

DIALIZER: APLIKASI SIMULASI DIAGRAM ALIR BERBASIS WEB

Danil Hendra Suryawan, Retno Mumpuni, Afina Lina Nurlaili
Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur
Jl. Raya Rungkut Madya, Gunung Anyar, Surabaya, Indonesia
danilhendrasr@gmail.com

ABSTRAK

Platform web yang pada mulanya hanya ditujukan untuk menampilkan dokumen kini telah berevolusi menjadi platform yang cukup mumpuni untuk menjalankan aplikasi interaktif. Salah satu keunggulan platform web adalah sifatnya yang lebih mudah diakses dibandingkan platform lain seperti platform desktop. Kendati demikian, sebagian aplikasi masih terdistribusikan hanya di platform desktop saja. Salah satu contohnya adalah RAPTOR yang merupakan aplikasi berbasis desktop yang banyak digunakan untuk mensimulasikan diagram alir. Karena sifatnya yang merupakan aplikasi berbasis desktop, RAPTOR tidak memiliki keunggulan-keunggulan aplikasi berbasis web seperti kemudahan akses. Melalui penelitian ini, kami bertujuan membuat sebuah aplikasi simulasi diagram alir yang lebih mudah diakses dengan berbasis web. Aplikasi ini yang kami namakan Dializer. Pengujian aplikasi dilakukan dengan metode *black box* dan survei pengguna. Hasil pengujian *black box* menunjukkan bahwa Dializer telah memiliki perilaku yang sesuai dengan ekspektasi untuk keseluruhan skenario yang diujikan. Sedangkan hasil survei pengguna dengan peramban internet, sistem operasi, serta kecepatan internet yang beragam memperlihatkan bahwa 82% responden menyatakan Dializer lebih mudah diakses dibandingkan solusi yang sudah ada, 94% responden menyatakan Dializer dapat dimuat dengan cepat, 85% menyatakan Dializer dapat mensimulasikan diagram alir dengan akurat, serta 80% responden merespon positif terkait kemungkinan mereka lebih memilih Dializer dibandingkan RAPTOR untuk memvisualisasikan diagram alir.

Kata kunci : *Dializer, aplikasi berbasis web, simulasi diagram alir*

1. PENDAHULUAN

Platform web yang pada mulanya hanya ditujukan untuk menampilkan dokumen [1] kini telah berevolusi hingga menjadi platform yang cukup mumpuni untuk digunakan menjalankan aplikasi interaktif. Yuxiao Wang dkk. [2] membuktikan bahwa platform web dapat digunakan untuk memvisualisasikan topologi jaringan 3 dimensi berskala besar dengan beragam interaksi. Robert Buels dkk. juga telah membuktikan bahwa platform ini mampu untuk digunakan untuk memfasilitasi keperluan bidang riset biologi khususnya dalam hal visualisasi data genome DNA [3].

Dengan makin mumpuninya platform web, sebagian peneliti telah memprediksi bahwa akan makin banyak aplikasi yang beralih ke platform ini dikarenakan keunggulan-keunggulan yang tidak dimiliki platform desktop [4], yaitu tidak memerlukan instalasi dan/atau peningkatan versi secara manual dari sisi pengguna (lebih mudah diakses), pen-deploy-an versi baru yang lebih mudah dengan cakupan pengguna yang lebih luas, serta format aplikasi terbuka yang memudahkan konten-konten yang tersedia di web untuk dapat dibaca oleh manusia dan cukup mudah untuk diproses oleh mesin, sehingga memicu berkembangnya jenis aplikasi baru yang mengagregasi data dari berbagai situs web.

Kendati demikian, masih terdapat sebagian aplikasi yang masih didistribusikan melalui platform desktop saja. Salah satunya adalah RAPTOR, sebuah aplikasi yang cukup banyak digunakan untuk mensimulasikan diagram alir. Kami merasa hal ini

cukup disayangkan dikarenakan diagram alir adalah salah satu media yang sangat memudahkan pemula untuk memahami algoritma [5]. Sehingga belum adanya alternatif aplikasi simulasi diagram alir yang lebih mudah diakses dapat menghambat proses belajar sebagian orang yang pada platform yang mereka gunakan tidak tersedia aplikasi sejenis.

