

# Pengembangan *Game* Ular Tangga Aritmatika Berbasis Web

Aziz Solihin<sup>1</sup>, Yuni Yamasari<sup>2</sup>

<sup>1,2</sup> Teknik Informatika, Universitas Negeri Surabaya

<sup>1</sup>[aziz.19049@mhs.unesa.ac.id](mailto:aziz.19049@mhs.unesa.ac.id)

<sup>2</sup>[yuniyamasari@unesa.ac.id](mailto:yuniyamasari@unesa.ac.id)

**Abstrak**— Permainan telah meluas seiring perkembangan teknologi dan digitalisasi, baik dalam bentuk aplikasi mobile maupun berbasis web. Selain bertujuan sebagai hiburan, permainan juga dapat dikembangkan dengan tujuan edukasi, salah satunya permainan sebagai sarana belajar matematika. Permainan ular tangga sebagai salah satu permainan papan klasik dapat dikembangkan menjadi sebuah aplikasi permainan edukasi dengan cara dipadukan materi aritmatika dasar. Untuk itu, penelitian ini memfokuskan pada pengembangan sebuah permainan atau *game* berbentuk aplikasi web yang dipadukan dengan aritmatika dasar. *Game* ini dapat dimainkan melalui perangkat pribadi, baik di desktop maupun seluler. Pengembangan dilakukan dengan metode *Game Development Life Cycle* (GDLC) dengan harapan mampu menghadirkan permainan yang berkualitas. Proses pengembangan ini berdasarkan model GDLC, yaitu *initiation*, *pre-production*, *production*, *testing*, dan *beta*. Hasil penelitian ini memperlihatkan aplikasi *game* Ular Tangga Aritmatika berbasis web dapat dimainkan melalui *browser* pada perangkat pengguna secara baik yang didasarkan pada hasil pengujian internal.

**Kata Kunci**— permainan, ular tangga, aritmatika dasar, aplikasi web, GDLC

## I. PENDAHULUAN

Permainan sudah melekat sebagai suatu hiburan, khususnya bagi anak-anak. Dengan perkembangan teknologi hingga saat ini, keberagaman permainan di dunia digital semakin meluas. Permainan atau biasa disebut *game* yang awalnya dimainkan dengan media papan (*board game*) sudah dapat dimainkan secara digital melalui gadget pribadi. Berbagai jenis *game* tersebar luas di internet dan memudahkan setiap orang yang ingin menghibur diri dengan bermain *game*, baik dalam bentuk *mobile game* maupun *web-based game*. Walau begitu, *game* tidak selalu dijadikan media hiburan saja, tetapi dapat juga menjadi media pembelajaran, khususnya untuk anak sekolah.

*Game* yang mengedukasi adalah *game* yang mampu memberikan motivasi dalam pembelajaran, bukan hanya untuk sekolah tapi juga untuk keseharian. Hakim & Sari (2019) menyebutkan bahwa suatu aplikasi *game* matematika seolah mengajak penggunanya bermain seperti *game* pada umumnya, namun di dalamnya dapat membantu menumbuh kembangkan kemampuan berhitung [1]. Dengan kata lain, aritmatika yang merupakan perhitungan yang sering digunakan sehari-hari dapat dipadukan dengan suatu *game* yang menyenangkan.

Ular tangga merupakan salah satu *board game* klasik yang sudah lama ada. Permainan ini memberikan tantangan berdasarkan keberuntungan pemain dan sangat mudah untuk

dimainkan. Digitalisasi telah membuat permainan ular tangga dapat dimainkan tanpa harus mencari alat-alat fisik (seperti papan ular tangga, bidak dan dadu) yang dibutuhkan untuk bermain, berupa aplikasi *game* ular tangga yang dapat diakses melalui gadget pribadi.

Penelitian terkait *game* ular tangga sebagai media edukasi sudah banyak dilakukan. Salah satunya adalah penelitian dari Firmansyah & Yamasari (2022) yang menghasilkan sebuah *game* ular tangga yang dipadukan dengan soal kepramukaan yang dimainkan pada perangkat Android dengan empat pemain pada setiap perangkat yang digunakan [2]. Sedangkan di bidang matematika, beberapa penelitian menunjukkan bahwa penggunaan media permainan ular tangga dapat berpengaruh dalam meningkatkan motivasi dalam pembelajaran matematika, kemampuan matematis, dan hasil belajar peserta didik [3], [4].

Menurut Pressman, *game* adalah sejenis aplikasi atau *software* penyedia hiburan. Sehingga metode pengembangan yang dapat digunakan adalah *Software Development Life Cycle* (SDLC). Akan tetapi, jika menggunakan SDLC, maka akan ada beberapa tantangan yang dihadapi karena SDLC merupakan proses rekayasa yang sistematis. Sedangkan *game* tidak sepenuhnya rekayasa, melainkan lebih fleksibel ke kreatifitas dan imajinatif dengan kombinasi beberapa aspek lain [5]. Oleh karena itu, dibutuhkan metode yang lebih spesifik seperti *Game Development Life Cycle* (GDLC) yang dikembangkan Ramadan & Widyani (2013) memungkinkan proses pengembangan *game* dengan harapan mampu menghadirkan kualitas *game* yang menyenangkan, fungsional, seimbang, dan kemudahan akses [5].

