



Implementasi *Pseudo Random Number Generator* (PRNG) dengan Algoritma *Linear Congruential Generator* (LCG) pada Permainan Mengetik Bertema Candi Prambanan

Galih Rizky Fahrezi^{#1}, Dolly Virgiana Shaka Yudha Sakti^{#2}, Hafiz Muhandi^{*3}

[#]Teknik Informatika, Universitas Budi Luhur

Jl. Ciledug Raya, Petukangan Utara, Kota Jakarta Selatan 12260

¹galihrfahrezi@gmail.com

²dolly.virgianshaka@budiluhur.ac.id

^{*}Informatika, Fakultas Teknik Universitas Tanjungpura

Jl. Prof. Dr. H.Hadari Nawawi Pontianak 78124

³hafizm@informatika.untan.ac.id

Abstrak— Komputer biasa digunakan untuk kebutuhan sehari-hari. apapun kebutuhan itu pastinya tidak luput dari kebutuhan akan mengetik maupun menekan tombol pada *keyboard* namun tidak semua orang mampu mengetik dengan cepat. Disaat bersamaan komputer juga digunakan sebagai sarana hiburan seperti bermain *game* akan tetapi ini menyebabkan hiburan lama seperti kisah rakyat menjadi ditinggalkan. Dikarenakan hal itu dibuatlah sebuah *game* mengetik bergenre *typing game* dengan latar belakang cerita rakyat berupa Candi Prambanan. *Game* akan dikembangkan menggunakan metode *Game Development Life Cycle* (GDLC) dan juga menggunakan algoritma *Linear Congruential Generator* (LCG) yang merupakan bagian dari *Pseudo Random Number Generator* (PRNG) pada permainannya. Hasil yang didapatkan dari penelitian ini adalah para pemain dapat dengan mudah mengerti kisah Candi Prambanan yang disampaikan saat memainkan *game* mengetik dengan genre *typing game* ini.

Kata kunci— *Game, Linear Congruential Generator, Game Development Life Cycle, Cerita Rakyat, Candi Prambanan*

I. PENDAHULUAN

Komputer terus menerus berkembang sesuai dengan berkembangnya zaman dan mencakup berbagai macam bidang, pada saat yang sama orang-orang dituntut untuk dapat mengetik untuk bisa menggunakan komputer seiring dengan berkembangnya zaman. Kemampuan mengetik merupakan kemampuan yang penting untuk dimiliki di masa modern[1]. Walaupun dituntut untuk bisa mengetik namun tidak semua orang dapat mengetik dengan cepat atau leluasa.

Salah satu bidang yang sudah dicakup oleh komputer adalah pendidikan. Pendidikan merupakan salah satu unsur

yang sangat penting dalam pembangunan suatu bangsa[2]. Sering sekali pendidikan pada masa ini mencakup komputer sebagai kurikulumnya.

Tidak hanya untuk kepentingan sehari-hari, komputer juga menjadi sarana hiburan modern seperti *game*. Terutama *game* digital yang sudah menjadi bagian dari aktifitas kehidupan manusia[3]. *Game* adalah permainan elektronik dari kreatifitas pengembang kemudian diimplementasikan ke sebuah media yang memanfaatkan komputer dan grafis supaya bisa ditampilkan diperangkat keras[4]. Pada awalnya *game* hanyalah sebuah media untuk hiburan namun untuk sekarang *game* telah digunakan untuk kebutuhan lain seperti simulasi pendidikan, bisnis, militer, kedokteran, dan lain-lain[5].

Namun, dengan adanya *game* maka hiburan lain seperti cerita rakyat mulai dilupakan dikarenakan kurang menarik lagi. Salah satu cerita rakyat yang terkenal telah menghibur banyak kalangan adalah kisah Candi Prambanan. Meskipun tidak sepopuler dulu lagi namun kisah tersebut masih bisa dikemas ke dalam *game* untuk membuatnya menjadi hiburan yang semakin menarik.

Game tersebut merupakan salah satu bentuk aplikasi edukatif yang artinya dapat digunakan sebagai media pembelajaran yang dalam prosesnya dapat diselesaikan melalui konsep pembelajaran dan *game*[6]. *Game* berjenis edukasi ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan siswa bisa lebih mudah memahami materi pelajaran yang disajikan[7]. *Game* mempunyai jenisnya sendiri atau bisa disebut dengan genre, fungsi dari genre *game* adalah digunakan untuk menggolongkan sebuah *game* berdasarkan bagaimana interaksi dari pemain tersebut[8]. Salah satu genre *game*

yang digunakan sebagai media pembelajaran adalah *game* dengan genre *typing game* (permainan mengetik) yang membuat pemain terpacu untuk mengetik kata dengan secepat dan seakurat mungkin. Genre *typing game* secara tidak langsung akan melatih kecepatan serta keakuratan mengetik pemain saat mengetik dalam *game* tersebut.