Pada penelitian ini kami membuat sebuah aplikasi berbasis web untuk mensimulasikan diagram alir. Aplikasi ini kami namakan Dializer. Dializer dirancang sebagai aplikasi simulasi diagram alir yang lebih mudah diakses dibandingkan solusi yang sudah ada seperti RAPTOR dengan memanfaatkan keunggulan-keunggulan platform web dibandingkan platform desktop.

2. TINJAUAN PUSTAKA

2.1. Aplikasi Berbasis Web

Aplikasi berbasis web adalah aplikasi yang berjalan di platform web. Aplikasi jenis ini dapat diakses hanya dengan mengetikkan URL pada peramban internet. Berbeda dengan aplikasi berbasis desktop, dalam aplikasi berbasis web elemen antarmuka ditampilkan dalam peramban secara langsung dan logika aplikasi juga dijalankan di peramban tersebut [6]. Aplikasi berbasis web modern umumnya diimplementasikan menggunakan kerangka kerja seperti Angular ataupun React dengan sebagian besar logika aplikasi berjalan di peramban internet pengguna serta menggunakan web API untuk berkomunikasi dengan web server [7].

2.2. Diagram Alir

Diagram alir atau *flowchart* adalah representasi visual berupa diagram dari sebuah algoritma. Diagram ini terdiri atas berbagai bentuk grafik dua dimensi (*node*) yang saling terhubung oleh garis penghubung. Setiap bentuk grafik pada diagram alir merepresentasikan operasi tertentu dan biasanya terdapat teks di dalamnya [8]. Dengan menggunakan diagram alir, proses-proses yang terjadi dalam suatu sistem/algoritma dapat divisualisasikan dengan bentuk-bentuk yang lebih mudah dipahami, sehingga siapapun yang hendak memahami bagaimana suatu program/algoritma bekerja tidak perlu melihat dan memahami kode programnya secara langsung.

2.3. REST API

Representational State Transfer Application Programming Interface atau biasa disingkat REST API merupakan salah satu jenis arsitektur antarmuka yang banyak digunakan di industri teknologi. Dalam REST API didefinisikan sekumpulan sumber daya serta sekumpulan operasi untuk mengakses sumber daya tersebut. Sumber daya dalam konteks REST API merupakan informasi apapun yang bisa diakses seperti objek, catatan basis data, dan algoritma [9]. Setiap sumber daya memiliki sekumpulan operasi yang dapat dilakukan terhadapnya yang didefinisikan menggunakan metode HTTP, yaitu GET, POST, PUT, PATCH, DELETE, serta OPTIONS. Selain itu, setiap sumber daya juga diidentifikasi menggunakan URI unik seperti `"/users"` untuk mengakses sumber daya yang berupa sekumpulan pengguna.

2.4. Black Box Testing

Pengujian *black box* merupakan salah satu jenis pengujian perangkat lunak yang banyak digunakan. Dalam metode pengujian ini, pengujian dilakukan dengan melalui antarmuka tanpa melihat alur kode. Pengujian pun dilakukan hanya dengan melihat apakah *output* yang dihasilkan sesuai dengan ekspektasi untuk input tersebut tanpa mengetahui bagaimana input tersebut diproses hingga menjadi sebuah *output* [10].

3. METODE PENELITIAN

3.1. Metode Perancangan

Perancangan sistem dilakukan dengan memanfaatkan diagram UML yaitu *class diagram*, *sequence diagram*, *activity diagram*, serta *use case diagram* untuk mendefinisikan beberapa aspek pada sistem seperti alur interaksi pengguna dengan sistem, fungsionalitas yang disediakan sistem, serta perilaku sistem secara umum.

3.2. Metode Pengujian Dengan Black Box Testing

Pengujian *black box* dilakukan dengan menentukan sekumpulan skenario pengujian yang terdiri atas masukan dan keluaran yang diharapkan dari sistem kemudian menguji apakah sistem dapat memenuhi skenario-skenario tersebut. Pengujian *black box* tidak dilakukan secara terpisah untuk bagian

backend dan *frontend*, melainkan kedua bagian tersebut diuji sebagai satu kesatuan dengan skenario-skenario yang dirancang untuk menggambarkan interaksi pengguna dengan aplikasi melalui antarmukanya.