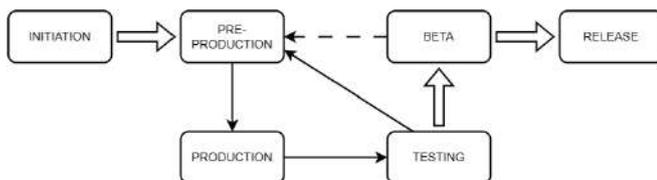
Suatu *game* dapat dikembangkan menjadi aplikasi seluler (*mobile*) bahkan aplikasi web. Aplikasi web modern memiliki kemampuan untuk meningkatkan *mobile experience* sebagai sebuah *Progressive Web App* (PWA). PWA mencakup kemampuan aplikasi yang progresif, responsif, memungkinkan kemampuan tanpa konektivitas internet, *push notification*, mudah ditemukan, dapat diinstal tanpa melalui *app store/play store*, dan terlihat seperti aplikasi *mobile* pada umumnya [6].

Suatu aplikasi web progresif memiliki kemampuan instalasi dengan sebuah *web app manifest*. *Web app manifest* mampu mengatur bagaimana *browser* menampilkan suatu PWA dan perilakunya di perangkat, baik desktop maupun *mobile*. Agar aplikasi web menjadi PWA, aplikasi tersebut harus dapat diinstal, dan agar dapat diinstal, aplikasi tersebut harus menyertakan *manifest* [7], [8].

Berdasarkan latar belakang tersebut, dilakukan penelitian yang bertujuan untuk mengembangkan sebuah *game* Ular Tangga Aritmatika dalam bentuk aplikasi web. *Game* ini memadukan permainan ular tangga dengan kuis aritmatika sederhana, yang dikembangkan dengan metode GDLC dan dimainkan melalui *browser* pada perangkat masing-masing, baik perangkat Android, iOS, maupun desktop.

## II. METODE PENELITIAN

Penelitian Pengembangan *Game* Ular Tangga Berbasis Web menerapkan metode *Game Development Life Cycle* (GDLC). Model GDLC yang digunakan yaitu model dari Ramadan & Widyani (2013) yang terdiri dari fase *initiation*, *pre-production*, *production*, *testing*, *beta*, dan *release* (lihat Gambar 1). Pada penelitian ini, tahapan dibatasi hingga ke fase *beta*.



Gambar 1. Model GDLC

### A. Game Development Life Cycle

1) *Initiation*: merupakan tahap penentuan ide dan konsep kasar dari *game* yang akan dibuat. Pada fase ini juga dilakukan persiapan kebutuhan pengembangan *game* dengan melakukan studi terkait permainan ular tangga dan matematika, teori pengembangan *web-based game*, dan *user experience* dari suatu *game*.

2) *Pre-production*: merupakan tahap siklus produksi (*production cycle*) yang berfokus pada gambaran umum dari *game*, alur permainan, mekanisme permainan, tantangan, desain tampilan dan prototipe dari *game* yang dibuat, serta logika dan aturan permainan.

3) *Production*: merupakan tahap pembuatan aset dan pemrograman dengan menggunakan bahasa pemrograman sehingga menjadi sebuah aplikasi dengan memperhatikan detail umum, seperti keseimbangan *game*, menambah fitur, perbaikan performa, dan perbaikan *bug*.

4) *Testing*: merupakan tahap uji coba internal untuk menguji kemampuan aplikasi *game* setelah diproduksi. Uji coba dilakukan dengan *playtesting* untuk mengetahui ada tidaknya *bug* saat memainkan *game* sebelum dilanjutkan ke tahap *beta*.

5) *Beta*: merupakan tahapan yang melibatkan pihak ketiga (biasa disebut *beta tester*) untuk menguji *game* yang sudah melalui siklus produksi. Tujuan dari pengujian pada tahap *beta* adalah untuk evaluasi *game* dengan menganalisis *user experience* dengan survei kepuasan terhadap *game* yang dimainkan.

### B. Analisis Kebutuhan

Dalam penelitian ini, *game* Ular Tangga Aritmatika dikembangkan menggunakan perangkat keras berupa sebuah laptop dengan spesifikasi sistem operasi Windows 10 Home, prosesor Intel Core i5-8265U, RAM 8GB, dan penyimpanan SSD 256GB. Sedangkan perangkat lunak yang digunakan antara lain: Visual Studio Code sebagai *code editor*; Affinity Designer sebagai aplikasi untuk mendesain tampilan dan pembuatan asset; dan Chrome dan Microsoft Edge sebagai *live server* dan *browser* uji coba. *Game* ini juga dikembangkan dengan *framework* Phaser dan menggunakan Firebase sebagai *backend* dan *database*.

1) *Phaser*: merupakan *game framework* HTML5 yang cepat dan gratis yang menawarkan rendering WebGL dan Canvas di *browser* desktop dan seluler, baik Android maupun iOS. Phaser dikembangkan oleh Photon Storm dengan dukungan API yang menggunakan bahasa utama Javascript atau Typescript [9]. Phaser dapat membantu pengembang *game* dalam membuat aplikasi *game* dua dimensi (2D) dengan target publikasi untuk *browser*.

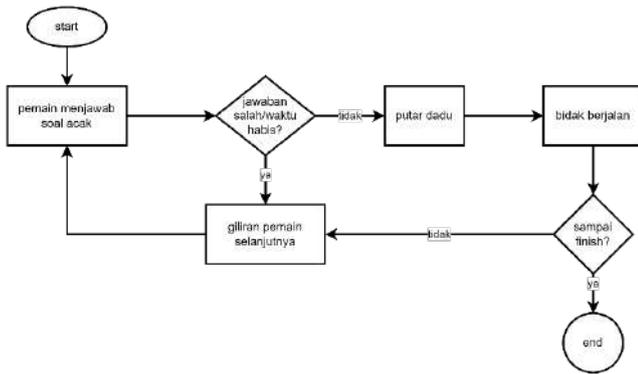
2) *Firebase*: adalah platform pengembangan terpercaya yang disediakan oleh Google yang terintegrasi dengan web, Android, dan iOS untuk mengembangkan aplikasi dan *game*. Firebase menyediakan berbagai produk, seperti Cloud Firestore, Realtime Database, Remote Config, Firebase Extensions, App Check, Cloud Functions, Authentication, Cloud Messaging, Hosting, Cloud Storage, dan Firebase ML [10]. Firebase membantu dalam pengembangan aplikasi yang berkualitas. Firebase merupakan *backend* yang juga digunakan sebagai *database* No-SQL untuk menyimpan data dalam format JSON [11].