Dalam sebuah permainan harus ada kompetisi agar pemain teransang untuk terus bermain kompetisi tersebut untuk menang atau kalah[9]. Namun *game* dapat bersifat monoton dan mudah ditebak jika menampilkan hal yang sama berulang-ulang, maka dari itu diperlukanlah algoritma untuk mengacak soal yang diberikan. Bilangan acak yang dibangkitkan oleh komputer adalah bilangan semu, karena pembangkitnya menggunakan operasi aritmatika[10]. Untuk mengatasi hal tersebut digunakanlah algoritma pengacakan dari *Pseudo Random Number Generator* (PRNG). Ciri khas dari Random Number Generator adalah terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan, hal ini adalah salah satu sifat dari metode ini, dan pseudo random generator pada umumnya[11].

Game Development Life Cycle (GDLC) adalah sebuah metode pengembangan populer yang digunakan dalam pengembangan *game*. *Game Development Life Cycle* (GDLC) merupakan sebuah metode yang menangani pengembangan *game* dari awal hingga akhir.

Dari kajian tersebut maka dibuatlah tujuan untuk membuat *game* dengan genre *typing game* bertemakan kisah Candi Prambanan. *Game* akan dikembangkan menggunakan metode *Game Development Life Cycle* (GDLC) dan juga algoritma *Linear Congruential Generator* (LCG).

II. METODE PENELITIAN

A. *Linear Congruential Generator* (LCG) dan Penerapan Algoritma

Linear Congruential Generator (LCG) adalah salah satu algoritma pengulangan yang termasuk dalam *Pseudo-Random Number Generator* (PRNG). Algoritma LCG (*Linier Congruential Generator*) adalah sebuah metode yang berfungsi untuk membangkitkan bilangan acak yang banyak dipergunakan dalam program komputer[12]. Ciri khas dari *Linear Congruential Generator* adalah terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan Algoritma *Linear Congruential Generator* (LCG) adalah algoritma sederhana dan cepat dalam hal pelaksanaannya[13].

Pada kasus penelitian ini Algoritma Linear Congruential Generator digunakan karena penggunaannya yang mudah dan cepat. Digunakan nya algoritma *Linear Congruential Generator* (LCG) adalah pada saat akan mengambil kata acak pada array yang tersedia. Algoritma *Linear Congruential Generator* (LCG) akan memilih nomor acak semu dan nomor acak tersebut akan digunakan untuk mengambil kata pada array yang tersedia dengan menyesuaikan nomor acak semu dengan nomor pada array kata yang tersedia.

Penggunaan *Linear Congruential Generator* (LCG) cukup sederhana namun mampu mendapat bilangan acak pada penggunaannya. Rumus dari *Linear Congruential Generator* (LCG) adalah:

$$X_{n+1} = (aX_n + c) \text{ mod } m$$

Algoritma ini memiliki beberapa parameter utama yaitu:

- Dengan X sebagai kumpulan bilangan acak semu.
- A sebagai pengali
- C sebagai penambah
- M sebagai modulus
- Xn sebagai bilangan acak pada deretnya
- Xn-1 sebagai bilangan acak pada deret sebelumnya

Sebagai contoh prosesnya adalah sebagai berikut:

$$A = 11$$

$$C = 7$$

$$X_0 = 1$$

$$M = 250$$

$$X_0 = (11(1)+7) \text{ mod } 250 = 18$$

$$X_1 = (11(18)+7) \text{ mod } 250 = 205$$

$$X_2 = (11(205)+7) \text{ mod } 250 = 12$$

$$X_3 = (11(12)+7) \text{ mod } 250 = 139$$

$$X_4 = (11(36)+7) \text{ mod } 250 = 153$$

Pada kasus ini modulus dipilih 250 karena menyesuaikan dengan Panjang array kata yang akan diambil berdasarkan hasil dari penjumlahan tersebut. Hasil dari penjumlahan *Linear Congruential Generator* (LCG) tersebut lalu akan digunakan untuk mengambil kata-kata pada array yang sudah disiapkan sesuai dengan angka hasil penjumlahan. Angka yang didapat kemudian digunakan sebagai untuk mendapat angka acak berikutnya dan akan berulang terus menerus.

Jika diteruskan menjadi tabel maka hasilnya akan seperti tabel berikut:

TABEL I
PERCOBAAN LINEAR CONGRUENTIAL GENERATOR (LOG)

n	(11(n)+7)	X _{n+1}
1	(11(1)+7) mod 250	18
2	(11(18)+7) mod 250	205
3	(11(205)+7) mod 250	12
4	(11(12)+7) mod 250	139
5	(11(139)+7) mod 250	36
6	(11(36)+7) mod 250	153
7	(11(153)+7) mod 250	190
8	(11(190)+7) mod 250	97
9	(11(97)+7) mod 250	74
10	(11(74)+7) mod 250	71
11	(11(71)+7) mod 250	38
12	(11(38)+7) mod 250	175
13	(11(175)+7) mod 250	182
14	(11(182)+7) mod 250	9
15	(11(9)+7) mod 250	106
16	(11(106)+7) mod 250	173
17	(11(173)+7) mod 250	160
18	(11(160)+7) mod 250	17
19	(11(17)+7) mod 250	194
20	(11(194)+7) mod 250	141