3.3. Metode Pengujian Dengan Survei Pengguna

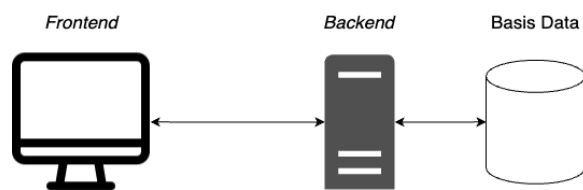
Selain pengujian dengan metode *black box*, Sistem Dializer juga diujikan secara langsung kepada pengguna dengan melakukan survei. Survei dilakukan dengan metode kuesioner yang diberikan kepada 41 mahasiswa Jurusan Informatika Universitas Pembangunan Nasional "Veteran" Jawa Timur yang pernah atau sedang menggunakan RAPTOR. Masing-masing responden diminta untuk menggunakan aplikasi Dializer dan mencoba fitur-fitur yang disediakan oleh Dializer seperti registrasi, login, ubah profil, hingga membuat diagram alir. Setelah mencoba fitur-fitur Dializer, responden diminta untuk mengisi sebuah form agar kami dapat melihat apakah Dializer dapat memenuhi kebutuhan mereka sebagai alternatif perangkat lunak untuk memvisualisasikan diagram alir.

Selain itu, untuk melihat bagaimana kemampuan Dializer dalam bekerja lintas platform, kami juga mengumpulkan data seperti peramban internet, sistem operasi, serta kecepatan internet yang digunakan masing-masing responden.

4. HASIL DAN PEMBAHASAN

4.1. Arsitektur Sistem

Dializer dirancang sebagai aplikasi yang di-*deploy* pada platform web, sehingga dapat dengan mudah diakses melalui peramban internet hanya dengan mengetikkan URL-nya. Ketika mengakses aplikasi ini, pengguna tidak perlu melakukan instalasi apapun selain peramban internet.



Gambar 1. Arsitektur Sistem Dializer

Gambar 1 menunjukkan pembagian Sistem Dializer menjadi dua bagian ditambah dengan satu basis data, yaitu bagian *frontend* dan bagian *backend*. Bagian *frontend* adalah bagian yang bertugas menangani interaksi pengguna. Melalui *frontend*, pengguna dapat melakukan interaksi seperti manajemen *workspace*, menyusun diagram alir, menjalankan visualisasi diagram alir, serta mengubah data pengguna. Sedangkan *backend* digunakan untuk memproses data-data yang dikirimkan oleh *frontend* dan menyimpannya ke ataupun membacanya dari basis data. Pengguna tidak akan berinteraksi secara langsung dengan bagian *backend*.

Format data yang digunakan untuk komunikasi antara *frontend* dan *backend* adalah JSON. Format merupakan salah satu format yang populer digunakan serta memiliki beberapa keunggulan seperti strukturnya yang sederhana, lebih mudah digunakan dalam pembuatan aplikasi, serta banyak didukung oleh bahasa pemrograman yang sudah ada tanpa harus menggunakan pustaka tambahan [11].

4.2. Format Representasi Diagram Alir

Data diagram alir direpresentasikan sebagai larik JSON yang masing-masing butir di dalamnya merupakan sebuah objek yang memiliki beberapa properti seperti ID *node*, posisi x, posisi y, serta konten dari *node* tersebut sebagaimana ditunjukkan oleh Gambar 2.



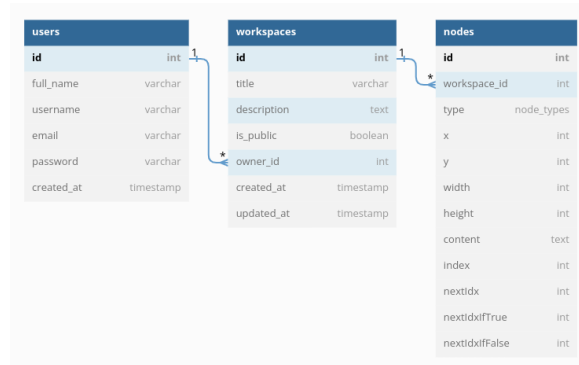
Gambar 2. Representasi diagram alir dalam bentuk JSON

Bagian *frontend* kemudian merekonstruksi representasi visual dari data JSON tersebut dengan menggunakan HTML5 Canvas dan data-data yang tersedia pada tiap butir larik JSON tersebut.

