

LAPORAN PROYEK AKHIR

**PEMBUATAN GAME CODING PUZZLE 2D
UNTUK MAHASISWA POLITEKNIK CALTEX
RIAU (PCR)**

Fanny

NIM. 2055301044

Pembimbing

Maksum Ro'is Adin Saf, S.Kom., M.Eng.

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK CALTEX RIAU

2024



LAPORAN PROYEK AKHIR

**PEMBUATAN GAME CODING PUZZLE 2D
UNTUK MAHASISWA POLITEKNIK CALTEX
RIAU (PCR)**

Fanny

NIM. 2055301044

Pembimbing

Maksum Ro'is Adin Saf, S.Kom., M.Eng.

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK CALTEX RIAU

2024

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

PEMBUATAN GAME CODING PUZZLE 2D UNTUK
MAHASISWA POLITEKNIK CALTEX RIAU (PCR)

Fanny
NIM. 2055301044

Proyek Akhir ini diajukan sebagai salah satu persyaratan untuk
memperoleh gelar Sarjana Terapan Komputer (S Tr Kom)
di Politeknik Caltex Riau

Pekanbaru, 10 Juli 2024

Disetujui oleh:

Pembimbing,

1. Maksum Roris Adin Saf,
S.Kom., M.Eng.
NIP. 158803

Penguji,

1. Meilany Dewi, S.T., M.T.
NIP. 048009

2. Jan Alif Kreshna, S.ST., M.S
NIP. 159007

Mengetahui,

Ketua Program Studi Teknik Informatika

Silvana Rasio Herim, S.S.T., M.T.
NIP. 068407

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam proyek akhir yang berjudul:

“Pembuatan Game Coding Puzzle 2D Untuk Mahasiswa Politeknik Caltex Riau (PCR)”

Adalah benar hasil karya saya, dan tidak mengandung karya ilmiah atau tulisan yang pernah diajukan di suatu Perguruan Tinggi.

Setiap kata yang dituliskan tidak mengandung plagiat, pernah ditulis maupun diterbitkan orang lain kecuali yang secara tertulis diacu dalam laporan proyek akhir ini dan disebutkan pada daftar pustaka. Saya siap menanggung seluruh akibat apabila terbukti melakukan plagiat.

Pekanbaru, 10 Juli 2024

Fanny

ABSTRAK

Pada awal semester, mahasiswa Politeknik Caltex Riau, khususnya yang mengambil jurusan Teknologi Informasi akan berhadapan dengan mata kuliah pemrograman. Dimana tidak semua mahasiswa PCR memiliki dasar-dasar pemrograman ketika masuk ke prodi Teknik Informatika, dan hasilnya ketika mengikuti mata kuliah pemrograman tidak sedikit yang merasa kesulitan dalam memahami konsep pemrograman di perkuliahan. Dari permasalahan masalah tersebut, telah dibangun sebuah game coding puzzle yang mana game ini telah diujicobakan kepada mahasiswa PCR Teknik Informatika untuk mengenalkan dan menguji kemampuan pemrograman mahasiswa sebagai media pembelajaran baru yang lebih menarik dimata mahasiswa PCR. Lalu, selain untuk mahasiswa yang baru belajar pemrograman, game ini juga dapat ditujukan untuk mahasiswa yang sudah memiliki dasar pemrograman namun ingin mengasah kembali kemampuannya. Game ini dikembangkan menggunakan Unity sebagai game enginenya. Selain itu, game ini menggunakan metode Game Development Life Cycle (GDLC) dalam pengembangannya, dengan mengikuti tahapan-tahapan GDLC telah dibuat model game yang sudah sesuai dengan modul perkuliahan algoritma dan pemrograman. Hasil implementasi memperlihatkan bahwa game coding puzzle ini telah memotivasi mahasiswa sebanyak 93% untuk belajar pemrograman dirumah dan dapat meningkatkan pemahaman konsep-konsep dasar pemrograman mahasiswa sebesar 80%.

Kata Kunci: Edukasi, Game, GDLC, Unity

ABSTRACT

In the new semester, Politeknik Caltex Riau students, specifically the ones who chose an Information Technology (ITI) major will learn programming courses. This is when the problem arises, not all PCR students have programming basics when joining the Informatics Engineering study program, and as a result, when they attend the courses, a lot of them have trouble learning the programming concept in lectures. From that problem, a game about coding with puzzle mechanics is created that has already been tested by PCR students to introduce and test students' programming skills as a new learning medium that's more interesting in the student's eyes. Then, besides the students who are new to programming, this game also can be played by the students who already know programming basics to sharpen their skills in programming. This game is developed using the Unity engine. And, this game is also using Game Development Life Cycle (GDLC) as a method in the development, following the GDLC steps, the model game is already matched with the algorithm and programming lectures module. The implementation result shows that the coding puzzle game motivates 93% students to learn programming at home and can increase 80% student's knowledge about basic programming concepts.

Keywords: Education, Game, GDLC, Unity

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan proyek akhir yang merupakan salah satu syarat untuk menyelesaikan jenjang pendidikan Diploma IV pada Program Studi Teknik Teknik Informatika di Politeknik Caltex Riau.

Dalam penyelesaian proyek akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan dan dukungan baik secara langsung maupun tidak langsung. Untuk itu penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan proyek akhir ini.
2. Orang tua yang selalu mendoakan dan memberikan motivasi kepada penulis.
3. Ibu Silvana Rasio Henim, S.T., M.T. selaku kepala Program Studi Teknik Informatika, Politeknik Caltex Riau.
4. Bapak Maksum Ro'is Adin Saf, S.Kom., M.Eng. selaku dosen pembimbing proyek akhir yang telah membimbing dan memberi bantuan untuk menyelesaikan proyek akhir.
5. Ibu Meilany Dewi, S.T., M.T. dan Bapak Jan Alif Kreshna, S.ST., M.Sc. selaku dosen penguji proyek akhir yang membantu dan memberikan masukan terhadap proyek akhir.
6. Ibu Meilany Dewi, S.T., M.T. selaku wali dosen dan koordinator proyek akhir.
7. Albes Cotois dan Tony Wijaya yang telah membantu dalam pengerjaan proyek akhir.
8. Dosen-dosen Politeknik Caltex Riau Teknik Informatika yang telah memberikan bekal ilmu kepada penulis selama perkuliahan dalam menyelesaikan proyek akhir.
9. Teman-teman kelas TI D G20 seperjuangan yang saling memberikan dukungan dan motivasi sehingga penulis dapat menyelesaikan proyek akhir.
10. Teman-teman G20-G23 yang berpartisipasi dalam kuesioner permasalahan dan pengujian proyek akhir sehingga penulis dapat menyelesaikan proyek akhir.

Akhir kata, penulis berharap bahwa laporan ini dapat bermanfaat bagi pembaca. Kemudian, penulis menyadari bahwa laporan akhir yang ditulis

ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran juga diharapkan demi perbaikan laporan ini untuk kedepannya.

Pekanbaru, 10 Juli 2024

Fanny

DAFTAR ISI

PERNYATAAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat	2
1.4.1 Tujuan.....	2
1.4.2 Manfaat.....	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Landasan Teori.....	7
2.2.1 Game Edukasi.....	7
2.2.2 Game Development Life Cycle (GDLC).....	8
2.2.3 Android.....	9
2.2.4 Algoritma dan Pemrograman.....	9
2.2.5 Game Puzzle	10
2.2.6 Unity	11
2.2.7 User Acceptance Testing (UAT)	11
2.2.8 Validasi Konten	12
2.2.9 Pre-test dan Post-test	12
BAB III PERANCANGAN.....	13
3.1 Initiation	13
3.1.1 Ide dan Konsep Game.....	13
3.1.2 Storytelling	14
3.2 Pre-Production.....	15

3.2.1	Kebutuhan Pengguna.....	15
3.2.2	Arsitektur Sistem.....	16
3.2.3	Flowchart Game Coding Puzzle.....	16
3.2.4	Use Case Game Coding Puzzle.....	18
3.2.4.1	Usecase Description.....	18
3.2.4.2	Skenario Mulai Game.....	20
3.2.4.3	Skenario Memilih Mode Kesulitan.....	20
3.2.4.4	Skenario Mengakses Materi.....	21
3.2.4.5	Skenario Mengikuti Test.....	21
3.2.4.6	Skenario Melihat Tentang Game.....	21
3.2.4.7	Skenario Keluar Game.....	22
3.3	Production.....	22
3.3.1	Prototype Game Coding Puzzle.....	22
3.4	Testing.....	27
3.4.1	User Acceptance Testing (UAT).....	27
3.4.2	Validasi konten.....	27
3.4.3	Pengujian responden (Pre-test dan Post-test).....	27
3.4.4	Kuesioner kepuasan pengguna.....	28
BAB IV PENGUJIAN DAN ANALISIS.....		29
4.1	Production.....	29
4.1.1	Proses Pembuatan Asset.....	29
4.1.2	Desain Antarmuka Game.....	30
4.1.3	Script Game Code Arcade.....	39
4.2	Pengujian.....	50
4.2.1	User Acceptance Testing (UAT).....	50
4.2.2	Beta Testing.....	54
4.2.3	Pengujian Validasi Konten.....	55
4.2.4	Pengujian Responden (Pre-test dan Post-test).....	57
4.2.5	Pengujian Kuesioner kepuasan pengguna.....	59
4.3	Analisis.....	62
4.3.1	Analisis User Acceptance Testing (UAT).....	62
4.3.2	Analisis Validasi Konten.....	62
4.3.3	Analisis Responden(Pre-test dan Post-test).....	62
4.3.4	Analisis Kepuasan Pengguna.....	63
4.4	Release.....	63
BAB V KESIMPULAN DAN SARAN.....		64
5.1	Kesimpulan.....	64
5.2	Saran.....	64

DAFTAR PUSTAKA.....	65
LAMPIRAN A – KUESIONER PERMASALAHAN	1
LAMPIRAN B – PUZZLE.....	1
LAMPIRAN C – VALIDASI KONTEN	1
LAMPIRAN D – PENGUJIAN UAT, RESPONDEN, KUESIONER KEPUASAN.....	1

DAFTAR GAMBAR

Gambar 2. 1 Fase dan Proses GDLC (Sumber : www.semanticscholar.org)	8
Gambar 3. 1 Grafik Mahasiswa Dengan Dasar Pemrograman	13
Gambar 3. 2 Grafik Mahasiswa Kesulitan Belajar	14
Gambar 3. 3 Grafik Minat Mahasiswa Mengikuti Pelajaran	14
Gambar 3. 4 Arsitektur Sistem	16
Gambar 3. 5 Flowchart Game Coding Puzzle	16
Gambar 3. 6 Use Case Game Coding Puzzle	18
Gambar 3. 7 Desain Antarmuka Awal.....	22
Gambar 3. 8 Desain antarmuka memilih mode kesulitan.....	23
Gambar 3. 9 Desain antarmuka menu module	23
Gambar 3. 10 Desain antarmuka materi	24
Gambar 3. 11 Desain antarmuka puzzle code completion	24
Gambar 3. 12 Desain antarmuka puzzle bug fixing	25
Gambar 3. 13 Desain antarmuka puzzle predict output.....	25
Gambar 3. 14 Desain antarmuka puzzle arrange code	26
Gambar 3. 15 Desain antarmuka halaman result.....	26
Gambar 4. 1 Sketching	29
Gambar 4. 2 Finalisasi	30
Gambar 4. 3 Tampilan Memilih Bahasa.....	30
Gambar 4. 4 Tampilan menu awal.....	31
Gambar 4. 5 Tampilan memilih mode kesulitan	31
Gambar 4. 6 Tampilan menu module easy	32
Gambar 4. 7 Tampilan menu module medium	33
Gambar 4. 8 Tampilan menu module hard	34
Gambar 4. 9 Tampilan halaman materi	35
Gambar 4. 10 Tampilan halaman test.....	36
Gambar 4. 11 Tampilan halaman result.....	37
Gambar 4. 12 Tampilan Halaman Practice.....	38
Gambar 4. 13 Tampilan tentang game.....	38
Gambar 4. 14 Script perpindahan scene di halaman pemilihan bahasa.....	39
Gambar 4. 15 Script Locale Selector.....	39
Gambar 4. 16 Script Bermain Game	40
Gambar 4. 17 Script Memilih Kesulitan.....	40
Gambar 4. 18 Script Game Manager	42
Gambar 4. 19 Script Membaca Materi	43

Gambar 4. 20 Script Quiz Manager.....	46
Gambar 4. 21 Script Hasil Pemain	48
Gambar 4. 22 Script Practice Manager.....	49
Gambar 4. 23 Gambar sebelum beta testing.....	54
Gambar 4. 24 Gambar setelah beta testing	54
Gambar 4. 25 Tampilan Penambahan Halaman Practice	55
Gambar 4. 26 Rata-rata waktu pre-test dan post-test.....	59
Gambar 4. 27 Grafik Kepuasan Pengguna	61
Gambar 4. 28 Website Instalasi Game coding puzzle yang direlease ...	63

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu.....	6
Tabel 3. 1 Tabel Kebutuhan Pengguna.....	15
Tabel 3. 2 Use Case Mulai Game	18
Tabel 3. 3 Use Case Memilih Mode Kesulitan	18
Tabel 3. 4 Use Case Mengikuti Pre-test	19
Tabel 3. 5 Use Case Mengakses Materi	19
Tabel 3. 6 Use Case Mengikuti Test.....	19
Tabel 3. 7 Use Case Melihat Tentang Game	20
Tabel 3. 8 Use Case Keluar Game.....	20
Tabel 3. 9 Skenario Mulai Game.....	20
Tabel 3. 10 Skenario Memilih Mode Permainan.....	21
Tabel 3. 11 Skenario Mengakses Materi	21
Tabel 3. 12 Skenario Mengikuti Test	21
Tabel 3. 13 Skenario Melihat Tentang Game	22
Tabel 3. 14 Skenario Keluar Game	22
Tabel 3. 15 Tabel Kuesioner Kepuasan.....	28
Tabel 4. 1 Pengujian Halaman Utama	50
Tabel 4. 2 Pengujian halaman mode kesulitan	51
Tabel 4. 3 Pengujian Halaman Module	51
Tabel 4. 4 Pengujian Halaman Materi	51
Tabel 4. 5 Pengujian Halaman Practice.....	52
Tabel 4. 6 Pengujian Halaman Test/Pre-test	52
Tabel 4. 7 Pengujian Halaman Result	53
Tabel 4. 8 Pengujian Halaman Tentang.....	53
Tabel 4. 9 Validasi Konten Section Variable	55
Tabel 4. 10 Validasi Konten Section Operator.....	55
Tabel 4. 11 Validasi Konten Section Input/Output	56
Tabel 4. 12 Validasi Konten Section Control Statement.....	56
Tabel 4. 13 Validasi Konten Section Array.....	56
Tabel 4. 14 Validasi Konten Section Exception Handling	56
Tabel 4. 15 Tabel perbandingan hasil pre-test dan post-test	57
Tabel 4. 16 waktu pengerjaan pre-test dan post-test.....	58
Tabel 4. 17 Rata-rata waktu pengerjaan	59
Tabel 4. 18 Kuesioner kepuasan pengguna	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap tahun, Politeknik Caltex Riau membuka pendaftaran untuk mahasiswa baru. Politeknik Caltex Riau memiliki berbagai macam program studi, salah satunya adalah Teknik Informatika. Seluruh mahasiswa baru yang memasuki program studi Teknik Informatika akan mendapatkan pembelajaran pemrograman dasar pada mata kuliah algoritma dan pemrograman. Pemrograman adalah suatu penulisan, pengujian serta memperbaiki dan memelihara tanda-tanda pembuatan sebuah program. (Riyanti & Yahfizham, 2023).