### C. Game Design

*Game* Ular Tangga Aritmatika menggunakan konsep *board game* ular tangga biasa yang dimainkan melalui *web browser* pada perangkat pengguna. *Game* ini menyajikan tantangan tidak hanya didasari keberuntungan, tetapi juga berupa kuis aritmatika dasar, seperti penjumlahan, pengurangan, perkalian dan pembagian. *Game* ini ditargetkan untuk kalangan umum, terutama para pelajar.

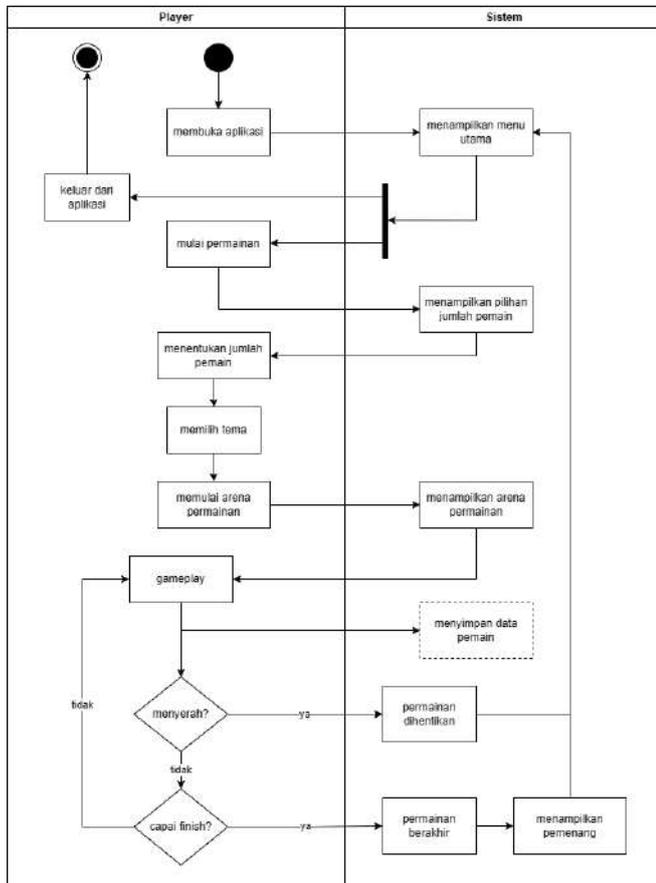
Soal aritmatika dalam *game* Ular Tangga Aritmatika disajikan berupa operasi penjumlahan, pengurangan, perkalian dan pembagian dari dua angka yang diacak oleh sistem.

Alur permainan dari *game* ini sama seperti permainan ular tangga pada biasanya, dengan menambahkan tantangan dimana pemain harus menjawab soal yang ditampilkan dengan benar dalam waktu tertentu agar dapat memutar dadu. Jika jawaban pemain salah atau kehabisan waktu, maka giliran pemain tersebut berakhir dan sistem akan membuat soal baru untuk giliran pemain selanjutnya. Pemain yang pertama kali mencapai *finish* adalah pemenangnya (lihat Gambar 2).



Gambar 2. Alur permainan Ular Tangga Aritmatika

Game Ular Tangga Aritmatika ini memiliki fitur pilihan dua hingga empat pemain, pilihan tema papan, dan soal operasi dua bilangan acak. Selama permainan berlangsung, sistem akan melakukan *log*/mencatat dan menyimpan data setiap permainan dan soal yang dijawab, baik yang benar maupun yang salah. Setelah salah satu pemain berhasil mencapai *finish*, maka permainan berakhir dan sistem akan menampilkan pemenang (lihat Gambar 3).



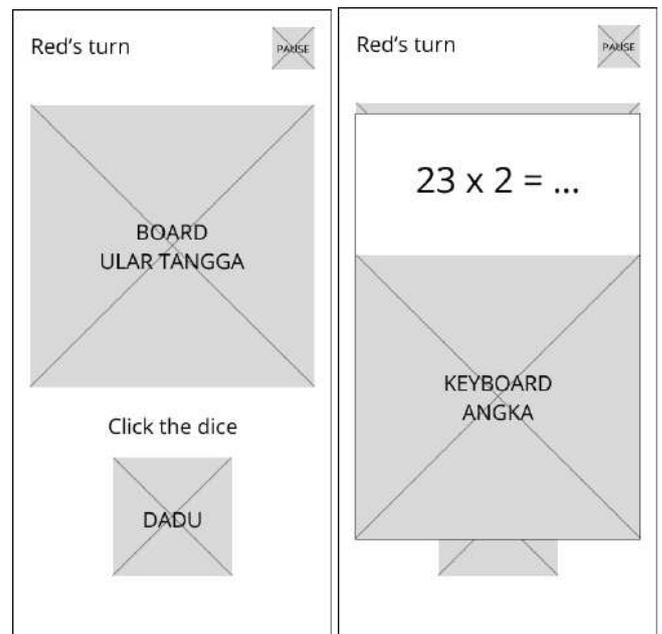
Gambar 3. Alur sistem permainan Ular Tangga Aritmatika

Pada tampilan antarmuka atau *user interface*, aplikasi ini didesain dengan acuan ukuran layar perangkat seluler (*mobile first design*). Hal ini akan membuat tampilan *game* menjadi

lebih nyaman dan mudah digunakan pada perangkat *mobile*. Kerangka awal dari *game* dapat dilihat pada Gambar 4 dan Gambar 5.