Demikian pula ketika pengguna melakukan perubahan pada diagram alir, pada dasarnya yang diubah adalah representasi JSON dari diagram alir tersebut yang kemudian direkonstruksi kembali bentuk visualnya oleh *frontend* setiap kali terjadi perubahan.

4.3. Rancangan Basis Data

Dalam perancangan struktur penyimpanan data pada basis data Sistem Dializer terdapat tiga entitas, yaitu *User*, *Workspace*, dan *Node*. Hubungan antar entitas ditunjukkan oleh Gambar 3.



Gambar 3. ERD Sistem Dializer

Entitas *User* yang digunakan untuk merepresentasikan data pengguna, memiliki atribut ID pengguna, nama pengguna, email, serta sandi pengguna. Sedangkan entitas *Workspace* yang digunakan untuk merepresentasikan data *workspace* yang dibuat oleh pengguna, memiliki atribut ID *workspace*, judul, deskripsi, pemilik, serta tanggal dibuat dan tanggal terakhir kali *workspace* tersebut di-update. Lalu, entitas *Node* yang digunakan untuk merepresentasikan data *node* memiliki atribut ID *node*, jenis *node*, posisi x dan y, *node* selanjutnya, serta ID *workspace* yang *node* tersebut berada di dalamnya. Ketiga entitas tersebut berhubungan satu sama lain, *User* dapat memiliki banyak *Workspace*, serta entitas *Workspace* dapat memiliki banyak *Node*.

4.4. Rancangan REST API

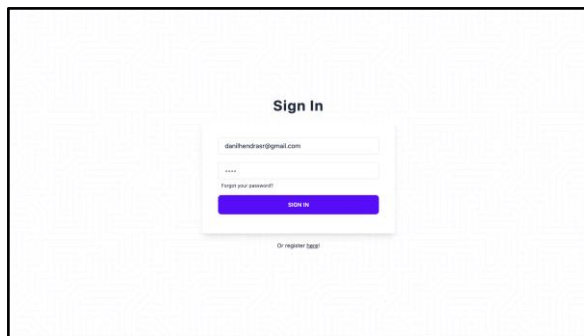
Bagian *frontend* sistem Dializer berkomunikasi dengan *backend* melalui antarmuka berupa REST API. Bagian *backend* menyediakan beberapa *endpoint* yang dapat digunakan oleh *frontend* untuk mengelola objek-objek pada sistem seperti pengguna dan *workspace*. Tabel 1 menunjukkan daftar *endpoint* yang disediakan oleh *backend* dalam antarmuka REST API-nya.

Tabel 1. *Endpoint* REST API pada bagian *backend*

URL	Metode	Fungsi
/workspaces	GET	Mengambil semua <i>workspace</i> yang dimiliki seorang pengguna
/workspaces/<id>	GET	Mengambil detail sebuah <i>workspace</i> berdasarkan ID-nya
/workspaces	POST	Membuat <i>workspace</i> baru
/workspaces/<id>	PUT	Mengubah data <i>workspace</i>
/workspaces/<id>	DELETE	Menghapus <i>workspace</i>
/workspaces/<id>/nodes	GET	Mengambil semua <i>node</i> dalam sebuah <i>workspace</i>
/workspaces/<id>/nodes	PUT	Mengubah susunan <i>node</i> dalam sebuah <i>workspace</i>
/users/<id>	GET	Mengambil data-data detail pengguna
/users/<id>	PUT	Mengubah data pribadi pengguna
/auth/login	POST	Login pengguna
/auth/register	POST	Mendaftarkan pengguna baru

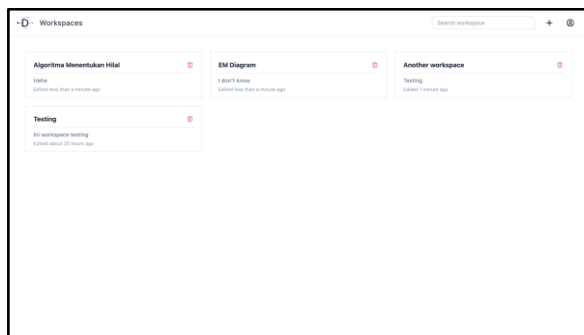
4.5. Antarmuka Aplikasi

Ketika mengakses Dializer, pengguna akan dihadapkan pada layar *login* sebagaimana diperlihatkan pada Gambar 4.