Berdasarkan hasil kuesioner yang dilakukan terhadap mahasiswa generasi 20 sampai generasi 23 jurusan Teknik Informatika di Politeknik Caltex Riau, 68.3% dari keseluruhan jumlah pengisi kuesioner belum memiliki pengetahuan dasar terkait pemrograman sebelum memasuki prodi Teknik Informatika. Dan 70% mahasiswa mengalami kesulitan dalam memahami konsep pemrograman selama kuliah.

Berdasarkan permasalahan tersebut, maka pada penelitian ini telah dibuat sebuah game edukasi yang digunakan untuk memudahkan mahasiswa dalam mengenal dan mengasah kemampuan pemrograman. Game ini dapat diterapkan pada saat pembelajaran mata kuliah algoritma dan pemrograman sebagai media pembelajaran yang lebih interaktif. Game yang dibuat berbasis mobile untuk mempermudah mahasiswa dalam memainkan game, dimana game ini dibuat menggunakan metode Game Development Life Cycle (GDLC).

Pembuatan game ini didasari oleh penelitian yang mengatakan bahwa salah satu metode yang ditemukan efektif membantu pelajar mempelajari pemrograman adalah dengan bermain game, yang merupakan salah satu produk yang dihasilkan dari pemrograman itu sendiri. Game yang dimaksud adalah permainan digital yang dirancang untuk membantu siswa meningkatkan coding skill dengan konteks yang menghibur (Hartanto et al., 2020). Berdasarkan penelitian terdahulu terkait pengaruh game edukasi dalam pembelajaran, didapat bahwa game edukasi yang dikembangkan telah memproduksi hasil yang memuaskan kebutuhan (Syaliman et al., 2022).

Dimana, pembuatan game ini telah berhasil dibangun dengan

metode GDLC dan di buat dengan format aplikasi yang dapat dipasang di perangkat android. Game ini juga terbukti meningkatkan minat belajar mahasiswa terutama dalam hal pemrograman yang mana mereka suka belajar sambil bermain melalui game yang mencapai angka 100% dan mereka tertarik untuk membiasakan penggunaan belajar pemrograman melalui game ini dirumah, dimana hasilnya mencapai angka 93%. Lalu, game ini dapat meningkatkan nilai dan pengetahuan mereka terkait materi pemrograman, dapat dilihat berdasarkan hasil survey perbandingan hasil sebelum dan setelah bermain yang mana 80% mahasiswa mengalami peningkatan secara keseluruhan.

1.2 Perumusan Masalah

Di Politeknik Caltex Riau, banyak mahasiswa yang belum memiliki pengetahuan dasar sebelum memasuki perkuliahan di program studi Teknik Informatika, sehingga menyebabkan beberapa mahasiswa mengalami kesulitan dalam memahami konsep pemrograman di perkuliahan.

1.3 Batasan Masalah

Adapun batasan masalah dalam pembuatan proyek akhir ini adalah:

- 1) Game yang dikembangkan hanya mengikuti kurikulum matakuliah algoritma dan pemrograman dari PCR
- 2) Game yang dikembangkan menggunakan materi dari modul Algoritma Pemrograman PCR
- 3) Game ditujukan untuk mahasiswa yang mempelajari mata kuliah pemrograman
- 4) Game engine yang digunakan dalam pembuatan game adalah Unity, dengan bahasa pemrograman yang digunakan adalah C#
- 5) Game yang dibangun berbasis mobile

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun tujuan dari penelitian ini ialah membangun coding

puzzle 2D matakuliah pemrograman untuk mahasiswa Politeknik Caltex Riau berbasis Android.

1.4.2 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

- 1) Membantu mahasiswa dalam mempelajari dasar pemrograman
- 2) Meningkatkan pemahaman pemrograman mahasiswa PCR, terutama yang belum memiliki dasar pemrograman sebelumnya

1.5 Metodologi Penelitian

Metode penelitian yang dipakai dalam pembuatan proyek akhir ini adalah:

- 1) Pembelajaran literatur
Pembelajaran literatur dilakukan dengan melakukan studi pustaka dengan mempelajari buku-buku dan mengunjungi situs-situs internet yang mendukung proyek akhir.
- 2) Pengujian Instrumen
Pengujian instrumen yang digunakan adalah kuesioner, dimana kuesioner digunakan untuk mengumpulkan data yang dibutuhkan pada pembuatan game.
- 3) Perancangan Game
Mengembangkan desain konsep game coding puzzle 2D, termasuk perencanaan tingkat, aliran permainan, dan elemen-elemen pembelajaran yang akan diimplementasikan.
- 4) Implementasi Game
Mengembangkan prototipe awal game coding puzzle 2D berdasarkan desain konsep yang telah dibuat. Kemudian mengembangkan game menggunakan Unity engine dan modelling asset menggunakan figma dan clip studio paint.
- 5) Pengujian Game
Pada tahap ini, game akan diuji sehingga dapat diketahui jika terdapat beberapa kebutuhan yang tidak berfungsi dengan baik

1.6 Sistematika Penulisan

Sistematika penulisan laporan proyek akhir ini secara keseluruhan terdiri dari empat bab, masing-masing terdiri dari beberapa sub bab. Adapun pokok pembahasan dari masing-masing bab tersebut secara garis besar sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, perumusan masalah dan ruang lingkup masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan beberapa hasil penelitian terdahulu dan landasan teori yang diperlukan untuk merancang sistem. Penelitian terdahulu yang diambil ada 4 jenis dengan tema yang berbeda-beda yang masih berhubungan dengan judul yang diambil. Terdapat juga tabel perbandingan yang melihat secara garis besar perbedaan dari setiap penelitian. Kemudian ada landasan teori yang terdiri dari semua teori yang terdapat dalam pengembangan game.

BAB III PERANCANGAN

Bab ini menjelaskan tentang perancangan sistem terdiri dari perancangan sistem yang akan dibangun. Pada bab ini, perancangannya menyesuaikan metode yang dipilih yaitu metode GDLC yang terdiri dari initiation, pre-production, production dan testing.

BAB IV PENGUJIAN DAN ANALISIS

Bab ini berisi bukti hasil dari pengujian pengerjaan dan analisis dari proyek akhir

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan hasil dari proyek akhir yang telah dikerjakan mulai dari hasil pengujian dan analisis sistem. Terdapat saran untuk melihat kekurangan dari proyek akhir.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu terkait pengembangan game edukasi pernah dilakukan oleh (Rohmawati & Menarianti, 2019). Penelitian tersebut berjudul “Pengembangan Game Edukasi Tentang Budaya Nusantara “Tanara” Menggunakan Unity 3d Berbasis Android”. Pada penelitian ini, game dikembangkan menggunakan unity 3D dengan mengembangkan game edukasi pengenalan budaya Nusantara. Game ini membahas mengenai budaya nusantara. Pengenalan tari Adat dari setiap daerah, pakaian adat yang dikenakan oleh masyarakat daerah dan jenis rumah adat yang ada di Indonesia, kemudian game edukasi ini bergenre Game tebak gambar. Game dimainkan oleh satu user. Pada penelitian ini, target pasarnya jelas yaitu siswa dan Masyarakat umum. Kemudian genre yang diambil juga tetap menarik bagi siswa ataupun khalayak umum untuk dimainkan. Aspek edukasi juga sudah terlihat jelas.

Adapun penelitian lain, yang dilakukan oleh (Khaerudin et al., 2021). Penelitian tersebut berjudul “Game Edukasi Dengan Menggunakan Unity 3d Untuk Menunjang Proses Pembelajaran”. Pada penelitian ini, penulis menggunakan Unity 3D untuk mengembangkan game edukasi dengan genre RPG, dimana player disajikan materi kompetensi dasar pada setiap level. Siswa akan meng-explore dungeon, dan dapat mengakses materi dengan cara menyelesaikan objective/mission, membunuh musuh, atau menjangkau area checkpoint. Setiap sesi level terdapat Boss yang harus dikalahkan dengan cara menjawab pertanyaan dengan benar tentang materi yang telah diberikan untuk naik level berikutnya. Metode penelitian yang digunakan dalam penelitian ini adalah menggunakan metode Penelitian dan Pengembangan (P&P). Namun pada penelitian ini kompetensi yang digunakan tidak spesifik, sehingga target pasar tidak terlalu jelas.

Untuk penelitian lainnya, dilakukan oleh (Silvana & Anistyasari, 2021), dengan judul penelitian tersebut adalah “Pengembangan Game “Script Labyrinth” Untuk Meningkatkan Computational Thinking Siswa Dalam Pelajaran Pemrograman Web Dan Perangkat Bergerak Di SMKN 2 Surabaya”. Pada penelitian ini juga menggunakan Unity 3D untuk mengembangkan game edukasi dengan genre maze, dimana player harus

mencari garis finish dari labirin dan player harus berhadapan dengan bos level dengan cara menjawab kuis dengan tepat terkait dengan pemrograman web. Berbeda dari penelitian sebelumnya, penelitian ini sudah membuat game, dimana game tersebut dapat digunakan oleh target pasar yang sesuai yaitu anak SMK dan kompetensi atau materi yang spesifik yaitu pemrograman web.

Terdapat juga penelitian dari (Nirwana & Purwanto, 2022) dengan judul “Pengembangan Teknologi Game Indonesia “Pramuka Asik” Menggunakan Unity 2D Engine Berbasis Android”. Dimana, game berikut dibuat menggunakan Unity 2D dan tujuan dibuatnya aplikasi ini yaitu untuk memperkenalkan kegiatan pramuka kepada peserta didik atau anggota pramuka dalam bentuk aplikasi Game di android dan mengasah kemampuan peserta didik atau anggota pramuka dalam mengenal pramuka. Dimana, pada penelitian ini sudah melakukan testing kepada target pasar dan memiliki respon yang baik.

Pada penelitian yang diusulkan, yaitu “Pembuatan Game Coding Puzzle 2D Untuk Mahasiswa Politeknik Caltex Riau (PCR)” akan dibuat sebuah game edukasi bermodel puzzle berbasis mobile menggunakan metode GDLC yang dibuat dengan Unity.

Tabel 2. 1 Penelitian Terdahulu

No	Penelitian	Judul Penelitian	Genre
1.	(Rohmawati & Menarianti, 2019)	Pengembangan Game Edukasi Tentang Budaya Nusantara “Tanara” Menggunakan Unity 3d Berbasis Android	Tebak Gambar
2.	(Khaerudin et al., 2021)	Game Edukasi Dengan Menggunakan Unity 3d Untuk Menunjang Proses Pembelajaran	RPG
3.	(Silvana & Anistyasari, 2021)	Pengembangan Game “Script Labyrinth” Untuk Meningkatkan Computational Thinking Siswa Dalam Pelajaran	Maze

		Pemrograman Web Dan Perangkat Bergerak Di SMKN 2 Surabaya	
4.	(Nirwana & Purwanto, 2022)	Pengembangan Teknologi Game Indonesia “Pramuka Asik” Menggunakan Unity 2D Engine Berbasis Android	Edukatif
5.	Penelitian yang diusulkan	Pembuatan Game Coding Puzzle 2D Untuk Mahasiswa Politeknik Caltex Riau (PCR)	Puzzle, Edukasi

2.2 Landasan Teori

2.2.1 Game Edukasi

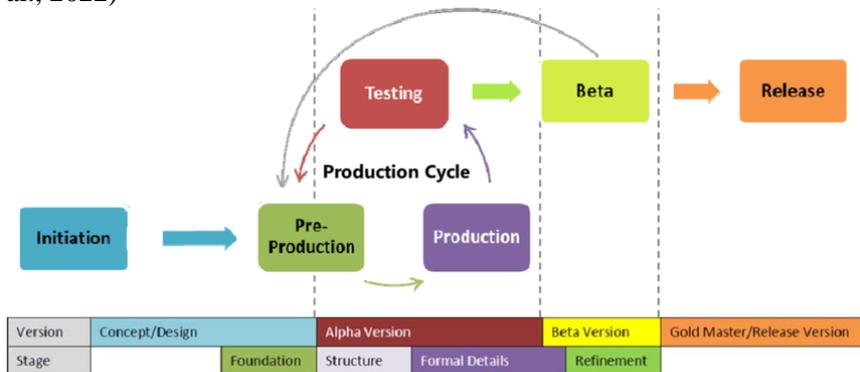
Game dalam kamus bahasa Indonesia memiliki arti permainan. Sedangkan menurut Wahono (dalam Agustina, R. Dan Chandra, A. 2017:25) game adalah suatu aktifitas baik itu terstruktur maupun semi terstruktur yang bertujuan sebagai sarana hiburan dan kadang untuk pendidikan. Pengertian edukasi menurut kamus besar bahasa Inggris disebut education yang artinya pendidikan. Sedangkan menurut Nugroho D, A. dkk. (2017:70) menyatakan bahwa edukasi adalah suatu proses belajar yang memiliki tujuan sebagai pengembang potensi diri pada murid dan proses belajar yang baik.

Game edukasi menurut Henry (dalam Hamka, W, A. dan Gani, A. 2016:81) menyatakan bahwa game yang berisi konten pendidikan dan memiliki tujuan sebagai pemancing minat belajar anak dalam menyerap materi pembelajaran sambil bermain, diharapkan dengan adanya game ini anak jadi lebih mudah memahami materi yang disampaikan.

Berdasarkan pengertian diatas dapat diartikan bahwa game edukasi adalah suatu permainan yang digunakan sebagai sarana hiburan bagi anak dan di dalamnya berisi konten pendidikan agar memudahkan tenaga pendidik dalam menyampaikan materi pelajaran supaya anak menjadi lebih bertanggung jawab, cerdas dan terampil (Pradana & Nita, 2019).