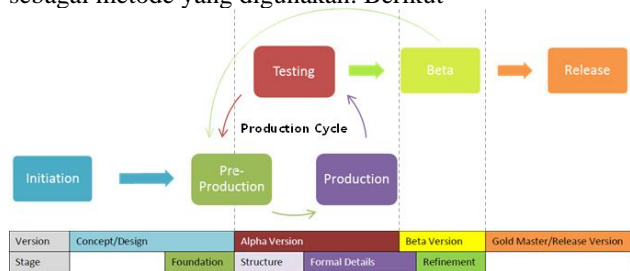
B. Unity

Unity adalah Game Engine lintas platform yang dikembangkan oleh Unity Technologies, pertama kali diumumkan dan dirilis pada juni 2005 di *Worldwide Developers Conference* milik Apple Inc. sebagai mesin permainan eksklusif Mac OS X[7]. Unity sendiri populer dikalangan pembuat video game seluruh dunia karena dapat menyajikan berbagaimacam keperluan dalam membuat game dan juga kemampuannya membuat berbagai macam tipe game baik 3D maupun 2D.

Perancangan sebuah game juga membutuhkan sebuah engine untuk mendukung pengembangan game[14]. Unity digunakan pada penelitian ini sebagai game engine selama proses pembuatan.

C. Game Development Life Cycle (GDLC)

Salah satu metode pengembangan yang sering digunakan adalah metode *Game Development Life Cycle* (GDLC) dimana dalam metode ini terdapat 6 tahapan pengembangan yang dilakukan diantaranya adalah, tahap inialisasi, tahap pra-produksi, tahap produksi, tahap tahap alpha, tahap beta dan versi rilis[15]. Pada penelitian kali ini digunakanlah *Game Development Life Cycle* (GDLC) sebagai metode yang digunakan. Berikut



Gambar 1. Game development life cycle (GDLC)

Sumber: <https://arifsetiawan.com/2016/01/game-development-life-cycle>

1) *Inialisasi*: Inialisasi adalah tahap perancangan serta pemikiran akan seperti apa game yang dibuat, target, permasalahan yang harus dihadapi serta juga bagaimana pembuatan game akan difokuskan.

2) *Pra-produksi*: Pada tahap ini dilakukan desain game yang akan dibuat, genre game, cara bermain, serta berbagai aspek lain yang mengarah pada perencanaan dari game tersebut.

Disusunlah rencana untuk membuat game bergenre *typing game* dengan tema kisah Candi Prambanan. Game akan didesain untuk menceritakan kisah Candi Prambanan dan membuat proses dalam pembangunan candi menjadi *typing game* dimana pemain harus mengetik secara terus-menerus dan berkompetisi dengan waktu demi membangun candi dan menyelesaikan permainan.

3) *Produksi*: Pada tahap produksi aset game akan mulai dibuat dan dikumpulkan. Game akan dibuat dengan mengikuti rencana dari pra-produksi dengan membuatnya memiliki genre *typing game* dan juga memiliki kisah Candi Prambanan.

4) *Tahap alpha*: pada tahap alpha, game akan di test oleh pembuat secara langsung untuk memastikan apakah sudah sesuai dengan yang direncanakan sebelumnya atau belum. Jika game belum sesuai ataupun masih terdapat error maka akan dikembalikan ke tahap pra-produksi untuk dicari solusi yang bisa diambil dan kembali dilanjutkan ke tahap produksi hingga kembali ke tahap alpha untuk di test kembali oleh pembuat.

Pengujian pada game yang dibuat untuk memastikan game bekerja dengan semestinya tanpa adanya masalah atau error saat digunakan. Game ini diharapkan dapat menjadi sarana untuk mempelajari cerita rakyat yang berkisah tentang candi prambanan, memiliki latar Indonesia namun juga dapat dimainkan dengan nyaman tanpa adanya masalah apapun selama memainkannya. Saat pertama membuka aplikasi, pemain akan masuk ke menu utama yang terdapat pilihan tombol main dan keluar. jika memilih tombol keluar maka program akan menutup, jika memilih tombol main maka akan masuk ke permainan. Saat memilih tombol main akan diceritakan reka adegan asal-usul mengapa konflik bisa terjadi antara karakter yang ada dengan musik yang membawa pemain agar merasakan konflik yang terjadi. Setelah test agar reka adegan dapat berjalan tanpa ada masalah dan berjalan sebagaimana mestinya maka pengujian berikutnya adalah bagaimana game tersebut dimainkan, game harus memiliki poin, waktu hitung mundur dan juga tujuan apa yang harus dilakukan. Tujuan yang harus dilakukan pada game tersebut adalah dengan mengetik kata yang ada dan tersedia dilayar, game harus dapat menghilangkan kata yang sudah diketik lalu memanggil kata secara terus-menerus dari array kumpulan kata yang telah disediakan pada array yang berisi kumpulan kata-kata dengan metode Linear Congruential Generator (LCG) secara terus menerus hingga waktu habis, jika waktu habis maka akan beralih ke level selanjutnya hingga ke level 3 dimana jika waktu habis maka akan dianggap kalah, namun jika berhasil menyelesaikan game sebelum waktu pada level 3 habis maka cerita akan berlanjut hingga ke akhir.