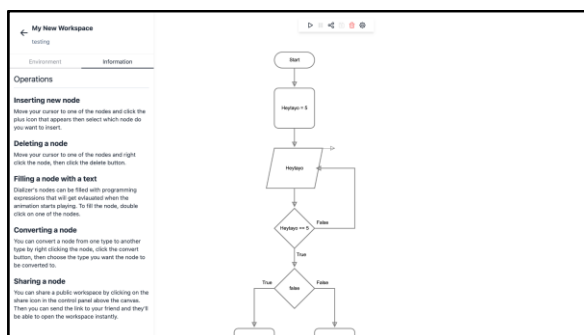
Gambar 4. Tampilan halaman login

Ketika pengguna sudah melakukan login, pengguna akan dibawa ke halaman dasbor di mana pengguna dapat membuat, mengubah, serta menghapus *workspace*. Gambar 5 menunjukkan tampilan halaman dasbor.



Gambar 5. Tampilan halaman dasbor

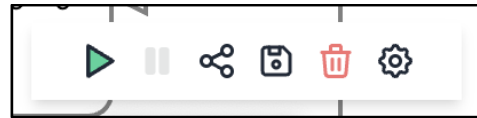
Ketika pengguna mengklik salah satu *workspace* yang sudah dibuat ataupun pengguna membuat *workspace* baru, pengguna akan diarahkan ke halaman *workspace* yang berisi lebih banyak fitur. Pada halaman *workspace* ini pengguna dapat menyusun diagram alir, mengisi *node* diagram alir dengan ekspresi pemrograman, menjalankan visualisasi diagram alir, menghapus *workspace*, membagikan *workspace*, serta mengatur metadata dan visibilitas *workspace*. Gambar 6 menunjukkan tampilan halaman *workspace*.



Gambar 6. Tampilan halaman workspace

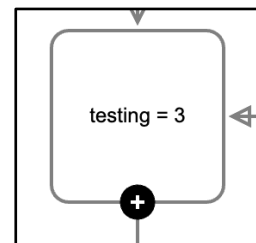
Pada bagian tengah atas halaman *workspace* terdapat panel kontrol untuk pengguna dapat melakukan beberapa operasi terhadap *workspace* dan diagram alir seperti menjalankan visualisasi,

menghapus *workspace*, mengubah metadata *workspace*, serta membagikan *workspace*. Gambar 7 menunjukkan panel kontrol pada halaman *workspace*.

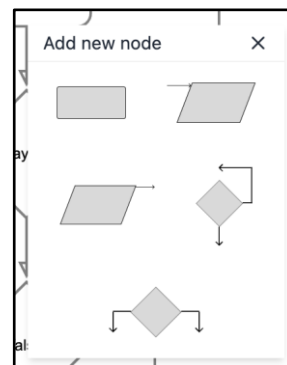


Gambar 7. Panel kontrol pada halaman workspace

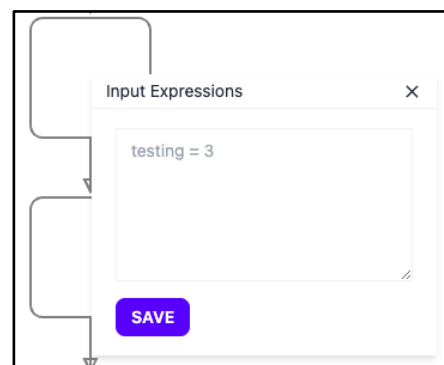
Pengguna juga dapat mengubah susunan diagram alir dengan menambahkan, menghapus, ataupun mengonversi *node* pada diagram alir. Gambar 8 menunjukkan tombol untuk menambahkan *node* baru yang akan muncul di bagian bawah sebuah *node* ketika pengguna mengarahkan kursornya pada *node* tersebut, sedangkan Gambar 9 menunjukkan pilihan *node* yang akan muncul ketika tombol tersebut diklik.