2.2.2 Game Development Life Cycle (GDLC)

Game Development Life Cycle (GDLC) yaitu sebuah pengembangan dari sebuah game yang lebih mengutamakan aspek interaktif yang memiliki enam fase pengembangan, dimulai dari fase inisialisasi/pembuatan konsep, preproduction, production, testing, beta dan release. Dari 6 fase tersebut dapat di kelompokkan menjadi 3 proses utama yaitu proses Inisialisasi, Produksi, dan Release (Agung Saputra et al., 2022)



Gambar 2. 1 Fase dan Proses GDLC (Sumber : www.semanticscholar.org)

Gambar 2.1 adalah beberapa proses yang dilakukan sesuai model pengembangan Game Development Life Cycle yang meliputi:

- 1) Initiation, meliputi konsep yang dibuat dalam game edukasi permainan susun puzzle dan disertai oleh pertanyaan singkat, dan dilengkapi dengan menu ensiklopedia untuk media pembelajaran tentang kebudayaan nusantara.
- 2) Pre-Production, melibatkan produksi prototipe game serta konsepsi dan desain dasar game. Data dikumpulkan selama tahap Pra-Produksi dengan mencari referensi game yang akan dirancang.
- 3) Production, merupakan proses inti setelah melakukan tahap Pra Produksi, proses pembuatan aset dan source code. Pada tahap produksi memiliki fokus untuk mengumpulkan data, merancang desain permainan, pembuatan aset, programming dan pembuatan source code.

- 4) Testing, dilakukan oleh peneliti untuk menguji fungsi dari aplikasi game susun puzzle yang sudah dibuat, tahap ini menggunakan metode uji black box testing untuk mengetes fungsi game berjalan secara optimal [8].
- 5) Beta, dilakukan setelah pengujian awal selesai, Beta meliputi pengujian aplikasi ke pengguna secara langsung melalui beberapa tester bila dirasa aplikasi masih memiliki kekurangan maka akan lakukan pemrosesan kembali ke tahap Pre-Production.
- 6) Release, bila aplikasi sudah di rasa memenuhi syarat dan layak guna, pada tahapan ini game puzzle akan di rilis ke pengguna

2.2.3 Android

Android adalah sebuah sistem operasi (OS) yang dikembangkan oleh Google untuk perangkat mobile, seperti smartphone, tablet, dan smartwatch. Android didasarkan pada kernel Linux dan dirancang khusus untuk mendukung berbagai aplikasi dan layanan yang berjalan pada perangkat bergerak. Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android umum digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple, dan BlackBerry OS (Ramadhan, 2023).

2.2.4 Algoritma dan Pemrograman

Algoritma (Sahid, 2020) adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Langkah-langkah dalam Algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar.

Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam terminologi algoritma. Dalam kehidupan sehari-hari pun banyak terdapat proses yang dinyatakan dalam suatu algoritma. Jadi dalam menjalankan algoritma harus: Mengerti setiap langkah dalam Algoritma, Mengerjakan operasi yang bersesuaian dengan langkah tersebut.

Agar dapat dilaksanakan oleh komputer, algoritma harus ditulis

dalam notasi bahasa pemrograman disebut dengan program. Jadi program adalah perwujudan atau implementasi teknis Algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer.

2.2.5 Game Puzzle

Puzzle berasal dari Bahasa Inggris yang artinya tebakkan. Bermain tebak-tebakan atau memecahkan suatu masalah dalam sebuah permainan adalah hal yang menghibur. Hal ini biasanya juga terdapat dalam sebuah pelajaran, contohnya matematika. Dengan bermain puzzle anak akan lebih tertarik dan memahami materi yang sedang mereka pelajari. Selain itu, bermain puzzle juga dapat meningkatkan kemampuan kognitif anak dan mengaktifkan saraf-saraf berpikir kreatif (GuruBelajar.ID, 2022).

Game merupakan bentuk permainan yang bisa menambah kreatifitas dan daya ingat siswa lebih mendalam. Munculnya motivasi untuk memecahkan masalah dalam puzzle menjadi faktor utama kemampuan tersebut terlatih, namun tetap menyenangkan. Karena, siswa akan ketagihan dan selalu tertantang untuk mencoba dan terus mencoba hingga berhasil. Oleh karena itu, guru bisa menggunakan puzzle sebagai salah satu media pembelajaran yang asyik dan menyenangkan.

Dengan bermain, siswa akan berkesempatan untuk berpikir dan lebih imajinatif. Hal tersebut erat kaitannya dengan perkembangan kreatifitas siswa, sehingga akan membuat kemampuan kognitif juga akan terasah. Berikut penjelasan lebih lanjut mengenai manfaat dari game puzzle untuk siswa:

- 1) Meningkatkan Kemampuan Kognitif (Cognitive Skill)
Kemampuan kognitif berhubungan dengan pembelajaran yang sifatnya memecahkan masalah. Puzzle merupakan salah satu contoh permainan yang membutuhkan kemampuan berpikir untuk menyelesaikannya. Dengan menyusun gambar menjadi sebuah satu kesatuan yang utuh, maka secara tidak langsung kemampuan kognitif anak akan berkembang.
- 2) Mengembangkan Motorik Halus (Fine Motor Skill)
Kemampuan ini berkaitan dengan kemampuan anak atau siswa untuk menggunakan otot-otot kecilnya, terkhususnya bagian tangan, dan jari-jari tangan. Agar puzzle tersusun dengan baik, maka perlu adanya kecermatan dan kehati-hatian dalam

- meletakkan bagian-bagiannya.
- 3) Belajar Bersosialisasi
Puzzle bisanya dimainkan secara perorangan, namun menyusun puzzle juga bisa secara kelompok. Dalam kelompok, siswa akan bekerjasama, saling menghargai, dan saling berdiskusi untuk menyelesaikan puzzle tersebut.
 - 4) Melatih Kemampuan Logika
Dengan menentukan berbagai gambar yang ada di puzzle, contohnya gambar manusia. Siswa akan menyimpulkan di mana bagian-bagian itu harus diletakkan.
 - 5) Melatih Kesabaran dan Ketekunan

2.2.6 Unity

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan game multi platform yang didesain untuk mudah digunakan. Unity itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada Unity dibuat dengan user interface yang sederhana. Editor ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan ranking teratas untuk editor game.

Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan directX. Unity mendukung semua format file, terutamanya format umum seperti semua format dari art applications. Unity cocok dengan versi 64-bit dan dapat beroperasi pada Mac OS x dan windows dan dapat menghasilkan game untuk Mac, Windows, Wii, iPhone, iPad dan Android (Harlanto, 2020)

2.2.7 User Acceptance Testing (UAT)

User acceptance testing yang juga sering disebut dengan beta testing atau end-user testing merupakan fase pengembangan sistem/perangkat lunak yang diuji secara langsung oleh end-user atau business representative.

User acceptance testing tidak melakukan pengujian yang berfokus pada fungsi dan menu yang dibangun oleh developer, melainkan berfokus pada sisi tim bisnis untuk memverifikasi bahwa sistem telah memenuhi seluruh business requirement dengan skenario dan data yang mewakili penggunaan aktual di dunia nyata–setelah dirilis (Sarah Juliandiny,

2023).

2.2.8 Validasi Konten

Uji validitas adalah pengujian kelayakan. Suatu instrumen penelitian yang telah dinyatakan valid memiliki makna instrument tersebut dapat mengukur variable yang akan diukur (Sugiyono, 2014). Pengujian validitas berhubungan dengan intepretasi skor dari instrument penelitian (Cook & Beckman, 2006). Pengujian validitas instrumen penelitian dapat dilakukan berdasarkan isi (content), kriteria, dan konsep (construct). Validitas isi (content) adalah pengujian kelayakan instrument penellitian oleh ahli. Uji validasi isi dapat dilakukan dengan pendekatan kualitatif, kuantitatif, atau penggabunagn kualitatif dan kuantitatif (Dwi Puspitasari & Febrinita, 2021).

2.2.9 Pre-test dan Post-test

Pre-test atau tes awal merupakan tes yang dilaksanakan dengan tujuan untuk mengetahui sejauh mana materi atau bahan pelajaran yang akan diajarkan telah dapat dikuasai oleh siswa. Sedangkan post test atau tes akhir adalah tes yang dilaksanakan dengan tujuan untuk mengetahui apakah semua materi yang tergolong penting sudah dapat dikuasai dengan sebaik-baiknya oleh siswa.

Post-test menurut kamus istilah pendidikan dan umum yang diuraikan oleh Syamsiah dalam skripsinya, menjelaskan bahwa post test adalah tes yang diselenggarakan setelah selesai pelaksanaan kegiatan belajar mengajar, guna mengetahui sampai di mana anak-anak dapat menguasai bahan pelajaran yang sudah diajarkan. Dapat kita simpulkan bahwa post test adalah tes akhir pada setiap program belajar (Wilman Juniardi, 2022).

BAB III PERANCANGAN

3.1 Initiation

3.1.1 Ide dan Konsep Game

Pada bagian ide dan konsep game adalah tahapan dimana untuk Merencanakan seperti apa game yang akan dibangun nantinya. Adapun beberapa hal yang perlu ditentukan dalam pembuatan ide dan konsep adalah:

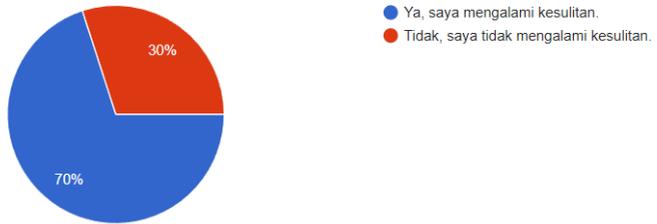
- a) Judul Game
Code Arcade
- b) Genre Game
Puzzle, Edukasi
- c) Interaksi
Antara komputer dan manusia
- d) Target Audience
Mahasiswa PCR prodi Teknik Informatika
- e) Media/Platform
Android
- f) Identifikasi Masalah

Untuk identifikasi masalah dapat dilihat pada gambar dibawah berikut:



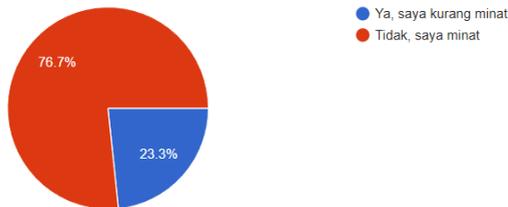
Gambar 3. 1 Grafik Mahasiswa Dengan Dasar Pemrograman

68.3 % mahasiswa tidak memiliki dasar pemrograman yang mempengaruhi kemampuan mahasiswa dalam mengikuti proses belajar pemrograman diperkuliahan.



Gambar 3. 2 Grafik Mahasiswa Kesulitan Belajar

Dan 70% mahasiswa mengalami kesulitan dalam proses belajar di perkuliahan. Meski mengalami kesulitan, beberapa mahasiswa masih memiliki minat dalam mengikuti pembelajaran khususnya di perkuliahan.



Gambar 3. 3 Grafik Minat Mahasiswa Mengikuti Pelajaran

Dibuktikan dengan 76.7% mahasiswa masih berminat dalam mengikuti proses belajar pemrograman selama kuliah. Dengan adanya game ini, diharapkan dapat memudahkan mahasiswa dalam mempelajari pemrograman.

3.1.2 Storytelling

Pada saat game Code Arcade dijalankan, game akan menampilkan halaman awal dimana akan ada menu mulai, tentang dan keluar game. Ketika pemain pertama kali memainkan game, maka akan diarahkan ke halaman menentukan kesulitan. Jika pemain memilih mode selain easy (medium atau hard), maka pemain akan ditunjukkan untuk melakukan pre-test. Untuk pemain yang memilih mode easy maka akan langsung menuju halaman module yang berisi beberapa section dengan materi yang dapat dipelajari, sedangkan untuk pemain yang melakukan pre-test maka akan bisa mengakses section dari mode easy sampai section mode berdasarkan kesulitan yang dipilih. Section pada game ini terkunci berdasarkan mode yang dipilih pemain. Pemain dengan mode easy hanya

bisa mengakses beberapa materi dasar dan baru bisa mengakses materi yang lebih sulit ketika sudah mencapai mode medium ataupun hard.

Setelah bisa mengakses section, maka pemain bisa membaca materi terlebih dahulu sebelum memainkan puzzle yang disediakan. Puzzle yang disediakan adalah puzzle logika berupa melengkapi potongan kode, memperbaiki bug, menebak output dari code dan menyusun potongan kode. Puzzle ini berlaku untuk semua section dimana kontennya akan menyesuaikan materi yang sedang diakses pemain. Puzzle disini tidak ada batasan waktu untuk dikerjakan.

Jika pemain memilih tentang game, maka pemain akan diarahkan ke halaman dimana terdapat informasi terkait developer yang mengembangkan game, seperti kontak. Dan jika pemain memilih keluar game, maka game akan berhenti berjalan.

3.2 Pre-Production

3.2.1 Kebutuhan Pengguna

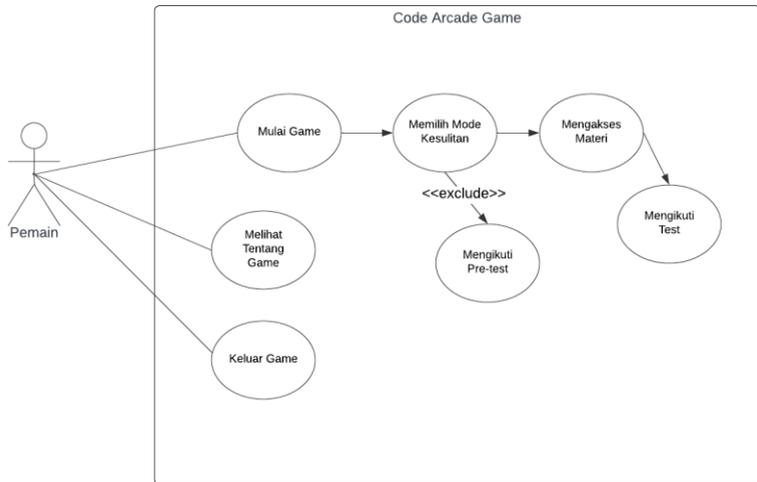
Kebutuhan pengguna untuk game coding puzzle ini dilakukan dengan melakukan wawancara kepada Bapak Erzi Hidayat, S.T., M.Kom. selaku dosen prodi Teknik Informatika yang mengajar mahasiswa PCR pada matakuliah Algoritma dan Pemrograman. Berdasarkan wawancara yang dilakukan telah didapatkan beberapa kebutuhan pada perancangan dan pengembangan game coding puzzle, sebagai berikut:

Tabel 3. 1 Tabel Kebutuhan Pengguna

No	Kebutuhan
1	Desain interface game yang menarik seperti penggunaan warna dan desain objek
2	Adanya tutorial dan penjelasan/narasi diawal sebagai pengenalan sebelum memainkan puzzle
3	Materi puzzle berupa logika dasar dan materi dasar Java

Pada gambar 3.5 dijelaskan terkait alur game coding puzzle. Ketika pengguna memulai game, maka akan diberikan pilihan antara memulai game, melihat tentang game atau keluar dari game. Jika memilih memulai game maka pemain akan dicek apakah baru pertama kali main atau sudah pernah main, jika belum pernah main maka pemain akan diarahkan untuk memilih mode kesulitan permainan. Kemudian jika pemain memilih mode kesulitan selain easy maka akan ditunjukkan ke halaman pre-test. Namun apabila pemain memilih easy, maka pemain diarahkan langsung ke halaman modul, sama seperti ketika pemain sudah pernah main. Apabila sudah mengakses halaman modul, maka pemain bisa mengakses section yang berisi materi dan test. Ketika ingin bermain ulang maka pilih tombol main ulang dan akan diarahkan kembali ke halaman game. Apabila pemain tidak ingin bermain ulang, maka akan dicek apakah materi yang diakses adalah materi terakhir dari mode, jika iya maka akan dicek mode terakhir pemain antara easy atau medium. Jika easy maka pemain akan masuk ke mode medium dan dapat mengakses materi easy-medium, dan jika medium maka masuk ke mode hard dan dapat mengakses materi easy-hard dan apabila sudah mode hard maka pemain tidak akan dicek dan dapat mengakses semua materi. Jika memilih tentang kami, maka pemain akan diarahkan ke halaman yang berisi informasi tentang game. Dan jika pemain tidak memilih keduanya maka pemain dianggap mau keluar dari game dan game berhenti berjalan.