Pengujian inipun dilakukan berkali-kali dan setiap ada komponen yang diubah sekecil apapun itu pengujian akan dilakukan ulang untuk memastikan bahwa tidak ada bagian lain didalam game yang menjadi error karena ada perubahan kecil dibagian tertentu. Ini untuk memastikan game dapat dijalankan sebagaimana mestinya selama bermain demi mendapatkan pengalaman bermain yang maksimal. Jika dijadikan tabel maka akan seperti berikut:

TABEL III
PENGUJIAN MAIN MENU

Target Pengujian	Hasil Yang Diharapkan
Tombol Main	Membawa pemain ke scene cerita awal game
Tombol Exit	Menutup Game

TABEL IIIII
PENGUJIAN MAIN DIALOG

Target Pengujian	Hasil Yang Diharapkan
Tombol selanjutnya	Membawa ke dialog berikutnya dan ke scene berikutnya jika dialog sudah habis
Tombol spasi pada keyboard	Melaksanakan tombol selanjutnya jika ditekan
Tombol skip	1. Melewati setiap Scene hingga ke satu scene sebelum sebelum level selanjutnya 2. Membawa ke scene adegan terakhir jika semua level sudah dilewati

TABEL IVV
PENGUJIAN MAIN LEVEL

Target Pengujian	Hasil Yang Diharapkan
Waktu hitung mundur	Menghitung mundur dan beralih ke scene berikutnya ketika waktu habis.
Skor poin	A. Bertambah ketika mengetik huruf yang benar. B. berkurang ketika mengetik huruf yang salah akan beralih ke scene adegan akhir saat mencapai 999.
Candi	Akan muncul berdasarkan banyaknya poin yang didapat selama <i>game</i> berlangsung.
Kata Generator	Memanggil kata secara terus menerus dari array yang disediakan dengan algoritma <i>Linear congruential Generator (LCG)</i> ketika tidak ada kata untuk diketik.
Cuaca	1. Jika malam pertama akan cerah berawan 2. Jika malam kedua akan hujan 3. Jika malam ketiga akan berkebabut Dapat membuat animasi sederhana pada setiap cuaca

Jika pengujian sudah dilakukan dan setiap target pengujian dapat berfungsi sesuai yang diharapkan maka siap untuk diberikan kepada pemain yang dipilih untuk tahap rilis beta.

5) *Tahap Beta*: Pada tahap beta *game* akan dirilis kepada orang-orang tertentu untuk di test secara tertutup. Orang-orang yang dipilih akan dapat memainkan *game* ini dengan tujuan menemukan error dan menilai apakah *game* sudah dapat dirilis secara terbuka. Jika *game* belum dapat dirilis secara terbuka maka akan dikembalikan lagi ke tahap pra-produksi untuk dicari tindakan yang tepat untuk *game* agar dapat diperbaiki kekurangannya.

6) *Versi Rilis*: pada tahap ini *game* akan dirilis karena dinilai sudah layak untuk dimainkan. Setelah memainkan *game* yang dirilis, para pemain akan diminta untuk mengisi kuisisioner untuk menjawab beberapa pertanyaan mengenai *game* tersebut.

III. HASIL DAN PEMBAHASAN

A. Desain Level

Game di desain agar memiliki genre *typing game* (permainan mengetik) sebagai inti permainan yang membuat pemain harus terus mengetik untuk memainkan *game* ini dan memiliki latar kisah Candi Prambanan. Hal

ini dilakukan agar membuat *game* menjadi edukatif serta dapat memberikan cerita rakyat berupa Kisah Candi Prambanan kepada para pemain dari *game* ini.

Untuk memberikan level pada *game*, cerita dalam *game* akan sedikit diubah dari aslinya yaitu membuat Bandung Bondowoso dapat memutar balikan waktu. Membuat waktu satu malam menjadi tiga malam demi memberi level pada *game* tersebut. Setiap level akan memiliki cuaca pengumpulan poin serta cuaca yang berbeda-beda. Nantinya level akan berubah jika pemain gagal menyelesaikan permainan pada level tersebut hingga akhirnya akan pemain akan dinyatakan kalah jika pemain gagal menyelesaikan *game* pada level 3.

Berikut adalah tabel untuk rancangan level *game*:

TABEL V
PERUBAHAN DATA TIAP LEVEL

Level	Perubahan
Level 1	Cuaca berawan
	Waktu awal 3 menit
Level 2	2 poin didapat untuk setiap huruf yang benar
	Cuaca hujan
Level 3	1 poin didapat untuk setiap huruf yang benar
	Cuaca berawan
	Waktu awal 1 menit
	1 poin didapat untuk setiap huruf yang benar

Pada saat dimainkan jika mengetik huruf yang benar akan menambahkan poin dan jika mengetik huruf yang salah maka poin akan berkurang. Pemain harus mencapai poin 999 untuk dapat menyelesaikan permainan. Perubahan poin juga membuat perubahan pada dunia *game* berdasarkan berapa banyaknya poin yang dimiliki. Berikut ini adalah urutan dari candi yang ada pada *game*:



Gambar. 2 Penomoran pada candi

Candi-candi akan muncul satu persatu seiring meningkatnya poin dan poin tersebut memegang jalannya pembangunan candi dan penyelesaian *game* ini. Bagaimana tampilan tembus pandang candi juga dipengaruhi oleh poin-poin tersebut. Masing-masing candi akan muncul setiap 200 kata dan dari 200 kata tersebut setiap 20 poin akan menambah transparansi candi sebanyak 10%. Berikut ini adalah tabel tentang transparansi masing candi:

TABEL VI
POIN KEMUNCULAN CANDI

Poin	Kemunculan Candi
1 sampai 200	Candi nomor 1
201 sampai 400	Candi nomor 2
401 sampai 600	Candi nomor 3
601 sampai 800	Candi nomor 4
801 sampai 1.000	Candi nomor 5

Berikut ini adalah tabel yang menunjukkan transparansi candi setiap 200 poin:

TABEL VII
POIN TRANSPARANSI CANDI

Poin	Transparansi Candi
1 sampai 20	10%
21 sampai 40	20%
41 sampai 60	30%
61 sampai 80	40%
81 sampai 100	50%
101 sampai 120	60%
121 sampai 140	70%
141 sampai 160	80%
161 sampai 180	90%
181 sampai 200	100%

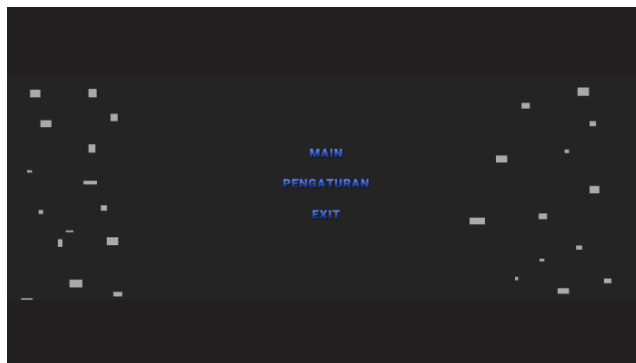
Jika diimplementasikan pada program maka akan didapatkan gambaran candi yang berubah dari semula tidak ada menjadi ada berdasarkan poin yang diketik oleh pemain. Seperti contoh pada gambar berikut memiliki poin 731 berarti candi 1 (poin 1 sampai 200), candi 2 (poin 201 sampai 400), dan candi 3 (poin 401 sampai 600) sudah terlewati yang menyebabkan candinya terlihat dan poin sedang berada pada candi 4 dimana masih memiliki transparansi 70% karena poinnya 731, sedangkan candi 5 belum terlihat karena poinnya masih dibawah angka 800.



Gambar. 3 Proses kemunculan dan transparansi pada candi

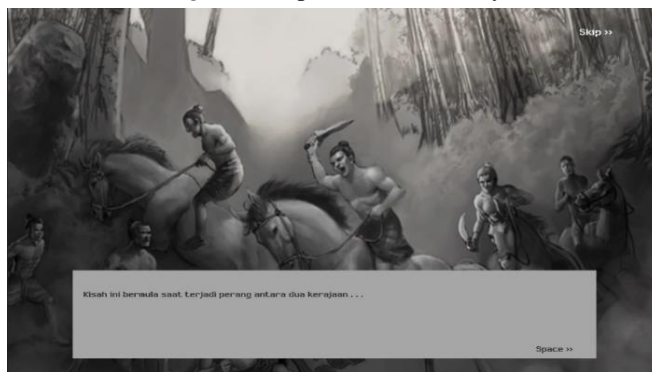
B. Tampilan Layar

Terdapat 2 tombol utama pada main menu. Tombol main akan membawa pemain ke dialog awal permainan sedangkan tombol exit akan menutup program.



Gambar. 4 Main menu

Dialog digunakan sebagai perantara untuk memberitahu isi dari kisah Candi Prambanan yang digunakan untuk tema *game*. Terdapat dua tombol pada dialog yaitu tombol untuk ke dialog selanjutnya seperti yang ada di bawah kanan kotak dialog yang bisa juga ditekan dengan menggunakan tombol spasi pada *keyboard* serta memiliki fungsi untuk melihat dialog berikutnya dan juga tombol *skip* yang bisa ditemukan di atas kanan layar yang berfungsi untuk melewati semua dialog lalu masuk kedalam tutorial *game* ataupun level berikutnya.