Gambar 8. Tombol untuk menambahkan node baru



Gambar 9. Pilihan node baru untuk dibuat

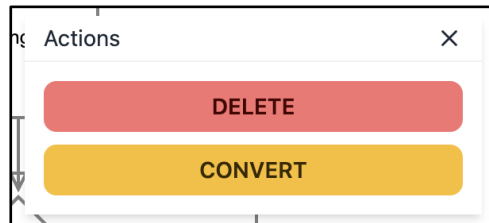


Gambar 10. Pop-up ekspresi pemrograman

Pengguna juga dapat mengisi *node* dengan ekspresi pemrograman dengan cara mengklik dua kali salah satu *node*. Aksi ini akan memunculkan *pop-up* di mana pengguna dapat memasukkan ekspresi

pemrograman yang diinginkan yang ditunjukkan pada Gambar 10.

Selain itu, pengguna dapat mengklik kanan salah satu *node* yang sudah ada untuk menghapus ataupun mengonversi *node* tersebut menjadi jenis *node* lain sebagaimana ditunjukkan oleh Gambar 11.



Gambar 11. Pilihan aksi ketika sebuah *node* diklik kanan

4.6. Hasil Pengujian Black Box

Tabel 2 menunjukkan serangkaian skenario yang diujikan kepada sistem. Berdasarkan tabel tersebut dapat dilihat bahwa perilaku sistem telah sesuai.

Tabel 2. Hasil pengujian black box

No.	Test Case	Hasil
1	Mengisi form dengan email yang tidak valid	Sesuai
2	Mengisi email yang belum terdaftar	Sesuai
3	Mengisi form dengan email dan password yang valid	Sesuai
4	Mengosongkan form	Sesuai
5	Mengisi form dengan email yang tidak valid	Sesuai
6	Mengisi form dengan email yang sudah terdaftar	Sesuai
7	Mengisi form dengan password yang tidak sama	Sesuai
8	Mengisi form dengan nama, email, dan password yang valid	Sesuai
9	Mengosongkan form	Sesuai
10	Menekan tombol + pada dasbor dan langsung diarahkan pada halaman <i>workspace</i>	Sesuai
11	Mengklik salah satu <i>workspace</i> dan diarahkan pada halaman <i>workspace</i>	Sesuai
12	Diagram alir terdiri atas 2 <i>node</i> saja, ketika tombol <i>play</i> diklik maka visualisasi tidak berjalan	Sesuai
13	Diagram alir terdiri atas lebih dari 2 <i>node</i> tetapi terdapat <i>node</i> yang kosong, ketika tombol <i>play</i> diklik maka visualisasi tidak dijalankan	Sesuai
14	Diagram alir terdiri atas lebih dari 2 <i>node</i> dan tidak ada <i>node</i> kosong, ketika tombol <i>play</i> diklik maka visualisasi dijalankan	Sesuai
15	Penambahan <i>node</i> input	Sesuai
16	Penambahan <i>node</i> output	Sesuai
17	Penambahan <i>node</i> process	Sesuai
18	Penambahan <i>node</i> if	Sesuai
19	Modifikasi konten <i>node</i> process	Sesuai
20	Modifikasi konten <i>node</i> if	Sesuai
21	Modifikasi konten <i>node</i> input	Sesuai
22	Modifikasi konten <i>node</i> output	Sesuai