Gambar 4. Desain kerangka menu awal (kiri); dan menu pilih tema dan jumlah pemain (kanan)



Gambar 5. Desain kerangka *in-game* (kiri); dan menu soal (kanan)

### III. HASIL & PEMBAHASAN

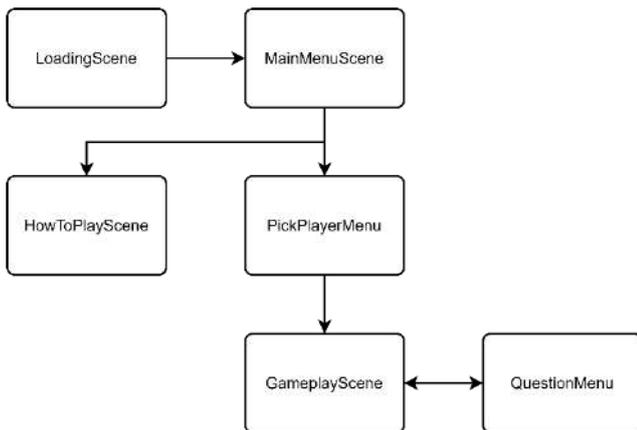
Penelitian ini menghasilkan sebuah aplikasi *game* Ular Tangga Aritmatika bernama "SnadderMath" yang merupakan permainan ular tangga digital sederhana yang dipadukan dengan kuis matematika sederhana. Setiap permainan dimainkan dalam satu perangkat, dimana dua sampai empat

pemain dapat bermain ular tangga sambil berhitung secara bergiliran. Pemain yang mencapai garis finish terlebih dahulu akan menjadi pemenang. Kuis matematika dalam *game* ini berupa operasi dua bilangan acak, seperti penjumlahan, pengurangan, perkalian, dan pembagian.

Aplikasi dipublikasi dengan URL [snaddermath.online](http://snaddermath.online) dan dapat dimainkan oleh pengguna internet.

#### A. Implementasi Game

*Game* SnadderMath dikembangkan dengan *framework* atau *game engine* Phaser. Terdapat beberapa *scene* dalam *game* ini yang diantaranya yaitu *LoadingScene*, *MainMenuScene*, *HowToPlayScene*, *GameplayScene*, *PickPlayerMenu*, dan *QuestionMenu*.

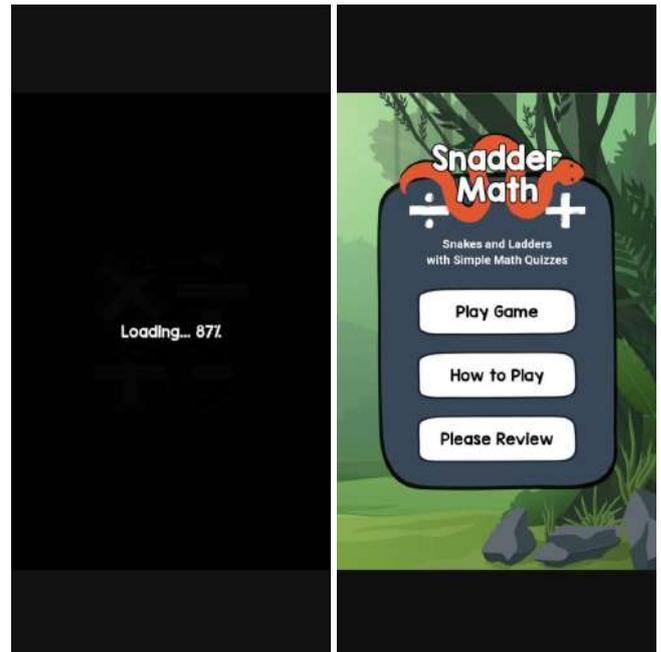


Gambar 6. *Game Scenes*

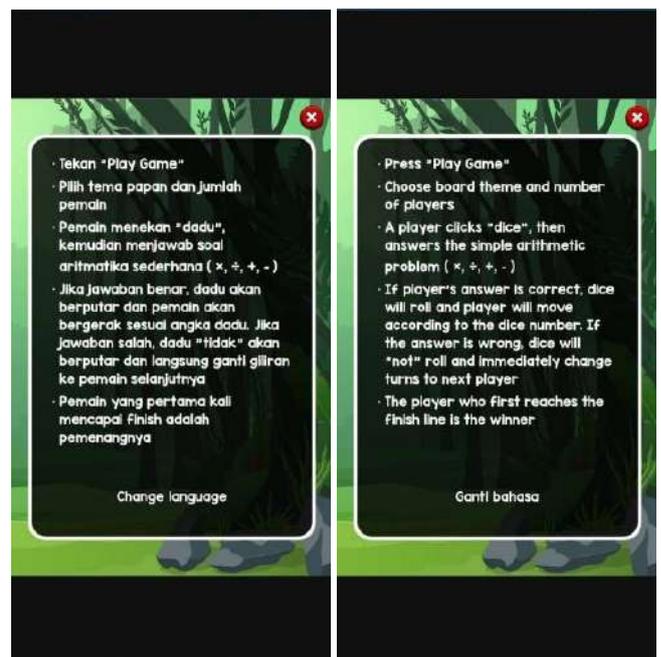
*LoadingScene* dijalankan pertama kali untuk memuat aset-aset yang dibutuhkan dan telah dibuat dengan menampilkan *loading screen* sebelum menjalankan *scene* yang lain. Setelah semua aset berhasil dimuat, *MainMenuScene* dijalankan untuk menampilkan menu awal dari *game*. Tampilan *loading screen* dan menu awal dapat dilihat pada Gambar 7.