Gambar. 5 Dialog cerita

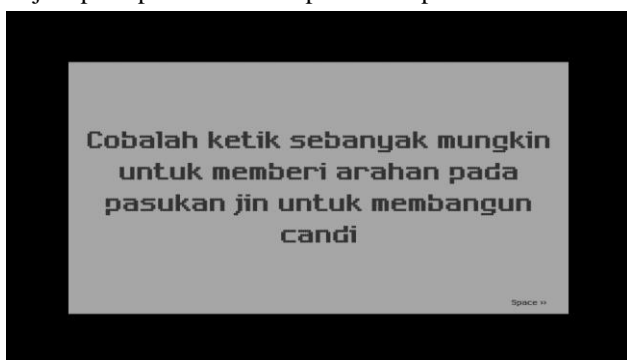


Gambar. 6 Dialog percakapan Bandung Bondowoso



Gambar. 7 Dialog perancangan Roro Jonggrang

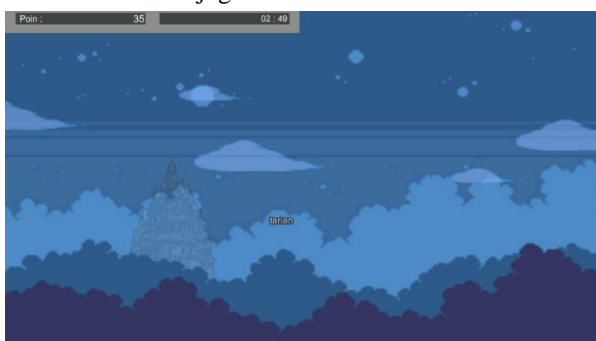
Dialog tutorial berfungsi untuk memberi arahan tentang apa yang harus dilakukan sebelum permainan dimulai. Dialog tutorial juga akan memberitahu apa saja yang akan terjadi pada permainan setiap sebelum permainan dimulai.



Gambar.8 Dialog tutorial

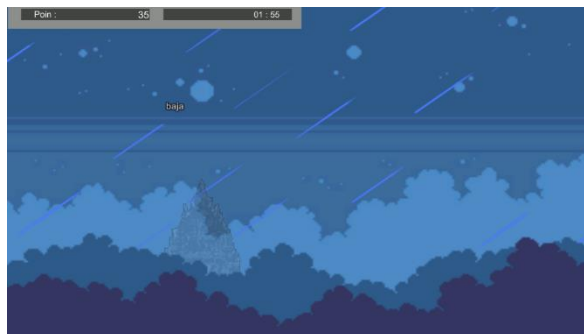
Tampilan layar permainan akan berisi animasi sederhana tentang cuaca yang sedang terjadi. Tampilan layar permainan juga berisi informasi mengenai skor dan juga waktu hitung mundur. Disini pemain akan mulai memainkan *game* dengan cara mengetik.

Berikut ini adalah tampilan layar pada level 1 dengan cuaca berawan dan juga waktu awal mulai adalah 3 menit:



Gambar. 9 Cuaca berawan pada level 1

Berikut ini adalah tampilan layar pada level 2 dalam keadaan hujan dan waktu hitung muncul awal yang didapat adalah 2 menit:



Gambar. 10 Cuaca hujan pada level 10

Berikut ini adalah tampilan layar pada level 3 dengan cuaca berkabut dan waktu awal yang diberikan adalah 1 menit:



Gambar. 11 Cuaca berawan pada level 3

C. Hasil Uji Coba Oleh Pemain

Setelah *game* selesai dibuat, dilakukan uji coba kepada pengguna. Terdapat 6 mahasiswa yang menjadi pemain dan melakukan uji coba *game* ini. *Game* dibagikan kepada pemain dan mereka menggunakan laptop / komputer milik pribadi. Setelah para pemain menyelesaikan *game*, berikutnya mereka mengisi kuesioner yang sudah dibuat.

Kuesioner dibuat menggunakan skala Guttman dengan dua pilihan jawaban yaitu Setuju dan Tidak Setuju [16]–[18]. Terdapat sepuluh pertanyaan pada kuesioner yang dibagikan. Sebelum dibagikan, kuesioner diuji terlebih dahulu validitasnya. Pengujian dilakukan terhadap lima orang pakar, ditanyakan apakah pertanyaan/ pernyataan kuesioner cukup sesuai dengan tujuan pengumpulan data.

TABEL VIII
PENGUJIAN KUESIONER

Pertanyaan	Sesuai	Tidak Sesuai
Game ini tidak memberatkan komputer saat dijalankan	5	0
Nilai kebudayaan sangat terasa pada game	5	0
Kontrol pada game tergolong mudah	5	0
Game ini dapat diselesaikan dengan cepat	3	2
Game mudah diselesaikan	4	1
Alur cerita pada game mudah dimengerti	5	0
Anda mengerti alur cerita game tersebut	5	0
Anda sangat menikmati game tersebut	5	0
Game tersebut sudah layak untuk dimainkan	5	0
Anda tertarik untuk memainkan game dengan genre typing game?	4	1

Berdasarkan perhitungan Skala Guttman, maka didapat total kemungkinan jawaban yang tersedia adalah 5 (pakar/responden) dikalikan 10 (pertanyaan) = 50. Total error (jawaban tidak sesuai) yang terjadi pada Tabel VIII adalah 4. Jumlah pertanyaan/pernyataan yang memiliki jawaban sesuai 100% adalah 7. Apakah kesalahan dari kemungkinan masih dianggap dapat diterima, ini perlu dicari koefisien reproduibilitas dan skalabilitasnya. Koefisien reproduibilitas dihitung dengan rumus:

$$K_r = 1 - \frac{e}{n}$$

Keterangan:

K_r = Koefisien reproduibilitas

e = Jumlah kesalahan / nilai error

n = Jumlah pertanyaan dikali jumlah responden

$$K_r = 1 - \frac{4}{50} = 0,92$$

Karena $K_r > 0,9$ dianggap baik maka skala Gutman diatas dengan $K_r = 0,92$ dianggap baik untuk digunakan. Berikutnya dicari koefisien skalabilitas yang dihitung dengan rumus:

$$K_s = 1 - \frac{e}{p}$$

Keterangan:

K_s = Koefisien skalabilitas

e = Jumlah kesalahan / nilai error

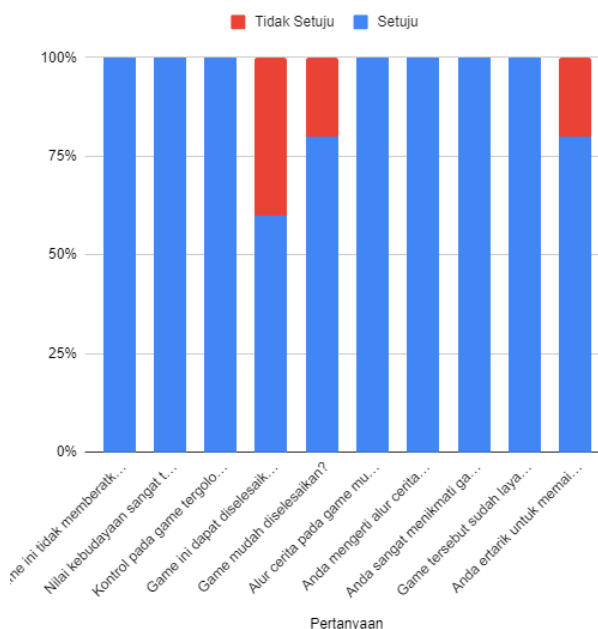
p = $0,5 \{ \text{jumlah pertanyaan dikali jumlah responden} \} - \text{Jumlah Jawaban "Ya"}$

$$K_s = 1 - \frac{4}{0,5 * (50 - 7)} = 0,81$$

Skala dengan nilai $K_s > 0,6$ dianggap baik, maka koefisien skalabilitas dari kuesioner ini adalah $K_s = 0,81$ dianggap baik untuk digunakan.

Setelah para pemain menyelesaikan game dan mengisi kuesioner, didapatlah hasil seperti terlihat pada Gambar 12. Mayoritas responden setuju dengan pernyataan-pernyataan yang diajukan dalam survei. Angka "Setuju" lebih tinggi daripada "Tidak Setuju" pada semua pernyataan. Game ini tidak memberatkan komputer saat dijalankan dan memiliki nilai kebudayaan yang terasa, hal ini mendapatkan respon positif dari semua responden. Kontrol pada game dinilai sangat mudah dan alur cerita mudah dimengerti oleh semua responden.

Mayoritas responden menyatakan bahwa mereka mengerti alur cerita game tersebut, menunjukkan bahwa narasi game tersebut berhasil disampaikan dengan jelas. Semua responden menikmati game ini dan menganggapnya layak untuk dimainkan, menunjukkan kualitas dan daya tarik dari game tersebut. Terdapat beberapa responden yang menyatakan ketertarikan lebih rendah terhadap genre game ini, namun mayoritas tetap tertarik.



Gambar. 12 Hasil kuesioner

D. Masukan Dari Pemain

Setelah melakukan kuisisioner ada lagi hal yang dilakukan. yaitu adalah mendapatkan saran dan masukan yang didapat dari pengguna setelah game ini dimainkan. Tahap ini berguna sebagai acuan jika ada penelitian lain yang dibuat di masa mendatang sebagai hal-hal yang harus dipertimbangkan. Beberapa masukan dari pemain yang didapat adalah sebagai berikut:

1. Banyak bagian yang bisa dikembangkan.
2. Gambar masih beresolusi rendah dan bisa ditingkatkan.
3. Waktu hitung mundur dipersingkat lagi.

IV. KESIMPULAN

Game berhasil dibuat dan dapat dimainkan dengan baik. Berdasarkan data Kuesioner, *game* ini mendapatkan tanggapan positif secara umum. Mayoritas responden setuju bahwa *game* ini tidak memberatkan komputer saat dijalankan, memiliki nilai kebudayaan berupa kisah Candi Prambanan yang terasa, serta alur cerita dan kontrol yang mudah dimengerti. Kesenangan dalam bermain *game* ini juga tercermin dari tingginya angka responden yang menikmati *game* dan menganggapnya layak dimainkan.

Meskipun mayoritas tanggapan positif, perlu diperhatikan bahwa beberapa responden memiliki pandangan berbeda terkait kecepatan menyelesaikan *game* dan daya tarik genre. Saran pengembangan berikutnya perlu mempertimbangkan penyesuaian dalam desain *game* untuk mengatasi perbedaan ini, seperti menambahkan kecerdasan buatan untuk variasi dalam tingkat kesulitan atau mengeksplorasi elemen *gameplay* yang lebih menarik. Dengan demikian, pengembang dapat lebih memaksimalkan potensi *game* ini dan menghadirkan pengalaman bermain yang lebih komprehensif bagi semua pemain.