23	Ketika <i>node</i> dihapus, <i>node</i> hilang dari diagram alir	Sesuai
24	Ketika <i>node</i> non-branching dikonversi ke jenis lain, <i>node</i> akan berubah menjadi jenis tersebut	Sesuai
25	Ketika <i>node</i> branching dikonversi menjadi jenis lain, <i>node</i> akan berubah menjadi jenis <i>node</i> tersebut dan jalur <i>node</i> - <i>node</i> yang berada pada jalur percabangan <i>node</i> branching tersebut akan terhapus	Sesuai
26	Edit judul <i>workspace</i>	Sesuai
27	Ketika tombol <i>share</i> diklik URL tersalin ke papan klip pengguna	Sesuai
28	Ketika pengguna lain mengakses <i>workspace</i> yang dibagikan, pengguna tersebut tidak dapat membuat perubahan pada <i>workspace</i>	Sesuai
29	Ketika tombol tempat sampah pada dasbor diklik maka <i>workspace</i> akan terhapus	Sesuai
30	Ketika tombol tempat sampah pada halaman <i>workspace</i> diklik maka sistem <i>redirect</i> ke dasbor dan <i>workspace</i> terhapus	Sesuai
31	Edit nama pengguna	Sesuai
32	Isi email dengan email yang tidak valid	Sesuai
33	Isi email dengan email yang valid	Sesuai
34	Isi dengan password yang tidak sesuai	Sesuai
35	Lakukan lupa password untuk email yang tidak terdaftar	Sesuai
36	Lakukan lupa password untuk email yang terdaftar	Sesuai

4.7. Hasil Survei Pengguna

Setelah diajukan pertanyaan-pertanyaan yang tertera pada Tabel 4 kepada 41 responden yang menjadi target survei, didapati nilai dari masing-masing pertanyaan.

Untuk mendapatkan nilai secara keseluruhan, masing-masing pertanyaan dihitung nilai rata-ratanya. Nilai ini berupa angka antara 1 sampai 5 dengan pengelompokan sebagaimana tercantum pada Tabel 3.

Tabel 3. Pengelompokan nilai

Nilai	Deskripsi
0 - 0.99	Sangat negatif
1 - 1.99	Negatif
2 - 2.99	Netral
3 - 3.99	Positif
4 - 5	Sangat positif

Tabel 4. Hasil survei pengguna

No	Pertanyaan	Rata-rata Nilai (/5)	Keterangan
1.	Seberapa cepat Dializer dapat dimuat pada peramban internet Anda?	4,63	Sangat cepat
2.	Seberapa mudah proses registrasi pengguna baru Dializer?	4,75	Sangat mudah
3.	Seberapa mudah dipahami	4,29	Sangat

	desain antarmuka Dializer?		mudah
4.	Secara keseluruhan, seberapa mudah digunakankah Dializer?	4,46	Sangat mudah
5.	Seberapa mudah Dializer dapat diakses jika dibandingkan dengan aplikasi sejenis seperti RAPTOR?	4,36	Jauh lebih mudah
6.	Seberapa akurat Dializer dalam menggambarkan diagram alir yang Anda buat?	4,34	Sangat akurat
7.	Seberapa memadai fungsionalitas input ekspresi pemrograman pada node diagram alir yang disediakan Dializer?	4,09	Sangat memadai
8.	Seberapa baik dan akurat animasi diagram alir dapat berjalan?	4,43	Sangat baik dan akurat
9.	Seberapa mudah proses membagikan workspace ke rekan Anda?	4,36	Sangat mudah
10.	Apakah fitur di atas memudahkan Anda?	4,56	Sangat memudahkan
11.	Apakah fitur-fitur yang disediakan Dializer mencukupi untuk kebutuhan Anda?	3,97	Mencukupi
12.	Jika situasi tidak mengharuskan Anda menggunakan RAPTOR, seberapa mungkin Anda lebih memilih menggunakan Dializer untuk memvisualisasikan diagram alir?	4,21	Sangat mungkin
Rata-rata Nilai		4,37/5	

Hasil survei juga menunjukkan bahwa 82% responden menyatakan Dializer lebih mudah diakses, 94% responden menyatakan Dializer dapat dimuat dengan cepat, 85% menyatakan Dializer dapat mensimulasikan diagram alir dengan akurat, serta 80% responden merespon positif terkait kemungkinan mereka lebih memilih Dializer dibandingkan RAPTOR untuk memvisualisasikan diagram alir.

Selain itu, didapati pula bahwa 68% pengguna menggunakan Google Chrome untuk mengakses Dializer dengan diiringi 24.4% pengguna menggunakan Microsoft Edge serta sisanya menggunakan Firefox dan Brave Browser, 92.7% pengguna menggunakan Sistem Operasi Windows dengan sisanya menggunakan Sistem Operasi Linux ataupun MacOS, 34.1% memiliki kecepatan internet 1MB/s - 5MB/s dengan diiringi 26.8% memiliki kecepatan internet 5MB/s - 10MB/s serta 17.1% memiliki kecepatan internet 500KB/s - 1MB/s dan lebih dari 10MB/s.