Pada menu awal terdapat tombol “*Play Game*”, “*How to Play*” dan “*Please Review*”. Menekan tombol “*Play Game*” akan menjalankan *PickPlayerMenu*. Menekan tombol “*How to Play*” akan menjalankan *HowToPlayScene*.

Menjalankan *HowToPlayScene* akan menampilkan halaman *how to play* yang berisi cara memainkan *game* SnadderMath dalam Bahasa Indonesia dan Bahasa Inggris seperti pada Gambar 8. Bahasa dapat diubah dengan menekan “*Change language*” atau “*Ganti bahasa*”.



Gambar 7. Tampilan *loading screen* (kiri) dan menu awal (kanan)



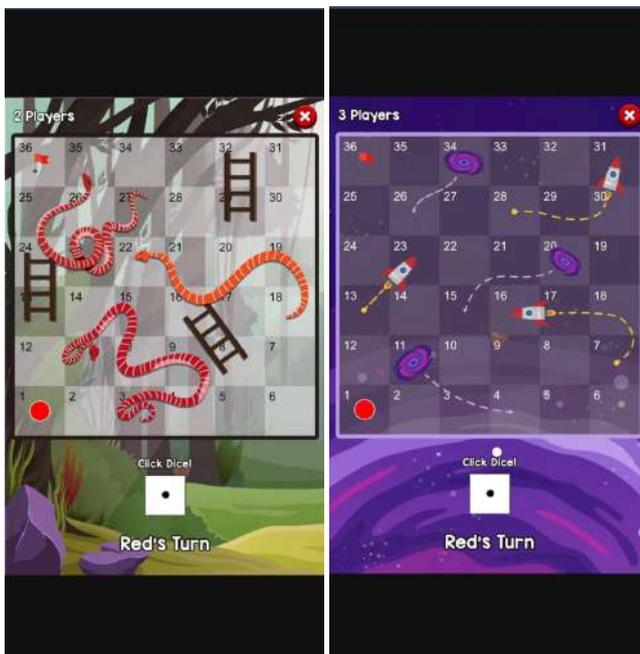
Gambar 8. Tampilan menu *how to play* dalam Bahasa Indonesia (kiri) dan Bahasa Inggris (kanan)

Menjalankan *PickPlayerMenu* akan menampilkan menu untuk memilih tema papan (*forest* atau *space*) dan jumlah pemain (2 sampai 4 pemain dengan pion berwarna *Red*, *Green*, *Blue*, dan *Yellow*) seperti pada Gambar 9. Dengan menekan salah satu pilihan jumlah pemain, *GameplayScene* akan dijalankan dan menampilkan arena permainan yang terdiri dari papan ular tangga dengan tema pilihan, dadu, dan pion pemain dengan warna masing-masing yang berbeda (lihat Gambar 10).



Gambar 9. Tampilan menu pilih tema dan jumlah pemain

Papan ular tangga yang digunakan dalam *game* SnadderMath ini memiliki ukuran 6×6 (36 petak) dengan aset ular dan tangga pada tema *forest* dan aset lubang hitam dan roket pada tema *space*.



Gambar 10. Tampilan arena permainan tema *forest* (kiri) dan tema *space* (kanan)

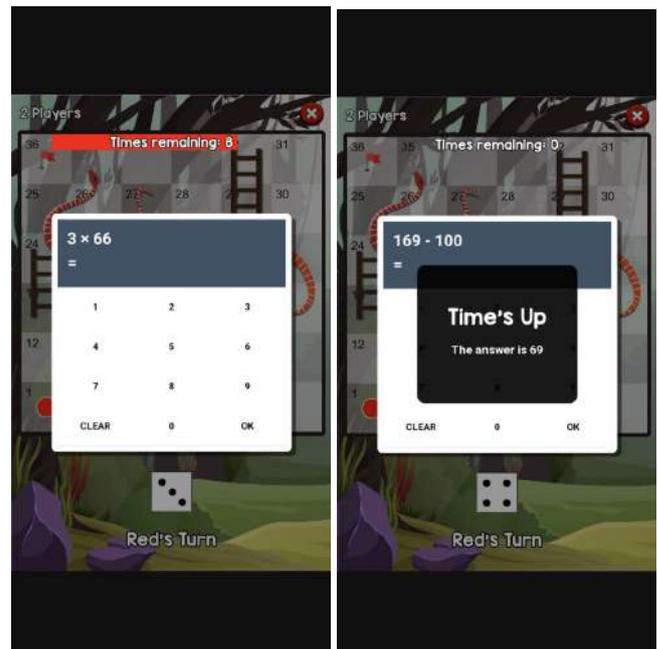
Alur permainan yang diimplementasikan menunjukkan kesesuaian dengan desain alur pada Gambar 2. Dengan menekan dadu, *QuestionMenu* akan dijalankan dan

menampilkan menu soal matematika dengan batas waktu 10 detik.