3.2.4 Use Case Game Coding Puzzle



Gambar 3. 6 Use Case Game Coding Puzzle

3.2.4.1 Usecase Description

Tabel 3. 2 Use Case Mulai Game

Nama Usecase	Mulai Game
Actor	Pemain
Pre-condition	Pemain menekan tombol Mulai Game pada menu utama
Main	Dapat memilih mode permainan
Post-condition	Menampilkan halaman mode permainan
Alternative Ways	-

Tabel 3. 3 Use Case Memilih Mode Kesulitan

Nama Usecase	Memilih Mode Kesulitan
Actor	Pemain
Pre-condition	Pemain memilih tombol mode kesulitan yang tersedia
Main	Dapat mengakses halaman modul sesuai mode yang dipilih

Post-condition	Menampilkan halaman modul sesuai mode
Alternative Ways	-

Tabel 3. 4 Use Case Mengikuti Pre-test

Nama Usecase	Mengikuti Pre-test
Actor	Pemain
Pre-condition	Menampilkan soal puzzle untuk mengetes kemampuan pemain
Main	Dapat bermain puzzle berdasarkan mode yang dipilih
Post-condition	Menampilkan halaman modul sesuai hasil test berdasarkan mode
Alternative Ways	-

Tabel 3. 5 Use Case Mengakses Materi

Nama Usecase	Mengakses Materi
Actor	Pemain
Pre-condition	Menampilkan materi yang akan diujikan
Main	Dapat mengakses materi yang akan diujikan
Post-condition	Menampilkan tombol test untuk menguji kemampuan
Alternative Ways	-

Tabel 3. 6 Use Case Mengikuti Test

Nama Usecase	Mengikuti Test
Actor	Pemain
Pre-condition	Menampilkan soal puzzle untuk mengetes kemampuan pemain
Main	Dapat bermain puzzle berdasarkan materi yang dipilih
Post-condition	Menampilkan hasil test pada halaman hasil
Alternative Ways	-

Tabel 3. 7 Use Case Melihat Tentang Game

Nama Usecase	Melihat Tentang Game
Actor	Pemain
Pre-condition	Pemain menekan tombol Tentang Game pada menu utama
Main	Dapat melihat informasi tentang game
Post-condition	Menampilkan halaman tentang game
Alternative Ways	-

Tabel 3. 8 Use Case Keluar Game

Nama Usecase	Keluar Game
Actor	Pemain
Pre-condition	Pemain menekan tombol Keluar Game pada menu utama
Main	Game berhenti berjalan
Post-condition	Kembali ke menu utama perangkat
Alternative Ways	-

3.2.4.2 Skenario Mulai Game

Pre-condition: Pemain memilih menu mulai game

Post-condition: Pemain dapat memilih mode permainan

Tabel 3. 9 Skenario Mulai Game

Pemain	Sistem
Memilih menu mulai game	
	Mengarahkan ke halaman mode permainan (logika dasar, flowchart, pseudocode, dan code)
Memilih mode permainan dan menekan <i>button</i> Kembali untuk Kembali ke halaman utama	

3.2.4.3 Skenario Memilih Mode Kesulitan

Pre-condition: Pemain memilih mode kesulitan game

Post-condition: Pemain dapat mengakses halaman modul atau pre-test

Tabel 3. 10 Skenario Memilih Mode Permainan

Pemain	Sistem
Memilih mode kesulitan	
	Mengarahkan ke halaman modul
Skenario Alternatif	
Memilih mode kesulitan	
	Mengarahkan ke halaman pre-test

3.2.4.4 Skenario Mengakses Materi

Pre-condition: Pemain dapat melihat materi

Post-condition: Pemain dapat mengakses test

Tabel 3. 11 Skenario Mengakses Materi

Pemain	Sistem
Membuka halaman materi	
	Menampilkan materi
Menekan tombol test	
	Mengarahkan ke halaman test

3.2.4.5 Skenario Mengikuti Test

Pre-condition: Pemain dapat melihat soal puzzle

Post-condition: Pemain dapat melihat hasil dari test

Tabel 3. 12 Skenario Mengikuti Test

Pemain	Sistem
Membuka halaman test	
	Menampilkan soal puzzle test
Menyelesaikan puzzle	
	Mengarahkan ke halaman hasil

3.2.4.6 Skenario Melihat Tentang Game

Pre-condition: Pemain memilih menu tentang game

Post-condition: Pemain dapat melihat informasi tentang game

Tabel 3. 13 Skenario Melihat Tentang Game

Pemain	Sistem
Memilih tentang	
	Menampilkan informasi pembuat game dan button Kembali
Membaca dan menekan button Kembali untuk Kembali kemenu utama	

3.2.4.7 Skenario Keluar Game

Tabel 3. 14 Skenario Keluar Game

Pemain	Sistem
Pemain memilih keluar	
	Game berhenti berjalan

3.3 Production

3.3.1 Prototype Game Coding Puzzle



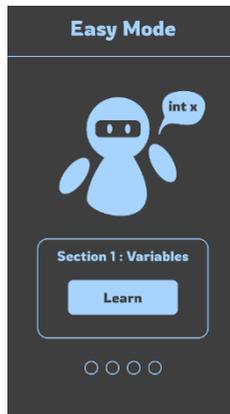
Gambar 3. 7 Desain Antarmuka Awal

Gambar 3.7 merupakan tampilan awal ketika pemain menjalankan game coding puzzle ini. Dimana terdapat menu memulai game, tentang game dan exit.



Gambar 3. 8 Desain antarmuka memilih mode kesulitan

Gambar 3.8 adalah tampilan awal pemain ketika pemain baru saja memainkan game. Berfungsi untuk menentukan kemampuan pemain dengan menyediakan pre-test apabila mengklik pilihan selain mode easy (I'm new).



Gambar 3. 9 Desain antarmuka menu module

Gambar 3.9 merupakan menu awal yang diperoleh apabila pemain memilih mode easy atau pemain sudah menyelesaikan pre-test dengan baik. Dimana terdapat section yang berisi materi dan test untuk pengujian.



Gambar 3. 10 Desain antarmuka materi

Gambar 3.10 adalah halaman materi yang menampilkan semua materi untuk section yang dipilih. Dimana pemain bisa membaca semua materi yang disediakan sebelum memainkan puzzle atau mengikuti test kemampuan.



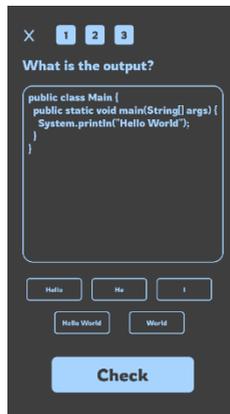
Gambar 3. 11 Desain antarmuka puzzle code completion

Gambar 3.11 adalah salah satu puzzle yang muncul ketika pemain memainkan game/test. Dimana, puzzle ini mengharuskan pemain untuk melengkapi potongan kode dengan benar dari opsi-opsi yang disediakan.



Gambar 3. 12 Desain antarmuka puzzle bug fixing

Gambar 3.12 adalah bagian dari puzzle yang mengharuskan pemain untuk mencari bug atau penyebab error dari suatu code dari opsi yang disediakan. Puzzle ini akan menyesuaikan kontennya berdasarkan materi.



Gambar 3. 13 Desain antarmuka puzzle predict output

Gambar 3.13 adalah puzzle lain yang mengharuskan pemain untuk menebak output dari suatu kode. Pemain harus memilih output yang benar dari pilihan yang disediakan agar puzzle selesai.



Gambar 3. 14 Desain antarmuka puzzle arrange code

Gambar 3.14 adalah puzzle lainnya yang mengharuskan pemain untuk menyusun potongan kode agar program berjalan dengan benar dengan cara memilih potongan kode dari opsi-opsi yang telah disediakan.



Gambar 3. 15 Desain antarmuka halaman result

Gambar 3.15 adalah halaman hasil ketika pemain menyelesaikan semua puzzle pada materi yang dipilih. Dengan adanya timestamp yang mengecek seberapa lama pemain berhasil menyelesaikan test dan akurasi pemain.

3.4 Testing

3.4.1 User Acceptance Testing (UAT)

UAT pada game berfokus pada kebutuhan functional system. Tujuan dari pengujian UAT ini adalah untuk memastikan bahwa fungsi-fungsi pada game sudah berjalan dengan baik dan sesuai dengan output yang diharapkan setiap fungsi yang terdapat pada aplikasi akan dimasukkan kedalam tabel test case kemudian akan dilihat apakah berjalan sesuai dengan scenario game yang dibuat, jika fungsi nya sesuai maka akan dinyatakan “valid” dan yang tidak sesuai akan dinyatakan “tidak valid”. UAT akan diujikan ke dosen prodi Teknik Informatika PCR.

3.4.2 Validasi konten

Pengujian ini akan dilakukan ketika konten atau isi dari game coding puzzle selesai dibuat, kemudian akan divalidasi oleh dosen yang bersangkutan mengenai isi dari game yang telah dibuat apakah telah sesuai dengan standarisasi pembelajaran pemrograman di kampus, pengujian ini bertujuan agar pembelajaran yang diterapkan digame sesuai dengan kurikulum pembelajaran algoritma dan pemrograman dasar mahasiswa PCR.

3.4.3 Pengujian responden (Pre-test dan Post-test)

Pengujian akan dilakukan kepada mahasiswa PCR yang terdiri atas 15 mahasiswa. Pengujian akan dilakukan di PCR dimana nantinya mahasiswa akan diuji secara individu untuk melakukan test yang diberikan yaitu pre-test dan post-test. Untuk tahap pertamanya mahasiswa akan diuji secara konvensional dimana mahasiswa akan diberikan soal pre-test terlebih dahulu, pengujian dinyatakan selesai jika mahasiswa telah berhasil menjawab semua soal pre-test yang diberikan.

Untuk tahap berikutnya siswa akan diminta untuk bermain game coding puzzle, kemudian akan dilihat seberapa banyak level yang dapat dibuka oleh mahasiswa tersebut, kemudian diminta game kembali untuk diambil nilai post-testnya, dan akan dilihat hasil dari pre-test dan post-test apakah mengalami peningkatan setelah bermain game atau tidak.

Penilaian dilihat dari seberapa cepat dan akurat mahasiswa menyelesaikan puzzle.

3.4.4 Kuesioner kepuasan pengguna

Setelah melakukan pre-test dan post-test mahasiswa diminta untuk mengisi kuesioner kepuasan pengguna dimana digunakan untuk memastikan desain dan fungsionalitas game sesuai dengan kebutuhan perancangan.

Tabel 3. 15 Tabel Kuesioner Kepuasan

No	Pertanyaannya	Jumlah jawaban		Persentase	
		Ya	Tidak	Ya	Tidak
1	Game sudah memiliki tampilan yang mudah dipahami				
2	Pembelajaran pemrograman lebih menarik melalui bermain game				
3	Materi mudah dipahami dengan menggunakan game				
4	Apakah kamu lebih suka belajar pemrograman melalui game?				
5	Apakah kamu mau menggunakan game ini untuk belajar pemrograman dan membiasakannya dirumah?				

BAB IV PENGUJIAN DAN ANALISIS

4.1 Production

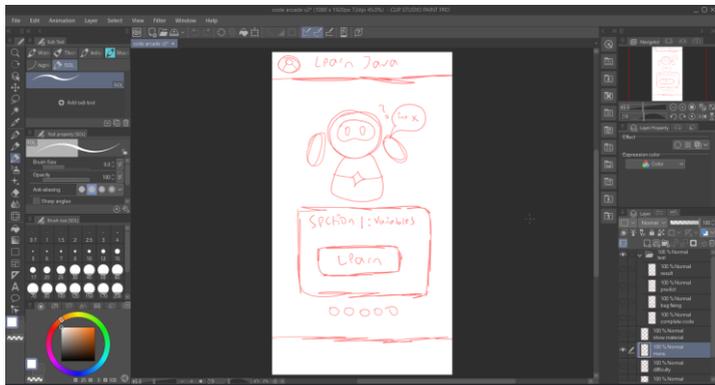
Produksi adalah proses inti yang berputar di sekitar penciptaan aset, pembuatan kode sumber, dan integrasi kedua elemen. Prototipe terkait dalam fase ini adalah perincian dan penyempurnaan formal.

4.1.1 Proses Pembuatan Aset

Terdapat 2 tahapan, yaitu *sketching* dan finalisasi. *Sketching* dilakukan di clip paint studio, dan finalisasi aset dilakukan di figma.

1) *Sketching*

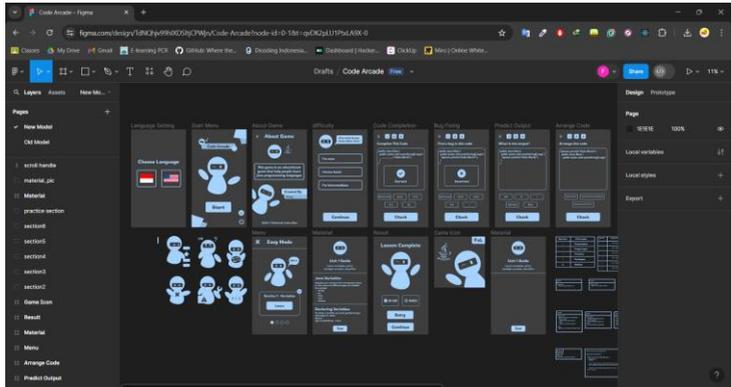
Pada proses ini, dilakukan penggambaran sketsa aset untuk tiap resources.



Gambar 4. 1 Sketching

2) *Finalisasi*

Pada proses ini, setiap hasil sketching akan di finalisasi kan menjadi aset untuk digunakan di game



Gambar 4. 2 Finalisasi

4.1.2 Desain Antarmuka Game

Pada tampilan awal game, akan muncul opsi memilih bahasa yang terdiri dari bahasa indonesia dan bahasa inggris jika baru pertama kali bermain game.



Gambar 4. 3 Tampilan Memilih Bahasa

Setelahnya akan masuk ke halaman menu awal yang berisi opsi mulai, tentang game dan exit.