Meskipun pengguna menggunakan peramban, sistem operasi, serta kecepatan internet yang beragam, berdasarkan hasil survei yang tertera pada Tabel 4

dapat dilihat bahwa Dializer masih dapat melakukan fungsinya dengan baik sesuai dengan ekspektasi pengguna dengan mendapatkan nilai rata-rata secara keseluruhan 4.37/5 yang tergolong sangat positif.

5. KESIMPULAN DAN SARAN

Aplikasi Dializer berhasil mendapatkan respon positif dari sebagian besar responden. Tercatat 82% responden menyatakan bahwa Dializer lebih mudah diakses dibandingkan RAPTOR, 18% responden memilih netral, serta tidak ada responden yang menyatakan Dializer lebih sulit diakses. Kemudahan akses Dializer didukung dengan fungsionalitas yang sudah memadai yang dapat dilihat dari nilai rata-rata Dializer secara keseluruhan yang berada di angka 4,37. Selain itu, Dializer mampu bekerja dengan baik lintas-platform. Tercatat meskipun responden menggunakan peramban internet dan sistem operasi yang berbeda-beda, semua responden menyatakan bahwa Dializer dapat diakses pada perangkat mereka dengan 94% responden menyatakan bahwa Dializer dapat dimuat dengan cepat dan 85% responden menyatakan Dializer dapat memvisualisasikan diagram alir dengan akurat. Kendati demikian, Dializer masih memiliki kekurangan, sebagian responden mengeluhkan tidak adanya fitur *auto-save* ketika pengguna dalam proses mengubah susunan diagram alir dan belum adanya fitur kolaborasi secara *real-time*.

DAFTAR PUSTAKA

- [1] L. Corral, A. Sillitti, G. Succi, A. Garibbo, and P. Ramella, "Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm," Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software, 2012. doi:10.1145/2048237.2157457.
- [2] Y. Wang, Y. Li, Y. Tan, Q. Zhang, and J. Zheng, "Large scale network topology visualization system based on three.js," Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications, 2016. doi:10.2991/icaita-16.2016.39.
- [3] R. Buels et al., "JBrowse: A dynamic web platform for Genome Visualization and analysis," Genome Biology, vol. 17, no. 1, 2016. doi:10.1186/s13059-016-0924-1.
- [4] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen, "The death of binary software: End user software moves to the web," 2011 Ninth International Conference on Creating, Connecting and Collaborating through Computing, 2011. doi:10.1109/c5.2011.9.
- [5] N. Tiwari and L. Prasad, "Reverse engineering tools for simplifying programming environment through flowcharting," International Journal of Engineering Trends and Technology, vol. 26, no. 2, pp. 65–71, 2015. doi:10.14445/22315381/ijett-v26p212.

-
- [6] K. C. Bourne, "Chapter 2 - Design," in *Application Administrators Handbook*, 1st ed, Elsevier, Inc, 2014, pp. 12–19
- [7] D. Wang, M. Galster, and M. Morales, "Application Monitoring for bug reproduction in web-based applications," *Journal of Systems and Software*, p. 111834, 2023. doi:10.1016/j.jss.2023.111834.
- [8] J. S. Kuruville et al., "Flowchart plagiarism detection system: An image processing approach," *Procedia Computer Science*, vol. 115, pp. 533–540, 2017. doi:10.1016/j.procs.2017.09.111.
- [9] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "Restful API Architecture Based on Laravel Framework," *Journal of Physics: Conference Series*, vol. 910, p. 012016, 2017. doi:10.1088/1742-6596/910/1/012016.
- [10] Rizky Parlika, Tasya Ardhian Nisaa', Shavira Maya Ningrum, and Berlianda Adha Haque, "Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box ", *teknomatika*, vol. 10, no. 2, pp. 131-140, Oct. 2020.
- [11] A.-R. Breje, R. Gyorödi, C. Gyorödi, D. Zmaranda, and G. Pecherle, "Comparative Study of Data Sending Methods for XML and JSON Models," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 12, 2018. doi:10.14569/ijacsa.2018.091229.