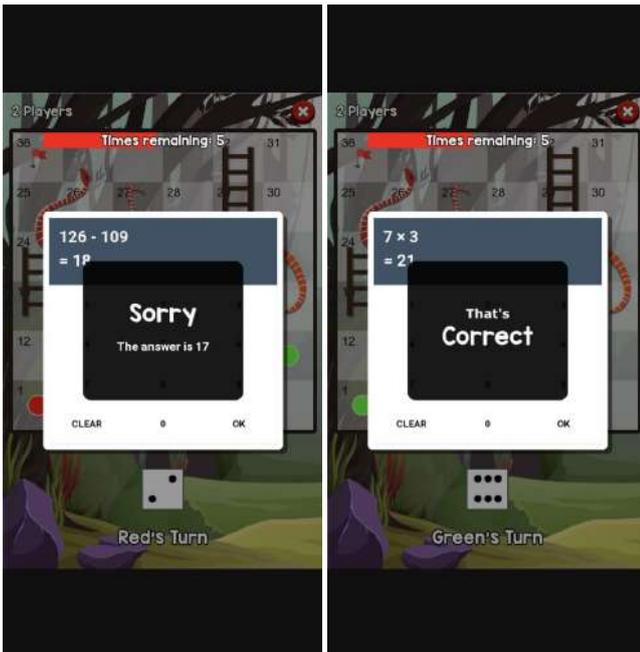
Dalam pembuatan soal, operator yang digunakan yaitu antara operator pembagian, perkalian, penjumlahan, atau pengurangan. Kemudian dibangkitkan dua bilangan bulat positif dan mengoperasikan kedua bilangan tersebut dengan operator yang sudah ditentukan. Operasi matematika ini dibatasi sebagai berikut.

- Jika pembagian, kedua bilangan tidak boleh sama dan harus dapat dibagi habis, dimana pembilang di antara 2 sampai 100 dan penyebut di antara 2 sampai 20.
- Jika perkalian, bilangan pertama di antara 2 sampai 20 dan bilangan kedua di antara 2 sampai 100 dengan hasil perkalian tidak lebih dari 200.
- Jika penjumlahan, kedua bilangan di antara 2 sampai 100.
- Jika pengurangan, bilangan pertama harus lebih besar dari bilangan kedua (pengurang) dimana bilangan pertama di antara 2 sampai 200 dan bilangan kedua di antara 2 sampai 120.

Saat permainan berlangsung, apabila pemain tidak sempat menjawab, maka akan menampilkan *pop up* waktu habis beserta jawaban yang seharusnya (lihat Gambar 11 kanan). Jika jawaban pemain salah, maka akan menampilkan *pop up* jawaban salah beserta jawaban yang seharusnya, dan jika jawaban pemain benar, maka akan menampilkan *pop up* jawaban benar (lihat Gambar 12).



Gambar 11. Tampilan menu soal (kiri) dan saat waktu habis (kanan)



Gambar 12. Tampilan saat menjawab salah (kiri) dan menjawab benar (kanan)

Permainan berakhir jika salah satu pion pemain mencapai *finish* dan menjadi pemenang. Kondisi menang akan menampilkan *pop up* pemenang (lihat Gambar 13) dan arena dapat ditutup.



Gambar 13. Tampilan saat mencapai *finish* dan permainan berakhir

### B. Pencatatan Data Permainan ke Database

Data setiap permainan Ular Tangga Aritmatika tercatat dalam bentuk objek data dan disimpan di Realtime Database

pada Firebase. Pencatatan data permainan dilakukan saat permainan berlangsung, seperti alur sistem pada Gambar 3.

Data tersebut berbentuk objek Javascript yang memiliki properti *id*, *start*, *end*, *winner*, dan *dataList*. Properti *id* berisi nomor unik (*id*) untuk setiap permainan. Properti *start* berisi tanggal dan waktu permainan dimulai. Properti *end* berisi tanggal dan waktu permainan berakhir. Properti *winner* berisi nama pemenang. Properti *dataList* berisi data setiap pemain.

```
gameObject = {
  id: 1694670199232,
  start: "14 Sep 2023, 13:43:20",
  end: "14 Sep 2023, 14:01:43",
  winner: "Red",
  dataList: playerList
};
```

Gambar 14. Contoh objek data permainan

Properti *dataList* berisi objek pemain dengan nama *Red*, *Green*, *Blue*, dan *Yellow*, tergantung jumlah pemain dalam satu permainan. Setiap objek pemain memiliki properti *position*, *correctAnswerTotal*, *wrongAnswerTotal*, *rollTotal*, *isWon*, dan *quiz*. Properti *position* berisi posisi pemain berdasarkan nomor kotak dari papan ular tangga. Properti *correctAnswerTotal* berisi jumlah soal yang dijawab benar. Properti *wrongAnswerTotal* berisi jumlah soal yang dijawab salah. Properti *isWon* berisi nilai *boolean* apakah pemain mencapai *finish* dan menang (*true*) atau tidak (*false*). Dan properti *quiz* merupakan daftar soal (dalam bentuk *array*) yang didapatkan oleh pemain tersebut.

```
playerList = {
  Red: {
    position: 36,
    correctAnswerTotal: 11,
    wrongAnswerTotal: 5,
    rollTotal: 16,
    isWon: true,
    quiz: quizList
  },
  Green: {
    position: 27,
    correctAnswerTotal: 10,
    wrongAnswerTotal: 2,
    rollTotal: 12,
    isWon: false,
    quiz: []
  },
  // Blue: {}
  // Yellow: {}
};
```

Gambar 15. Contoh objek data pemain

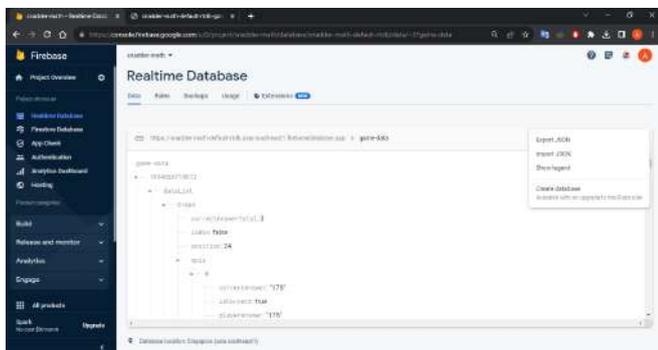
Soal yang didapatkan oleh setiap pemain dicatat dalam bentuk *array* dari objek-objek soal yang memiliki properti *question*, *correctAnswer*, *playerAnswer*, dan *isCorrect*. Properti *question* berisi soal operasi dua bilangan yang diacak. Properti *correctAnswer* berisi jawaban dari *question*. Properti *playerAnswer* merupakan jawaban yang diinput oleh pemain. Dan *isCorrect* berisi nilai *boolean* apakah *playerAnswer* sama dengan *correctAnswer* atau tidak. Jika sama, maka *isCorrect*