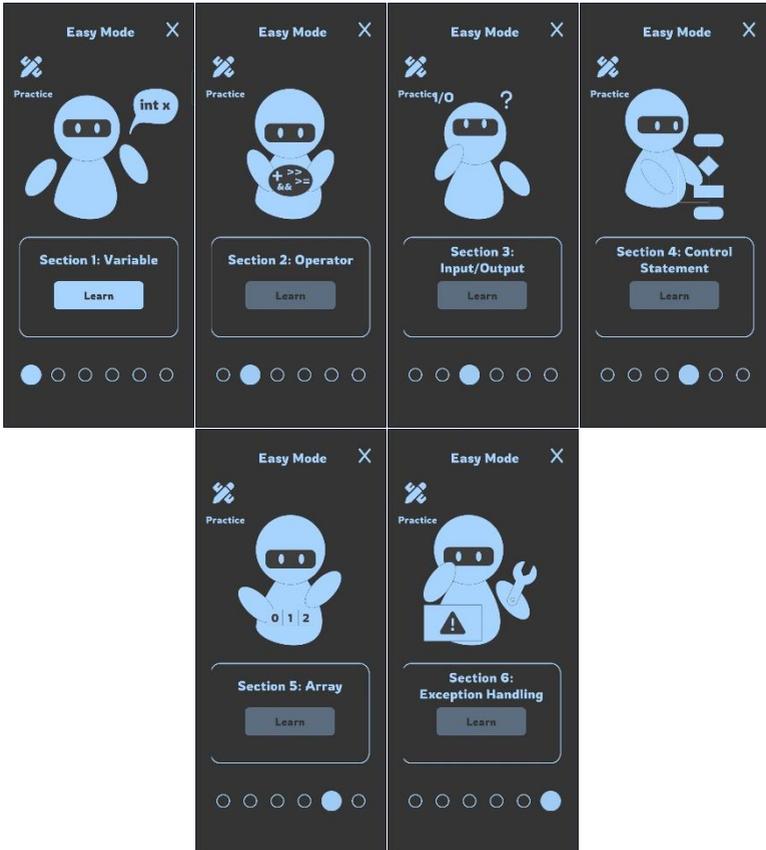
Gambar 4. 4 Tampilan menu awal

Kemudian, jika baru pertama kali bermain, akan muncul opsi memilih kesulitan untuk menentukan kemampuan pemain yang akan mengarahkan pemain mengikuti pre-test jika pemain memilih mode selain mode easy (I'm new).

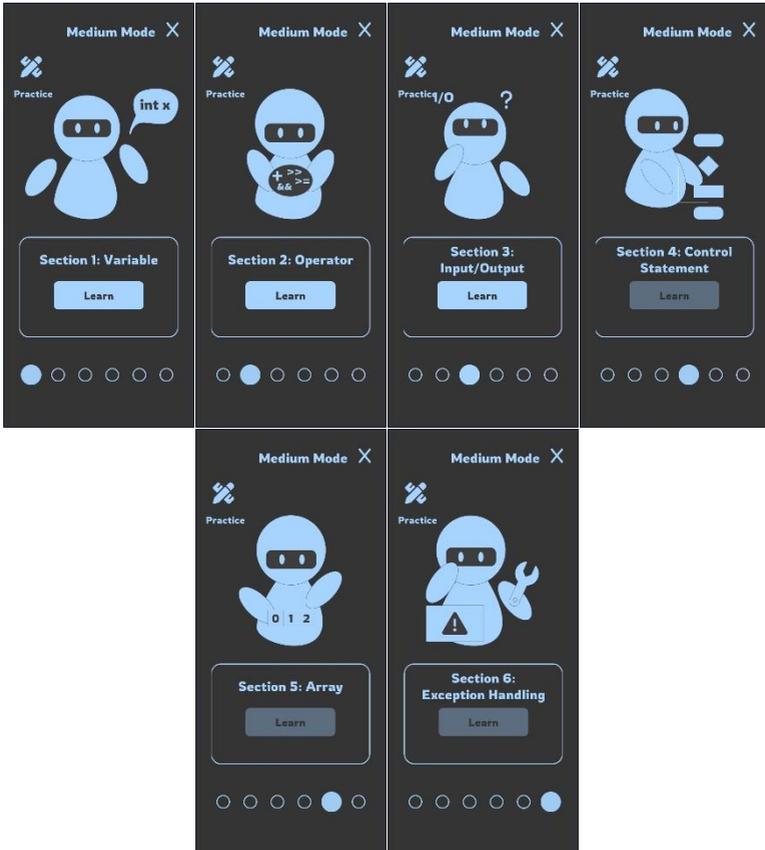


Gambar 4. 5 Tampilan memilih mode kesulitan

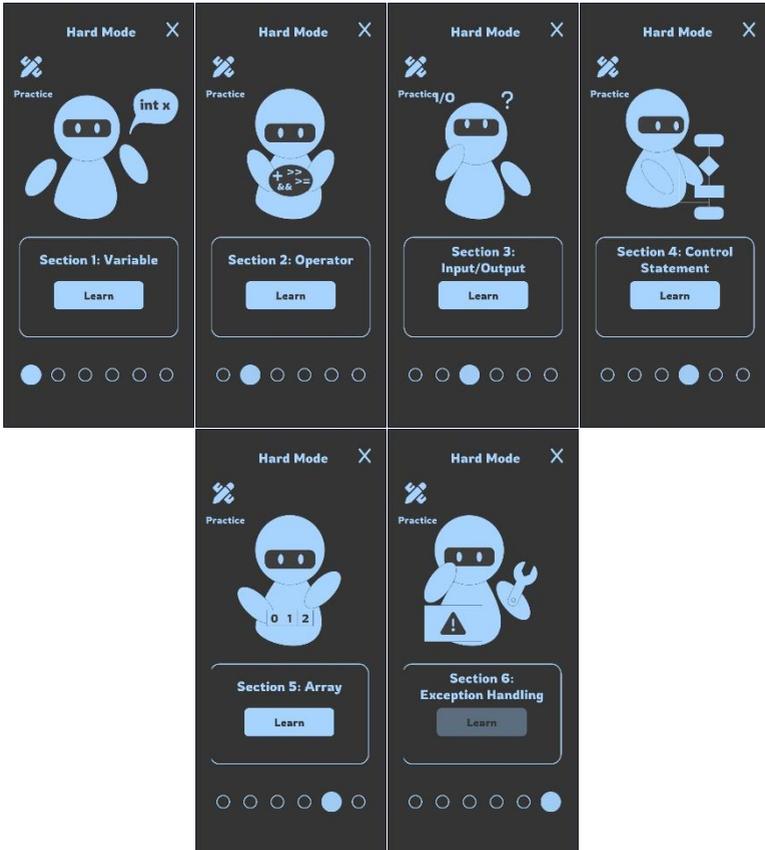
Apabila sudah memilih mode, maka pemain akan diarahkan ke halaman menu module, dimana pemain dapat memilih belajar atau latihan.



Gambar 4. 6 Tampilan menu module easy

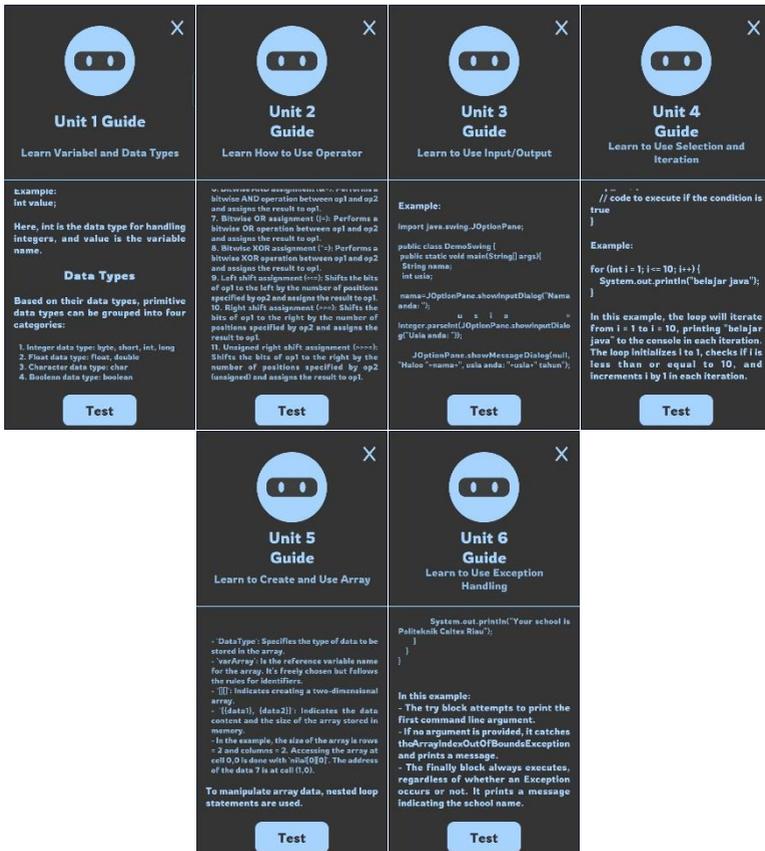


Gambar 4. 7 Tampilan menu module medium



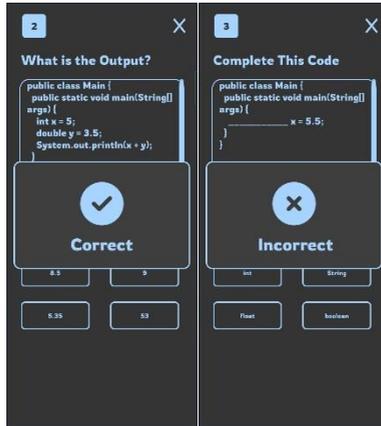
Gambar 4. 8 Tampilan menu module hard

Jika memilih belajar maka akan muncul materi pembahasan sesuai materi yang dipilih dan ada button test untuk menguji pengetahuan materi tersebut.



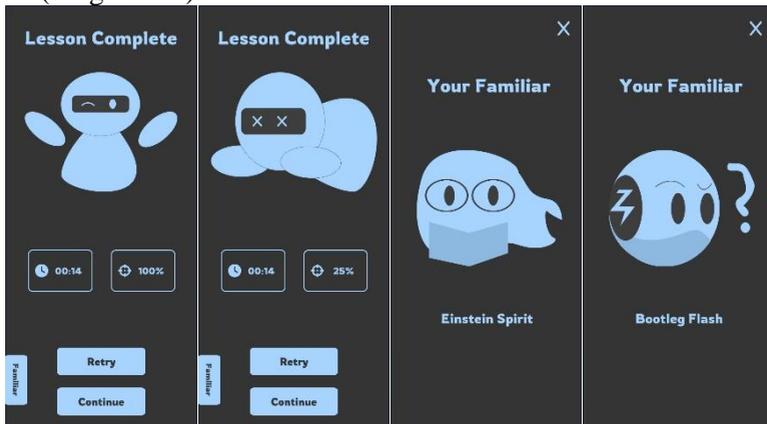
Gambar 4. 9 Tampilan halaman materi

Berikut adalah tampilan test yang digunakan untuk test materi, practice maupun pre-test. Yang membedakannya hanya soal dan jawaban tergantung dengan pilihan user (apakah pre-test, test materi, practice). Untuk setiap puzzle berdasarkan materi ada di lampiran B.



Gambar 4. 10 Tampilan halaman test

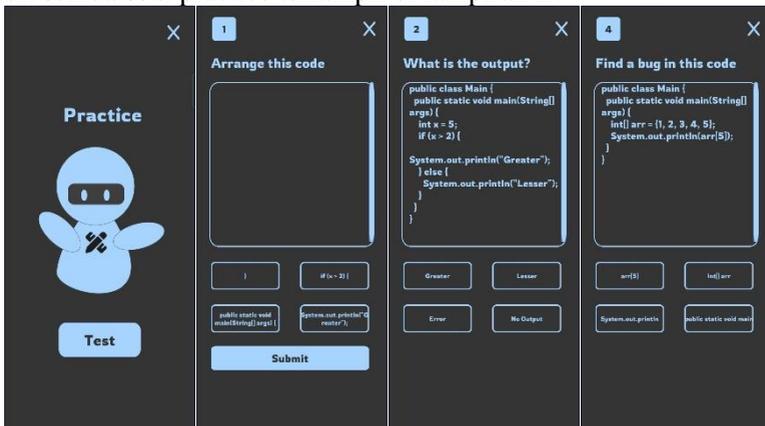
Dan apabila sudah selesai tes, maka pemain dapat melihat hasil yang mencakup timestamp untuk melihat kecepatan pemain dan akurasi. Selain itu, pemain juga dapat melihat familiar/khodam mereka berdasarkan kecepatan dan akurasi mereka. Terdapat 4 jenis familiar/khodam yaitu, cepat dan akurat (einstein spirit), cepat tapi tidak akurat (bootleg flash), lambat tapi akurat (L ninja), dan lambat dan tidak akurat (sadge turtle).

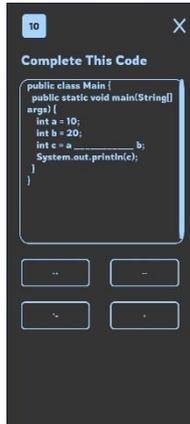




Gambar 4. 11 Tampilan halaman result

Jika memilih practice maka pemain akan diarahkan untuk mengikuti test dengan soalnya terdiri dari semua materi yang ada di game. Untuk semua soal practice terlampir di lampiran B.





Gambar 4. 12 Tampilan Halaman Practice

Selanjutnya jika ingin mengetahui tentang game lebih lanjut dapat melihat halaman tentang yang berisi deskripsi singkat game dan *creator* game.



