengan-bahasa-pemrograman-python/

[10] R. N. Ramadhon, "Pengertian Algoritma." Accessed: May 26, 2023. [Online]. Available: https://www.unida.ac.id/teknologi/artikel/pengertian-algoritma.html

[11] Y. Cheon and C. Chavez, "Converting Android Native Apps to Flutter Cross-Platform Apps," *Proc. - 2021 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2021*, pp. 1898–1904, 2021, doi: 10.1109/CSCI54926.2021.00355.

[12] G. Idan Arb and K. Al-Majdi, "A Freights Status Management System Based on Dart and Flutter Programming Language," *J. Phys. Conf. Ser.*, vol. 1530, no. 1, 2020, doi: 10.1088/1742-6596/1530/1/012020.

[13] L. Bak, "How Dart learned from past object-oriented systems (keynote)," pp. 3–3, Oct. 2015, doi: 10.1145/2814189.2833212.

[14] A. Del Sole, "Introducing Visual Studio Code," *Vis. Stud. Code Distill.*, pp. 1–17, 2019, doi: 10.1007/978-1-4842-4224-7_1.

[15] M. Y. Laksono and H. B. Alexande, "Kawasan Bundaran HI Dipercantik dengan Konsep Jejak Bangsa di Gerbang Indonesia." Accessed: May 28, 2024. [Online]. Available: https://www.kompas.com/properti/read/2022/02/04/140000221/kawasan-bundaran-hi-dipercantik-dengan-konsep-jejak-bangsa-di-gerbang?page=all