bernilai *true* dan jawaban pemain dianggap benar. Sedangkan jika tidak sama, maka *isCorrect* bernilai *false* dan jawaban pemain dianggap salah.

```
quizList = [  
  {  
    question: "64 + 16",  
    correctAnswer: "4",  
    playerAnswer: "3",  
    isCorrect: false  
  },  
  {  
    question: "3 x 64",  
    correctAnswer: "192",  
    playerAnswer: "192",  
    isCorrect: true  
  },  
  // ...  
]
```

Gambar 16. Contoh list soal tiap pemain

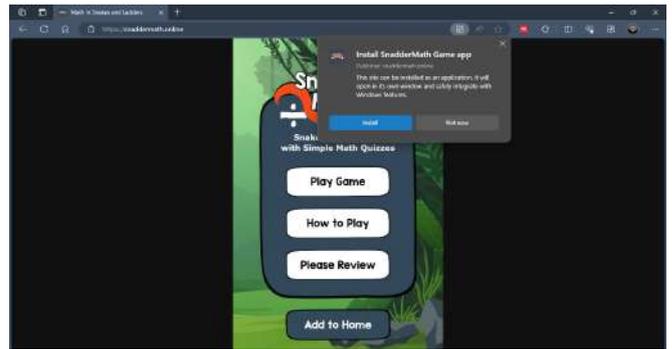
Data-data permainan dicatat secara *realtime* ke *database* dalam proyek Firebase “snadder-math” dalam bentuk JSON.



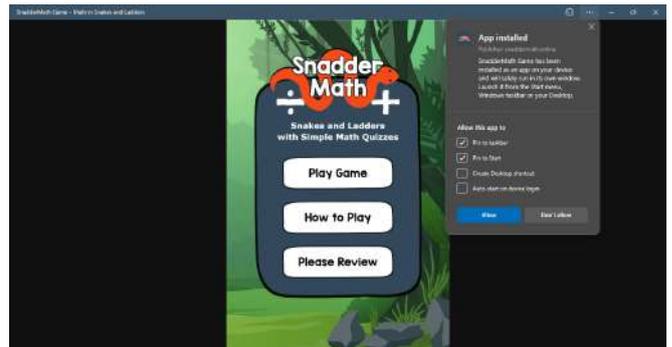
Gambar 17. Tampilan Realtime Database proyek "snadder-math"

### C. Pemanfaatan Web App Manifest

Game SnadderMath dikembangkan sebagai *Progressive Web App* (PWA) yang memiliki kemampuan dan kemudahan pemasangan (*installable*) dengan *manifest*. *Manifest* mampu mengatur bagaimana perilaku *browser* terhadap aplikasi game SnadderMath, seperti menampilkan tombol instalasi di sebelah kanan alamat web pada *browser* desktop dan membuat *custom button* pada menu awal (tombol "Add to Home") untuk mempermudah pengguna perangkat seluler jika ingin menginstal *game*.

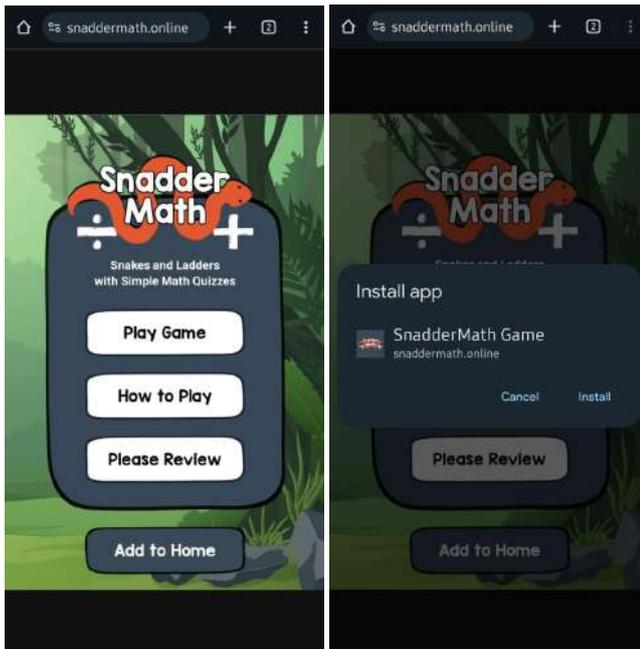


Gambar 18. Tampilan game SnadderMath pada browser Microsoft Edge



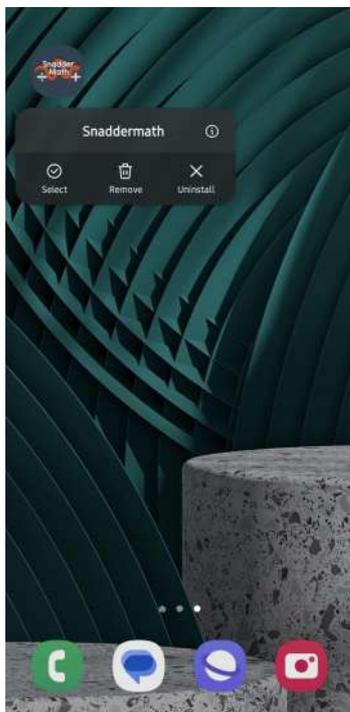
Gambar 19. Tampilan aplikasi game SnadderMath setelah diinstal melalui browser Microsoft Edge

Dengan menginstal aplikasi *game*, pengguna dapat mengakses/membuka *game* melalui ikon aplikasi pada perangkat pengguna (lihat Gambar 21) tanpa harus membuka *link* dari aplikasi lagi. Tampilan dari aplikasi *game* SnadderMath setelah diinstal di perangkat *mobile* sama seperti pada Gambar 7.