Gambar 4. 13 Tampilan tentang game

4.1.3 Script Game Code Arcade

Logika yang di pakai dalam pembuatan Game menggunakan Bahasa C#, dengan coding sebagai berikut:

a. Memilih Bahasa

Pemilihan bahasa ditentukan oleh localeSelector, dimana ID 1 adalah bahasa indonesia dan ID 0 adalah bahasa inggris. Setelah memilih bahasa maka akan diarahkan ke halaman startMenu menggunakan function GoToMain()

```
5 public class ChooseLanguage : MonoBehaviour
6 {
7     @ Unity Message | 0 references
8     void Awake()
9     {
10        // Check if AlreadyPlay PlayerPrefs key exists
11        if (!PlayerPrefs.HasKey("AlreadyPlay"))
12        {
13            // Set AlreadyPlay to 0 if it doesn't exist
14            PlayerPrefs.SetInt("AlreadyPlay", 0);
15        }
16
17        // If player has already played, set the build index to the "StartMenu" scene
18        if (PlayerPrefs.GetInt("AlreadyPlay") == 1)
19        {
20            SceneManager.LoadScene("StartMenu");
21        }
22
23        // Start is called before the first frame update
24        @ Unity Message | 0 references
25        void Start()
26        {
27            // If player is playing for the first time, log it
28            if (PlayerPrefs.GetInt("AlreadyPlay") == 0)
29            {
30                Debug.Log("First Time Opening");
31            }
32        }
33
34        // Method to be called when player wants to proceed to "StartMenu" scene
35        0 references
36        public void GoToMain()
37        {
38            SceneManager.LoadScene("StartMenu");
39        }
40    }
41 }
```

Gambar 4. 14 Script perpindahan scene di halaman pemilihan bahasa

```
1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.Localization.Settings;
4
5 @ Unity Script | 1 asset reference | 1 reference
6 public class LocaleSelector : MonoBehaviour
7 {
8     private static LocaleSelector instance;
9
10    @ Unity Message | 2 references
11    private void Start()
12    {
13        if (instance == null)
14        {
15            instance = this;
16            DontDestroyOnLoad(gameObject); // Ensure this GameObject persists across scene changes
17            int savedLocaleID = PlayerPrefs.GetInt("LocaleKey", 0);
18            ChangeLocale(savedLocaleID);
19        }
20        else
21        {
22            Destroy(gameObject); // If another instance exists, destroy this one
23            return;
24        }
25    }
26
27    1 reference
28    public void ChangeLocale(int localeID)
29    {
30        LocalizationSettings.SelectedLocale = LocalizationSettings.AvailableLocales.Locales[localeID];
31        PlayerPrefs.SetInt("LocaleKey", localeID);
32        PlayerPrefs.Save(); // Save PlayerPrefs immediately after changing the locale
33    }
34 }
```

Gambar 4. 15 Script Locale Selector

b. Bermain game

Pada script ini, ditentukan bahasa menggunakan key dari halaman memilih bahasa lalu generate text sesuai dengan bahasa dan akan mengarahkan pemain ke menu selanjutnya tergantung apakah baru pertama kali bermain atau tidak.

```
6 public class PlayGame : MonoBehaviour
7 {
8     @ Unity Message 0 references
9     private void Start()
10    {
11        int localeID = PlayerPrefs.GetInt("LocaleKey");
12        Debug.Log("Locale ID: " + localeID);
13        LocalizationSettings.SelectedLocale = LocalizationSettings.AvailableLocales.Locales[localeID];
14        PlayerPrefs.SetInt("LocaleKey", localeID);
15        PlayerPrefs.Save(); // Save PlayerPrefs immediately after changing the locale
16    }
17
18    0 references
19    public void StartBtn()
20    {
21        if (PlayerPrefs.GetInt("AlreadyPlay") == 1)
22        {
23            Debug.Log("NOT First Time Opening");
24            SceneManager.LoadScene("MainMenu");
25        }
26        else
27        {
28            Debug.Log("First Time Opening");
29            SceneManager.LoadScene("ChooseDifficulty");
30        }
31    }
32
33    0 references
34    public void AboutGame()
35    {
36        SceneManager.LoadScene("AboutGame");
37    }
38
39    0 references
40    public void ExitGame()
41    {
42        Application.Quit();
43    }
44 }
```

Gambar 4. 16 Script Bermain Game

c. Memilih kesulitan

Script ini mengatur mode easy dan medium/hard tergantung pilihan pemain. Ketika memilih mode easy maka script akan otomatis membuka section pertama, membuat player sudah pernah bermain, dan menentukan mode game. Untuk mode medium/hard maka pemain akan diarahkan ke halaman pre-test

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 @ Unity Script (1 asset reference) 0 references
7 public class ChooseDiff : MonoBehaviour
8 {
9     0 references
10    public void PlayEasy()
11    {
12        PlayerPrefs.SetInt("AlreadyPlay", 1);
13        PlayerPrefs.SetInt("GameMode", 1);
14        PlayerPrefs.SetInt("UnlockedSections", 1);
15        SceneManager.LoadScene("MainMenu");
16    }
17
18    0 references
19    public void PlayMediumOrHard()
20    {
21        SceneManager.LoadScene("Test");
22    }
23 }
```

Gambar 4. 17 Script Memilih Kesulitan

d. Game Manager

Game manager berfungsi sebagai core aplikasi untuk menu module, seperti menampilkan UI, mengatur perpindahan scene, dll.

```
10 public class GameManager : MonoBehaviour
11 {
12     public GameObject materialPlace;
13     public GameObject material;
14
15     public TMP_Text diffTitle;
16
17     @ Unity Message 0 references
18     void Start()
19     {
20         MaterialInfo materialInfo = material.GetComponent<MaterialInfo>();
21
22         if (materialInfo == null)
23         {
24             Debug.LogError("MaterialInfo component not found on the material prefab.");
25             return;
26         }
27
28         int numSections = materialInfo.sectionTitle.Length;
29
30         for (int i = 0; i < numSections; i++)
31         {
32             GameObject duplicatedMaterial = Instantiate(material, materialPlace.transform);
33             TMP_Text sectionTitle = duplicatedMaterial.GetComponentInChildren<TMP_Text>();
34
35             Locale selectedLocale = LocalizationSettings.SelectedLocale;
36             StringTable table = LocalizationSettings.StringDatabase.GetTable("Content");
37             string translation = table.GetEntry("SectionKey").GetLocalizedString(selectedLocale);
38             sectionTitle.text = translation + " " + (i + 1) + " " + materialInfo.sectionTitle[i];
39
40             Image sectionImage = null;
41             Image checkImage = null;
42
43             // Find images by their tags
44             Image[] images = duplicatedMaterial.GetComponentsInChildren<Image>();
45             foreach (Image image in images)
46             {
47                 if (image.CompareTag("Section"))
48                 {
49                     sectionImage = image;
50                 }
51                 else if (image.CompareTag("Finished"))
52                 {
53                     checkImage = image;
54                 }
55             }
56
57             Button buttonLearn = duplicatedMaterial.GetComponentInChildren<Button>();
58
59             // Check if both images were found
60             if (sectionImage != null && checkImage != null)
61             {
62                 // Modify sectionImage
63                 sectionImage.sprite = materialInfo.image[i];
64
65                 // Add listener to button
66                 int sectionIndex = i; // Capture current index for the listener
67                 buttonLearn.onClick.AddListener(() => LearnMaterial(sectionIndex));
68
69                 string sectionKey = "sectionCompleted_" + sectionIndex;
70                 // Modify checkImage alpha
71                 if (PlayerPrefs.GetInt(sectionKey) == 0)
72                 {
73                     Color imageColor = checkImage.color;
74                     imageColor.a = 0;
75                     checkImage.color = imageColor;
76                 }
77             }
78
79             int gameMode = PlayerPrefs.GetInt("GameMode");
80             if (gameMode == 3) // Hard Mode
81             {
82                 diffTitle.text = "Hard Mode";
83                 if (i < PlayerPrefs.GetInt("UnlockedSections"))
84                 {
85                     buttonLearn.interactable = true; // Unlocked
86                 }
87                 else
88                 {
89                     buttonLearn.interactable = false; // Locked
90                 }
91             }
92         }
93     }
94 }
```

```

92
93     else if (gameMode == 2) // Medium Mode
94     {
95         diffTitle.text = "Medium Mode";
96         if (i < PlayerPrefs.GetInt("UnlockedSections"))
97         {
98             buttonLearn.interactable = true; // Unlocked
99         }
100        else
101        {
102            buttonLearn.interactable = false; // Locked
103        }
104    }
105    else if (gameMode == 1) // Easy Mode
106    {
107        diffTitle.text = "Easy Mode";
108        if (i < PlayerPrefs.GetInt("UnlockedSections"))
109        {
110            buttonLearn.interactable = true; // Unlocked
111        }
112        else
113        {
114            buttonLearn.interactable = false; // Locked
115        }
116    }
117 }
118
119 1 reference
120 public void LearnMaterial(int unit)
121 {
122     Debug.Log("Clicked on section " + (unit + 1));
123     PlayerPrefs.SetInt("LearnUnit",unit);
124     SceneManager.LoadScene("ReadMaterial");
125 }
126 0 references
127 public void Practice()
128 {
129     SceneManager.LoadScene("Practice");
130 }

```

Gambar 4. 18 Script Game Manager

e. Membaca Materi

Script ini berfungsi untuk mengenerate materi dan mengatur responsive scroll view materi menyesuaikan berbagai device.

```

9 public class ReadManager : MonoBehaviour
10 {
11     public TMP_Text unitTitle;
12     public TMP_Text learnDesc;
13     public GameObject textContentPlace;
14     public TMP_Text textContent;
15     public GameObject textViewPort;
16     public GameObject panel;
17     // Start is called before the first frame update
18     void Start()
19     {
20         MaterialData materialData = textContent.GetComponent<MaterialData>();
21
22         int unit = PlayerPrefs.GetInt("LearnUnit");
23
24         if(LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
25         {
26             unitTitle.text = materialData.materialTitle[unit];
27             learnDesc.text = materialData.materialDesc[unit];
28             textContent.text = materialData.materialPoint[unit];
29         } else
30         {
31             unitTitle.text = materialData.materialTitle[0];
32             learnDesc.text = materialData.materialDesc[0];
33             textContent.text = materialData.materialPoint[0];
34         }
35
36         // Calculate preferred height
37         float preferredHeight = textContent.preferredHeight;
38
39         // Adjust the height of the RectTransform
40         RectTransform rectTransform = textContent.GetComponent<RectTransform>();
41         rectTransform.sizeDelta = new Vector2(rectTransform.sizeDelta.x, preferredHeight);
42
43         // Adjust the height of the RectTransform of the viewport
44         RectTransform viewportRectTransform = textViewPort.GetComponent<RectTransform>();
45         float panelHeight = panel.GetComponent<RectTransform>().rect.height;
46         float parentHeight = viewportRectTransform.parent.GetComponent<RectTransform>().rect.height;
47
48         // Calculate the offset needed to align the bottom of the viewport with the bottom of the panel
49         float bottomOffset = (parentHeight - panelHeight) + 50f;

```

```

50 viewportRectTransform.offsetMin = new Vector2(viewportRectTransform.offsetMin.x, bottomOffset);
51
52 }
53
54
55 0 references
56 public void PlayTest()
57 {
58     SceneManager.LoadScene("Test");
59 }
60 0 references
61 public void Exit()
62 {
63     SceneManager.LoadScene("MainMenu");
64 }

```

Gambar 4. 19 Script Membaca Materi

f. Quiz Manager

Script ini berfungsi untuk mengatur semua yang berkaitan dengan puzzle. Seperti mengenerate pertanyaan dan opsi jawaban, menentukan benar atau salahnya jawaban, dan mengkalkulasikan akurasi jawaban pemain dan menghitung durasi gameplay pemain.

```

9 public class QuizManager : MonoBehaviour
10 {
11     public string currentUnit;
12     public List<QuestionAnswer> Question;
13     public List<QuestionAnswer> unansweredQuestion;
14     public GameObject[] options;
15     public int currentQuestion;
16     public int questionCount;
17
18     public TMP_Text QuestionNumber;
19     public TMP_Text QuestionType;
20     public TMP_Text QuestionText;
21
22     private int correctAnswers = 0;
23     private int totalQuestionsAnswered = 0;
24
25     public float elapsedTime;
26
27     public GameObject correctPanel;
28     public GameObject incorrectPanel;
29     public GameObject submitBtn;
30
31     @ Unity Message | 0 references
32     private void Start()
33     {
34         correctPanel.SetActive(false);
35         incorrectPanel.SetActive(false);
36         submitBtn.SetActive(false);
37         InitializeQuestions();
38         PlayerPrefs.SetString("PreviousScene", SceneManager.GetActiveScene().name);
39     }
40     @ Unity Message | 0 references
41     private void Update()
42     {
43         elapsedTime += Time.deltaTime;
44         PlayerPrefs.SetFloat("ResultTimer", elapsedTime);
45     }
46     2 references
47     public void Correct()

```

```

47     {
48         correctAnswers++;
49         totalQuestionsAnswered++;
50         unansweredQuestion.RemoveAt(currentQuestion);
51         StartCoroutine(ShowCorrectPanel());
52     }
53 }
54
55 1 reference
56 IEnumerator ShowCorrectPanel()
57 {
58     correctPanel.SetActive(true);
59     yield return new WaitForSeconds(1.5f);
60     correctPanel.SetActive(false);
61     GenerateQuestion();
62 }
63
64 2 references
65 public void Wrong()
66 {
67     totalQuestionsAnswered++;
68     unansweredQuestion.RemoveAt(currentQuestion);
69     StartCoroutine(ShowIncorrectPanel());
70 }
71
72 1 reference
73 IEnumerator ShowIncorrectPanel()
74 {
75     incorrectPanel.SetActive(true);
76     yield return new WaitForSeconds(1.5f);
77     incorrectPanel.SetActive(false);
78     GenerateQuestion();
79 }
80
81 1 reference
82 void SetAnswer()
83 {
84     if (unansweredQuestion[currentQuestion].IsMultipleChoice)
85     {
86         for (int i = 0; i < options.Length; i++)
87         {
88             options[i].GetComponent<AnswerScript>().IsCorrect = false;
89             options[i].GetComponent<AnswerScript>().IsAnswer = false;
90         }
91         options[i].GetComponent<AnswerScript>().IsCorrect = true;
92     }
93     submitBtn.SetActive(false);
94 }
95
96 else
97 {
98     for (int i = 0; i < options.Length; i++)
99     {
100         options[i].GetComponent<AnswerScript>().IsCorrect = false;
101         options[i].GetComponent<AnswerScript>().IsAnswer = true;
102         options[i].GetComponent<AnswerScript>().IsCorrect = true;
103     }
104     submitBtn.SetActive(true);
105 }
106
107 public List<string> storedList = new List<string>();
108
109 1 reference
110 public string[] StoreArrangement(string userArrange)
111 {
112     storedList.Add(userArrange);
113     string[] storedArr = storedList.ToArray();
114     for (int i = 0; i < storedArr.Length; i++)
115     {
116         Debug.Log("array: " + storedArr[i]);
117     }
118     return storedArr;
119 }
120
121 2 references
122 public void ResetStoredList()
123 {
124     storedList.Clear();

```