REFERENSI

- [1] M. W. Pamengas, T. Afirianto, and W. S. Wardhono, "Pengembangan gim edukasi mengetik menggunakan metode touch-typing untuk meningkatkan kecepatan mengetik," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, 2020.
- [2] A. Sonita and R. E. Fitri, "Aplikasi Cerdas Cermat Menggunakan Algoritma Linear Congruential Generator Berbasis Android," *Jukomika - (Jurnal Ilmu Komput. Dan Inform.*, vol. 3, pp. 541–548, 2020, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jukomika>
- [3] R. M. M. Prasetyo, H. Syaputra, W. Cholil, and S. Sauda, "Rancang Dan Bangun *Game* Edukasi Anak-Anak Berbasis Android Dengan Unity Menggunakan Metode *Game* Development Life Cycle," *J. Nas. Ilmu Komput.*, 2021, doi: 10.47747/jurnalnik.v2i2.526.
- [4] A. S. Rahman and I. G. Anugrah, "Pengembangan *Game* Edukasi Pencegahan COVID-19 Menggunakan Metode *Game* Development Life Cycle," *Ilk. J. Comput. Sci. Appl. Informatics*, 2022, doi: 10.28926/ilkomnika.v4i1.446.
- [5] R. Nurcholis, A. I. Purnamasari, A. R. Dikananda, O. Nurdiawan, and S. Anwar, "*Game* Edukasi Pengenalan Huruf Hiragana Untuk Meningkatkan Kemampuan Berbahasa Jepang," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 338–345, 2021, doi: 10.47065/bits.v3i3.1091.
- [6] R. Royani and U. Darusalam, "Media Pembelajaran Smart Quiz Berbasis Android menggunakan Metode Congruential Generator," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 2, pp. 419–428, 2022, doi: 10.29100/jupi.v7i2.2767.
- [7] M. A. Adilla and P. Windriyani, "Pengembangan Typing *Game* Sebagai Sarana Melatih Meningkatkan Kecepatan Mengetik Berbasis Desktop," *KALBISIANA J. Mhs. Inst. Teknol. dan Bisnis Kalbis*, vol. 8, no. 1, pp. 400–414, 2022, [Online]. Available: <http://ojs.kalbis.ac.id/index.php/kalbisiana/article/view/277>
- [8] R. A. Krisdiawan, M. F. Rohmana, and A. Permana, "Pembuatan *Game* Runaway From Culik Dengan Algoritma Fuzzy Mamdani," *Buffer Inform.*, 2020.
- [9] M. F. Bhinekas, B. Dirgantoro, and A. S. Raharjo, "PENGEMBANGAN PERMAINAN EDUKASI PENGENALAN JENIS HEWAN UNTUK SISWA SEKOLAH DASAR DENGAN ALGORITMA PERMUTED CONGRUENTIAL GENERATOR UNTUK PENGACAKAN SOAL," in *eProceedings of Engineering*, 2020, vol. 7, no. 1, pp. 1554–1561.
- [10] R. A. Krisdiawan, H. Budianto, T. Sutabri, and A. Kurniawan, "Implementasi Algoritma Linear Congruent Method (LCM) Pada Media Pembelajaran Bagian-Bagian Bunga Berbasis Virtual Reality (VR)," *Nuansa Inform.*, vol. 16, no. 2, pp. 94–105, 2022, doi: 10.25134/nuansa.v16i2.5899.
- [11] R. Prasetyadi, N. Budi Nugroho, and Azlan, "Implementasi Metode Multiplicative Random Number Generator (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer," *J. CyberTech*, vol. 3, no. 2, pp. 224–229, 2020, [Online]. Available: <https://ojs.trigunadharma.ac.id/>
- [12] A. Prakarsa, A. A. Sunarto, and P. Prajoko, "Model Pengacakan Soal Ujian Online SMA Menggunakan Metode Linear Congruential Generator dan Fisher Yates," *Progresif J. Ilm. Komput.*, 2020, doi: 10.35889/progresif.v16i2.519.
- [13] R. A. Krisdiawan, "Implementasi Model Pengembangan Sistem Gdlc Dan Algoritma Linear Congruential Generator Pada *Game* Puzzle," *Nuansa Inform.*, 2018.
- [14] S. Syarif, T. Hasanuddin, and M. Hasnawi, "Perancangan *Game* Puzzle Labirin menggunakan Metode *Game* Development Life Cycle (GDLC) berbasis Unreal Engine," *Bul. Sist. Inf. dan Teknol. Islam*, 2022, doi: 10.33096/busiti.v3i1.582.
- [15] R. Yanwastika Ariyana, E. Susanti, M. Rizqy Ath-Thaariq, and R. Apriadi, "Penerapan Metode *Game* Development Life Cycle (GDLC) pada Pengembangan *Game* Motif Batik Khas Yogyakarta," *J. Sains dan Teknol.*, vol. 1, no. 6, 2022.
- [16] Suranto, M. Musrofi, and A. Widodo, "Analisis Kepuasan Konsumen Dengan Skala Guttman," *J. Ilm. Tek. Ind.*, 2004.
- [17] M. Nazir, *Metode Penelitian*. Bogor: Ghalia Indonesia, 2009.
- [18] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta, 2013.