Gambar 20. Tampilan *game* SnadderMath pada *browser* Chrome Android (kiri) dan konfirmasi jika tombol “Add to Home” ditekan (kanan)

Karena keterbatasan kemampuan dari *manifest*, fitur instal *game* SnadderMath saat ini hanya berlaku untuk *browser* Chrome, Edge, Safari (pada iOS), dan Samsung Internet [12]. Hal ini memungkinkan untuk tombol “Add to Home” tidak muncul.



Gambar 21. Tampilan ikon aplikasi pada *homescreen* perangkat Android

#### D. Pengujian Internal

Pengujian setelah tahap produksi dilakukan untuk mengetahui apakah *game* sudah dapat berjalan dan fitur-fitur berfungsi dengan baik sesuai dengan desain *game* yang sudah dibuat, baik alur permainan, alur sistem, dan tampilan. Pengujian ini juga dilakukan untuk menghindari *bug* sebelum diuji cobakan ke pihak ketiga untuk dievaluasi. Hasil pengujian pada tahap ini yaitu:

- Aplikasi *game* SnadderMath dan fitur-fiturnya dapat berjalan dengan baik.
- Data permainan dapat tercatat di *database*.
- Aplikasi *game* dapat diinstal di desktop, Android dan iOS (hanya jika diakses melalui *browser* yang kompatibel).

#### IV. KESIMPULAN

Pengembangan *game* Ular Tangga berbasis web ini telah dapat digunakan untuk media belajar matematika. Hal ini didasarkan pada hasil baik pada pengujian internal. *Game* SnadderMath dapat diinstal ke *homescreen* perangkat melalui *browser* Chrome, Edge, Safari (pada iOS), dan Samsung Internet.

Penelitian selanjutnya dari *game* Ular Tangga Aritmatika ini adalah pengembangan atau penambahan fitur level kesulitan dan fitur *multiplayer* agar para pemain dapat bermain melalui perangkat masing-masing tanpa harus menggunakan satu perangkat untuk satu permainan. Pengembangan halaman admin dan *backend* juga disarankan agar data permainan dapat dikelola lebih optimal.

#### V. UCAPAN TERIMA KASIH

Puji syukur ke hadirat Allah Swt. yang telah memberikan rahmat dan hidayah-Nya sehingga penelitian dapat berjalan dengan baik. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu dan mendukung dalam penelitian dan penyusunan laporan dan artikel ini. Semoga penelitian ini dapat bermanfaat, terutama di bidang informatika.

#### VI. REFERENSI

- [1] D. L. Hakim dan R. M. M. Sari, “Aplikasi *game* matematika dalam meningkatkan kemampuan menghitung matematis,” *JPPM (Jurnal Penelitian dan Pengembangan Matematika)*, vol. 12, no. 1, hlm. 129–141, 2019.
- [2] I. J. E. Firmansyah dan Y. Yamasari, “Rancang Bangun *Game* Edukasi Ular Tangga Kepramukaan Berbasis Android,” *Journal of Informatics and Computer Science*, vol. 04, no. 02, hlm. 207–218, 2022.
- [3] L. Novita dan F. S. Sundari, “Peningkatan Hasil Belajar Siswa Menggunakan Media *Game* Ular Tangga Digital,” *Jurnal Basicedu*, vol. 4, no. 3, hlm. 716–724, Jun 2020, doi: 10.31004/basicedu.v4i3.428.
- [4] I. Suciati, “Media Permainan ‘Ular Tangga’ pada Pembelajaran Matematika,” *Kognitif: Jurnal Riset*

- HOTS Pendidikan Matematika*, vol. 1, no. 1, hlm. 10–21, Jun 2021, doi: 10.51574/kognitif.v1i1.5.
- [5] R. Ramadan dan Y. Widyani, “Game development life cycle guidelines,” dalam *International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, IEEE Computer Society, 2013, hlm. 95–100. doi: 10.1109/ICACSIS.2013.6761558.
- [6] A. Biørn-Hansen, T. A. Majchrzak, dan T. M. Grønli, “Progressive Web Apps: The Possible Web-native Unifier for Mobile Development,” dalam *WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies*, SciTePress, 2017, hlm. 344–351. doi: 10.5220/0006353703440351.
- [7] “Making PWAs installable,” MDN Web Docs. [Daring]. Tersedia pada: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps/Guides/Making\\_PWAs\\_installable](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Making_PWAs_installable)
- [8] P. LePage, F. Beaufort, dan T. Steiner, “Add a web app manifest,” web.dev. Diakses: 21 September 2023. [Daring]. Tersedia pada: <https://web.dev/add-manifest/>
- [9] “Phaser API Documentation,” Photon Storm. Diakses: 27 September 2023. [Daring]. Tersedia pada: <https://newdocs.phaser.io/docs/3.60.0>
- [10] “Firebase - Make your app the best it can be,” Google for Developers. Diakses: 27 September 2023. [Daring]. Tersedia pada: <https://firebase.google.com/>
- [11] C. Khawas dan P. Shah, “Application of Firebase in Android App Development-A Study,” *Int J Comput Appl*, vol. 179, no. 46, hlm. 49–53, Jun 2018, doi: 10.5120/ijca2018917200.
- [12] “Web app manifests,” MDN Web Docs. Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://developer.mozilla.org/en-US/docs/Web/Manifest>