```

124
125
126 public bool CheckArrangement(string[] userArrange)
127 {
128     bool correctArr = true;
129     if (userArrange.Length != unansweredQuestion[currentQuestion].CorrectArrangement.Length)
130     {
131         Debug.LogError("Length of user's arrangement does not match the correct arrangement length.");
132         correctArr = false;
133     }
134     else
135     {
136         for (int i = 0; i < userArrange.Length; i++)
137         {
138             if (userArrange[i].Trim() != unansweredQuestion[currentQuestion].CorrectArrangement[i])
139             {
140                 correctArr = false;
141             }
142         }
143     }
144     return correctArr;
145 }
146
147
148 #reference
149 void GenerateQuestion()
150 {
151     // Reset AnswerScript variables
152     AnswerScript[] answerScripts = FindObjectsOfType<AnswerScript>();
153     foreach (AnswerScript answerScript in answerScripts)
154     {
155         answerScript.ResetVariables();
156     }
157
158     if (unansweredQuestion.Count > 0)
159     {
160         currentQuestion = Random.Range(0, unansweredQuestion.Count);
161         var count = questionCount++;
162         QuestionNumber.text = count.ToString();
163         QuestionType.text = unansweredQuestion[currentQuestion].QuestionType;
164         QuestionText.text = unansweredQuestion[currentQuestion].QuestionText;
165         SetAnswer();
166     }
167     else
168     {
169         Debug.Log("Out of questions");
170         float accuracyPercentage = CalculateAccuracyPercentage();
171         Debug.Log("Accuracy Percentage: " + accuracyPercentage.ToString("0" + "%");
172         PlayerPrefs.SetFloat("PlayerResult", accuracyPercentage);
173         SceneManager.LoadScene("result");
174     }
175 }
176
177 #reference
178 float CalculateAccuracyPercentage()
179 {
180     if (totalQuestionsAnswered == 0)
181     {
182         return 0f;
183     }
184     return ((float)correctAnswers / totalQuestionsAnswered) * 100f;
185 }
186
187 #reference
188 void InitializeQuestions()
189 {
190     int unit = PlayerPrefs.GetInt("LearnUnit", -1);
191     if (unit == -1)
192     {
193         currentUnit = "PreTest";
194     }
195     else if (unit >= 0 && unit <= 6)
196     {
197         int incrementUnit = unit + 1;
198         currentUnit = "Unit " + incrementUnit.ToString();
199     }
200     else
201     {
202         currentUnit = "Practice";
203     }
204 }
205
206 if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
207 {
208     if (QuizData.Questions.ContainsKey(currentUnit))
209     {
210
211
212

```

```

203         Question = QuizData.Questions[currentUnit];
204         Debug.Log("Number of questions in the list: " + Question.Count);
205         unansweredQuestion = new List<QuestionAnswer>(Question),
206         GenerateQuestion();
207     }
208     {
209         Debug.LogError("Unit not found in the dictionary.");
210     }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }

```

Gambar 4. 20 Script Quiz Manager

g. Hasil Pemain

Script ini mengatur tampilan halaman hasil test pemain, mengatur hasil pre-test untuk menentukan section yang terbuka, mengatur hasil test materi untuk membuka section selanjutnya, mengatur level selesai, dan menambahkan tombol back dan retry

```

7 public class PlayerResult : MonoBehaviour
8 {
9     public TMP_Text accuracy;
10    public TMP_Text timer;
11    public Button continueBtn;
12    public int totalSections = 6;
13
14    public Image resultImage;
15
16    public GameObject familiarPanel;
17    public Image familiarImage;
18    public TMP_Text familiarName;
19
20    // Unity Message | 0 references
21    void Start()
22    {
23        familiarPanel.SetActive(false);
24        float accuracyValue = PlayerPrefs.GetFloat("PlayerResult");
25        float timerValue = PlayerPrefs.GetFloat("ResultTimer");
26        int minutes = Mathf.FloorToInt(timerValue / 60);
27        int seconds = Mathf.FloorToInt(timerValue % 60);
28        accuracy.text = accuracyValue.ToString("0") + "%";
29        timer.text = string.Format("{0:00}:{1:00}", minutes, seconds);
30
31        int unit = PlayerPrefs.GetInt("LearnUnit", -1);
32
33        if (accuracyValue == 100f)
34        {
35            resultImage.sprite = Resources.Load<Sprite>("Images/Result/Chip");
36
37            if (timerValue < 60f)
38            {
39                familiarImage.sprite = Resources.Load<Sprite>("Images/Result/khodan_1");
40                if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
41                {
42                    familiarName.text = "Einstein Spirit";
43                } else
44                {
45                    familiarName.text = "Arwah Einstein";
46                }
47            }
48        }
49    }
50 }

```

```

47     }
48     else
49     {
50         familiarImage.sprite = Resources.Load<Sprite>("Images/Result/khodam_2");
51         if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
52         {
53             familiarName.text = "Bootleg Flash";
54         }
55         else
56         {
57             familiarName.text = "Flash MW";
58         }
59     }
60     else if (accuracyValue >= 50f)
61     {
62         resultImage.sprite = Resources.Load<Sprite>("Images/Result/Chip");
63         if (timerValue < 60f)
64         {
65             familiarImage.sprite = Resources.Load<Sprite>("Images/Result/khodam_3");
66             if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
67             {
68                 familiarName.text = "L Ninja";
69             }
70             else
71             {
72                 familiarName.text = "Ninja Kopleset";
73             }
74         }
75         else
76         {
77             familiarImage.sprite = Resources.Load<Sprite>("Images/Result/khodam_4");
78             if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
79             {
80                 familiarName.text = "Sadge Turtle";
81             }
82             else
83             {
84                 familiarName.text = "Kura-kura Sadboy";
85             }
86         }
87     }
88     else

```

```

89     {
90         resultImage.sprite = Resources.Load<Sprite>("Images/Result/failed_test");
91         familiarImage.sprite = Resources.Load<Sprite>("Images/Result/khodam_4");
92         if (LocalizationSettings.SelectedLocale == LocalizationSettings.AvailableLocales.Locales[0])
93         {
94             familiarName.text = "Sadge Turtle";
95         }
96         else
97         {
98             familiarName.text = "Kura-kura Sadboy";
99         }
100     }
101     if (unit < 0)
102     {
103         HandlePreTestResult(accuracyValue);
104     }
105     else
106     {
107         HandleTestResult(accuracyValue);
108     }
109 }
110
111
112 1 reference
113 void HandlePreTestResult(float accuracyValue)
114 {
115     if (accuracyValue == 100f)
116     {
117         PlayerPrefs.SetInt("AlreadyPlay", 1);
118         PlayerPrefs.SetInt("GameMode", 3); // Hard mode
119         PlayerPrefs.SetInt("UnlockedSections", 5);
120     }
121     else if (accuracyValue >= 50f)
122     {
123         PlayerPrefs.SetInt("AlreadyPlay", 1);
124         PlayerPrefs.SetInt("GameMode", 2); // Medium mode
125         PlayerPrefs.SetInt("UnlockedSections", 3);
126     }
127     else
128     {
129         Debug.Log("Pretest failed. Please retry."); // Pretest failed
130         continueBtn.interactable = false;

```

```
130     }
131     }
132
133     1 reference
134     void HandleTestResult(float accuracyValue)
135     {
136         if (accuracyValue >= 50f)
137         {
138             Debug.Log("Pass");
139             LevelFinished(PlayerPrefs.GetInt("LearnUnit"));
140         }
141         else
142         {
143             Debug.Log("Failed"); // Failed test
144         }
145     }
146
147     1 reference
148     void LevelFinished(int sectionIndex)
149     {
150         string sectionKey = "sectionCompleted_" + sectionIndex;
151
152         if (PlayerPrefs.GetInt(sectionKey) == 0)
153         {
154             PlayerPrefs.SetInt("UnlockedSections", PlayerPrefs.GetInt("UnlockedSections") + 1);
155             PlayerPrefs.SetInt(sectionKey, 1);
156
157             if (PlayerPrefs.GetInt("UnlockedSections") == 3)
158             {
159                 PlayerPrefs.SetInt("GameMode", 2);
160             } else if (PlayerPrefs.GetInt("UnlockedSections") == 5)
161             {
162                 PlayerPrefs.SetInt("GameMode", 3);
163             }
164         }
165         else
166         {
167             Debug.Log("Section " + sectionIndex + " is already completed");
168         }
169     }
170
171     0 references
172     public void Retr()
173     {
174         SceneManager.LoadScene("Test");
175     }
176
177     0 references
178     public void Continue()
179     {
180         SceneManager.LoadScene("MainMenu");
181     }
182
183     0 references
184     public void CheckFamiliar()
185     {
186         familiarPanel.SetActive(true);
187     }
188
189     0 references
190     public void ExitCheck()
191     {
192         familiarPanel.SetActive(false);
193     }
194 }
```

Gambar 4. 21 Script Hasil Pemain

h. Practice Manager

Menuju halaman practice dengan menentukan learn unit nya selain learn unit yang tersedia

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class PracticeManager : MonoBehaviour
7 {
8     public void PracticeTest()
9     {
10         PlayerPrefs.SetInt("LearnUnit", 99);
11         SceneManager.LoadScene("Test");
12     }
13
14     public void Exit()
15     {
16         SceneManager.LoadScene("MainMenu");
17     }
18 }
```

Gambar 4. 22 Script Practice Manager

i. About Game

Script yang mengatur perpindahan scene dari halaman tentang ke halaman menu awal

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class AboutGame : MonoBehaviour
7 {
8     public void BackToStart()
9     {
10         SceneManager.LoadScene("StartMenu");
11     }
12 }
```

4.2 Pengujian

4.2.1 User Acceptance Testing (UAT)

UAT pada Game berfokus pada kebutuhan fungsional system. Tujuan dari pengetesan UAT ini adalah untuk memastikan bahwa fungsi-fungsi pada game sudah berjalan dengan baik dan benar. Apabila soal tidak tepat, maka akan dilakukan review kembali terhadap soal-soal yang disediakan, namun apabila tepat, maka soal dinyatakan “Sesuai” sementara fungsi yang tidak berhasil ditandai dengan keterangan “Tidak Sesuai”. UAT dilakukan oleh dosen Teknik Informatika Politeknik Caltex Riau dengan bukti validasi terlampir pada lampiran C.

1) Pengujian halaman utama

Tabel 4. 1 Pengujian Halaman Utama

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Mulai	Menampilkan halaman mode kesulitan ketika pemain baru pertama kali bermain	Menampilkan halaman mode kesulitan dengan 3 kesulitan	Sesuai
<i>Button</i> Mulai	Menampilkan halaman menu module ketika pemain sudah memilih mode kesulitan	Menampilkan halaman module dengan section sesuai mode kesulitan yang dipilih	Sesuai
<i>Button</i> Tentang	Menampilkan informasi tentang game	Adanya informasi mengenai game dan <i>creator</i>	Sesuai
<i>Button</i> Keluar	Game berhenti berjalan	Game berhenti berjalan	Sesuai

2) Pengujian halaman mode kesulitan

Tabel 4. 2 Pengujian halaman mode kesulitan

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> I'm New/Saya Pemula	Menampilkan halaman module	Menampilkan halaman module dengan 1 section terbuka	Sesuai
<i>Button</i> I'm intermediate/Saya tingkat menengah	Menampilkan halaman pre-test	Menampilkan halaman pre-test dengan soal pre-test yang sudah disediakan	Sesuai

3) Pengujian halaman module

Tabel 4. 3 Pengujian Halaman Module

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Learn/Pelajari	Menampilkan halaman materi	Menampilkan halaman materi sesuai dengan section yang dipilih	Sesuai
<i>Button</i> Practice	Menampilkan halaman practice	Menampilkan halaman practice	Sesuai
<i>Button</i> Kembali	Kembali ke halaman menu awal	Kembali ke halaman menu awal	Sesuai

4) Pengujian halaman materi

Tabel 4. 4 Pengujian Halaman Materi

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Test	Menampilkan	Menampilkan	Sesuai

	halaman test	halaman test materi sesuai dengan materi yang dipilih	
<i>Button</i> Kembali	Kembali ke halaman module	Kembali ke halaman module	Sesuai

5) Pengujian halaman practice

Tabel 4. 5 Pengujian Halaman Practice

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Test	Menampilkan halaman test	Menampilkan halaman test dengan soal latihan yang terdiri dari semua materi	Sesuai
<i>Button</i> Kembali	Kembali ke halaman module	Kembali ke halaman module	Sesuai

6) Pengujian halaman test/pre-test

Tabel 4. 6 Pengujian Halaman Test/Pre-test

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Opsi Jawaban	Lanjut ke soal selanjutnya	lanjut ke soal selanjutnya ketika menekan jawaban yang benar/jawaban yang salah	Sesuai
<i>Button</i> Kembali	Kembali ke halaman materi/module/memilih kesulitan	Kembali ke halaman materi/module/memilih kesulitan berdasarkan darimana akses test/pre-test diperoleh	Sesuai

7) Pengujian halaman result

Tabel 4. 7 Pengujian Halaman Result

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Retry/ulang	Menampilkan halaman test/pre-test	Menampilkan halaman test/pre-test sebelumnya	Sesuai
<i>Button</i> Continue/selanjutnya	Kembali ke halaman module	Kembali ke halaman module	Sesuai

8) Pengujian halaman tentang

Tabel 4. 8 Pengujian Halaman Tentang

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
<i>Button</i> Kembali	Kembali ke halaman menu awal	Kembali ke halaman menu awal	Sesuai

4.2.2 Beta Testing

Pada pengujian beta testing ini dosen diminta untuk mencoba bermain game coding puzzle yang bertujuan untuk melihat aplikasi game secara keseluruhan baik itu dari error maupun keluhan dan saran yang perlu dilakukan penambahan ke aplikasi sebelum aplikasi dilakukan perilsan, beta testing dilakukan ke pada dosen dari prodi Teknik Informatika PCR seperti yang terlampir pada lampiran C yang menghasilkan saran dalam menambahkan difficulty advance untuk mengetes kemampuan mahasiswa. Adapun tampilan awal dari aplikasi sebagai berikut.



Gambar 4. 23 Gambar sebelum beta testing

Kemudian game setelah melalui tahap beta testing dan dilakukan penambahan section baru yaitu “Practice” dengan soal yang terdiri dari semua materi yang ada di game.



Gambar 4. 24 Gambar setelah beta testing

Tampilan halaman practice yang berisi button test dengan test soal yang terdiri dari seluruh materi yang ada di game.



Gambar 4. 25 Tampilan Penambahan Halaman Practice

4.2.3 Pengujian Validasi Konten

Berdasarkan validasi yang telah diberikan dan dilakukan oleh dua dosen dari prodi Teknik Informatika PCR seperti yang telah terlampir di halaman lampiran C yaitu bapak Erzi Hidayat, S.T., M.Kom. dan ibu Yuliska, S.T., M.Eng. bahwasannya game coding puzzle yang telah dibuat dan ditujukan untuk mahasiswa ini telah sesuai dengan kebutuhan pembelajar algoritma dan pemrograman.

Tabel 4. 9 Validasi Konten Section Variable

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-
Soal “lengkapi kode ini”	Sesuai	-
Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

Tabel 4. 10 Validasi Konten Section Operator

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-

Soal “lengkapi kode ini”	Sesuai	-
Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

Tabel 4. 11 Validasi Konten Section Input/Output

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-
Soal “lengkapi kode ini”	Sesuai	-
Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

Tabel 4. 12 Validasi Konten Section Control Statement

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-
Soal “lengkapi kode ini”	Sesuai	-
Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

Tabel 4. 13 Validasi Konten Section Array

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-
Soal “lengkapi kode ini”	Sesuai	-
Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

Tabel 4. 14 Validasi Konten Section Exception Handling

Konten	Sesuai/Tidak Sesuai	Keterangan
Materi	Sesuai	-
Soal “apa outputnya”	Sesuai	-
Soal “lengkapi kode ini”	Sesuai	-

Soal “temukan bug di kode ini”	Sesuai	-
Soal “susun kode ini”	Sesuai	-

4.2.4 Pengujian Responden (Pre-test dan Post-test)

Pengujian ini dilakukan dengan pre-test dan post-test dengan tujuan melihat pengaruh dan perbedaan hasil belajar mahasiswa Teknik Informatika PCR dengan bukti pengujian terlampir pada lampiran D. Pre-test dan post-test ini juga berfungsi untuk melihat tingkat keefektifan game ini digunakan untuk mahasiswa generasi 20 sampai generasi 23.

Pengujian ini dilakukan secara online menggunakan google form yang disediakan, dimana google form tersebut disebarakan melalui email ke mahasiswa G20-G23 dan ke dosen-dosen Teknik Informatika PCR untuk memperoleh relawan berjumlah 15 orang untuk melakukan pengujian.

Dalam pengujian terdapat beberapa tahapan, tahapan pertama melakukan pengujian secara konvensional di mana mahasiswa diberi pre-test terlebih dahulu, tahap pertama selesai ketika mahasiswa telah mengerjakan semua soal pada soal pre-test.

Pada tahap kedua, mahasiswa diminta untuk memainkan game bagian section materi yang mana mahasiswa diminta untuk mempelajari terlebih dahulu materi-materi pemrograman Java yang ada.

Pada tahap ketiga, mahasiswa diminta untuk menyelesaikan bagian practice yang mana mahasiswa akan diberi 12 soal dengan setiap soal poinnya 1.

Tabel 4. 15 Tabel perbandingan hasil pre-test dan post-test

No	Keterangan	Pre-test	Post-test	Status
1	Responden 1	83	100	Meningkat
2	Responden 2	83	100	Meningkat
3	Responden 3	83	100	Meningkat
4	Responden 4	100	100	Tetap
5	Responden 5	75	75	Tetap
6	Responden 6	75	75	Tetap
7	Responden 7	50	75	Meningkat
8	Responden 8	75	83	Meningkat
9	Responden 9	83	100	Meningkat
10	Responden 10	83	100	Meningkat

11	Responden 11	75	100	Meningkat
12	Responden 12	92	100	Meningkat
13	Responden 13	67	100	Meningkat
14	Responden 14	83	100	Meningkat
15	Responden 15	83	100	Meningkat

Pada tabel 4.15 berisikan keterangan responden dan data nilai pretest dan post-test yang didapat dari setiap responden. pre-test merupakan nilai yang didapat dari hasil jawaban mahasiswa dalam menjawab soal yang diberikan secara konvensional melalui google form dan nilai dari post-test disini didapat dari hasil mahasiswa mengerjakan practice pada game yang mana sebelumnya mahasiswa telah menyelesaikan materi terlebih dahulu.

Rumus yang digunakan untuk mengetahui persentase peningkatan pengetahuan pemrograman java secara keseluruhan dengan menggunakan game sebagai berikut:

$$\text{Persen (\%)} = \frac{\text{jumlah peningkatan}}{\text{Jumlah seluruh mahasiswa}} \times 100\%$$

Dari data tabel diatas menunjukan bahwa 12 responden mengalami peningkatan nilai setelah memainkan game coding puzzle yang berarti peningkatan keseluruhan di angka 80%.

Pengujian pre-test dan post-test dilakukan dengan memberi waktu bebas kepada mahasiswa untuk menyelesaikan soal yang ada. Berikut adalah hasil waktu yang berhasil dicatat dari pre-test dan post-test.

Tabel 4. 16 waktu pengerjaan pre-test dan post-test

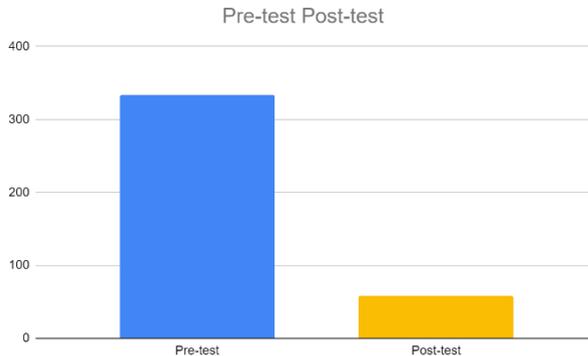
No	Nama	Pre-test		Post-test	
		Total Waktu	Rata-rata per-soal	Total Waktu	Rata-rata per-soal
1	Responden 1	180	15	48	4
2	Responden 2	300	25	31	3
3	Responden 3	420	35	35	3
4	Responden 4	240	20	47	4
5	Responden 5	135	11	73	6
6	Responden 6	180	15	176	15

7	Responden 7	320	27	64	5
8	Responden 8	600	50	186	16
9	Responden 9	360	30	44	4
10	Responden 10	240	20	31	3
11	Responden 11	720	60	22	2
12	Responden 12	120	10	38	3
13	Responden 13	240	20	24	2
14	Responden 14	44	4	31	3
15	Responden 15	900	75	36	3

Dengan rata-rata waktu pengerjaan adalah sebagai berikut:

Tabel 4. 17 Rata-rata waktu pengerjaan

Rata-rata waktu	
Pre-test	Post-test
333	59



Gambar 4. 26 Rata-rata waktu pre-test dan post-test

4.2.5 Pengujian Kuesioner kepuasan pengguna

Pengujian kepuasan pengguna ini dilakukan kepada 15 orang mahasiswa yang telah melakukan pengujian pre-test dan post-test dan juga telah mencoba untuk memainkan game dan menyelesaikan practice, pengujian ini ditujukan sebagai acuan untuk memastikan fungsionalitas game dan desain dari game coding puzzle ini telah sesuai dengan

kebutuhan perancangan saat game ini dibangun. Pada kuesioner yang diberikan maka didapatkan rekapitulasi hasil sebagai berikut:

Tabel 4. 18 Kuesioner kepuasan pengguna

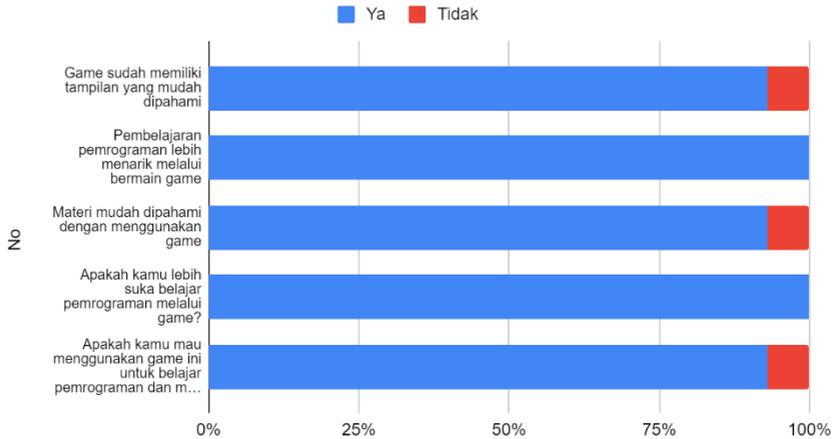
No	Pertanyaannya	Jumlah jawaban		Persentase	
		Ya	Tidak	Ya	Tidak
1	Game sudah memiliki tampilan yang mudah dipahami	14	1	93%	7%
2	Pembelajaran pemrograman lebih menarik melalui bermain game	15	0	100%	0%
3	Materi mudah dipahami dengan menggunakan game	14	1	93%	7%
4	Apakah kamu lebih suka belajar pemrograman melalui game?	15	0	100%	0%
5	Apakah kamu mau menggunakan game ini untuk belajar pemrograman dan membiasakannya dirumah?	14	1	93%	7%

Berdasarkan tabel 4.18 menunjukkan bahwa tampilan game yang mudah dipahami, materi mudah dipahami melalui game, dan keinginan

untuk membiasakan penggunaan game dalam pembelajaran pemrograman java dirumah mendapatkan angka 93%. Sementara itu, untuk pembelajaran lebih menarik melalui game dan kesukaan belajar pemrograman melalui game mencapai angka 100%.

Berikut merupakan grafik data kepuasan pengguna yang dapat dari hasil penelitian terhadap mahasiswa Teknik Informatika PCR.

Kepuasan Pengguna



Gambar 4. 27 Grafik Kepuasan Pengguna

Dari penjelasan gambar 4.27 tingkat kepuasan pengguna dari 5 aspek yang telah dilakukan pengujian terlihat bahwa tingkat kepuasan terendah berada pada tampilan yang mudah dipahami, materi, dan keinginan belajar dirumah yang mana masih berada diangka 93% dan tingkat kepuasan pengguna tertinggi berada pada pembelajaran yang lebih menarik melalui game, dan kesukaan belajar melalui game.

4.3 Analisis

4.3.1 Analisis User Acceptance Testing (UAT)

Dilihat dari pengujian UAT (User Acceptance Testing) game coding puzzle yang telah dibangun telah sesuai secara fungsional dari setiap fitur dan menu yang tersedia semua telah berjalan sesuai dengan tujuan dan fungsi yang diharapkan. Dikarenakan proses pengembangan dan pengujian internal yang ketat telah memastikan bahwa setiap komponen diuji secara menyeluruh untuk kompatibilitas dan kinerja. Pengembang juga melakukan iterasi perbaikan berdasarkan umpan balik awal dari pengguna, sehingga mengurangi kemungkinan adanya bug atau kesalahan dalam versi akhir.

4.3.2 Analisis Validasi Konten

Sesuai dengan hasil validasi konten yang dilakukan kepada dosen Teknik Informatika PCR yang mana konten yang ditampilkan pada game coding puzzle telah sesuai dengan standar dari pembelajaran algoritma dan pemrograman untuk mahasiswa PCR. Dikarenakan dosen-dosen yang terlibat dalam validasi merupakan ahli di bidang algoritma dan pemrograman. Para dosen telah memastikan konten sesuai dengan kurikulum yang ada dan kebutuhan pembelajaran mahasiswa. Sehingga, game coding puzzle ini telah siap untuk di release dan dimainkan untuk mahasiswa sebagai salah satu metode pembelajaran pemrograman java.

4.3.3 Analisis Responden(Pre-test dan Post-test)

Sesuai dengan hasil pengujian pre-test dan post-test yang dilakukan yaitu sebesar 80% dari 15 mahasiswa PCR yang diujikan menghasilkan peningkatan nilai dari nilai pre-test dan post-test nya. Dikarenakan game coding puzzle ini menyediakan pendekatan pembelajaran yang interaktif dan *engaging*, yang memfasilitasi pemahaman yang lebih baik terhadap konsep-konsep pemrograman. Interaktivitas game memungkinkan mahasiswa untuk langsung mempraktikkan apa yang mereka pelajari, sehingga meningkatkan retensi pengetahuan. Dapat diartikan game coding puzzle ini telah sesuai dan cocok untuk dijadikan salah satu metode pembelajaran dan pembiasaan pembelajaran pemrograman bagi mahasiswa. Game coding puzzle dapat

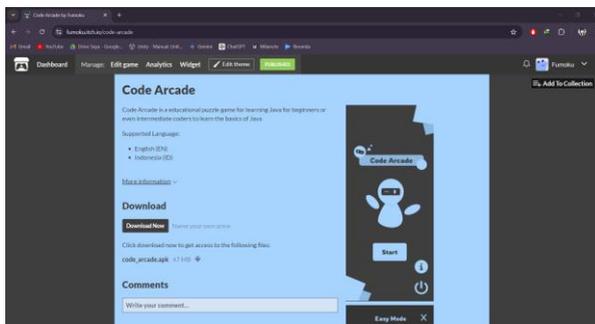
difokuskan penggunaannya kepada mahasiswa Teknik Informatika yang mengalami peningkatan secara keseluruhan.

4.3.4 Analisis Kepuasan Pengguna

Sesuai dengan hasil pengujian kepuasan pengguna yang telah diujikan maka tingkat kepuasan pengguna terbawah diangka 93% yaitu pada bagian tampilan yang mudah dipahami, materi yang mudah dipahami, dan keinginan untuk membiasakan pembelajaran dirumah karena game ini dirancang dengan antarmuka pengguna yang intuitif dan materi yang disajikan secara sederhana dan jelas, sehingga mudah diikuti oleh mahasiswa. Adanya struktur yang terorganisir dengan baik membantu mahasiswa untuk mempelajari materi secara bertahap tanpa merasa kewalahan. Untuk bagian lainnya berada di angka 100% dimana berarti mahasiswa yang diujikan tertarik dalam belajar menggunakan game karena elemen gamifikasi yang diterapkan dalam game ini membuat proses belajar lebih menarik dan menyenangkan, sehingga mahasiswa merasa lebih termotivasi dan tertarik untuk terus menggunakan game ini sebagai alat bantu belajar, dimana game ini dapat dijadikan salah satu alternatif pembelajaran pemrograman java.

4.4 Release

Game di release ke dalam marketplace itch.io dengan format apk yang bisa di download langsung pada link berikut: <https://fumoku.itch.io/code-arcade>.



Gambar 4. 28 Website Instalasi Game coding puzzle yang direlease

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Game coding puzzle untuk mahasiswa PCR berhasil dibangun dengan penerapan metode GDLC dan di buat dengan format aplikasi yang dipasang ke perangkat android.
2. Game coding puzzle untuk mahasiswa PCR berdasarkan hasil validasi konten kepada dosen Teknik Informatika PCR telah sesuai dengan standar dari pembelajaran algoritma dan pemrograman untuk mahasiswa PCR. Sehingga, game coding puzzle ini telah siap untuk di release dan dimainkan untuk mahasiswa sebagai salah satu metode pembelajaran pemrograman java
3. Game coding puzzle untuk mahasiswa PCR ini dapat meningkatkan minat belajar mahasiswa terutama dalam hal pemrograman yang mana mereka suka belajar sambil bermain melalui game yang mencapai angka 100% dan mereka tertarik untuk membiasakan penggunaan belajar pemrograman melalui game ini dirumah bahkan hasilnya sampai di angka 93%.
4. Game coding puzzle untuk mahasiswa PCR ini dapat meningkatkan nilai dan pengetahuan mereka terkait materi pemrograman, dapat dilihat berdasarkan hasil survey perbandingan hasil sebelum dan setelah bermain yang mana 80% mahasiswa mengalami peningkatan secara keseluruhan.

5.2 Saran

Adapun saran untuk pengembangan proyek akhir selanjutnya ialah penambahan materi algoritma dan pemrograman yang lebih lengkap

DAFTAR PUSTAKA

- Agung Saputra, A., Nonggala Putra, F., & Darma Rusdian Yusron, R. (2022). Pembuatan Game Edukasi Pengenalan Kebudayaan Indonesia Menggunakan Metode Game Development Life Cycle (GDLC) Berbasis Android Design an Educational Game Introducing Indonesian Culture Using the Android-Based Game Development Life Cycle (GDLC) Method. *JACIS: Journal Automation Computer Information System*, 2(1), 66–73.
- Dwi Puspitasari, W., & Febrinita, F. (2021). PENGUJIAN VALIDASI ISI (CONTENT VALIDITY) ANGKET PERSEPSI MAHASISWA TERHADAP PEMBELAJARAN DARING MATAKULIAH MATEMATIKA KOMPUTASI. *Focus ACTION Of Research Mathematic*, 4(1). <https://doi.org/10.30762/factor-m.v4i1.3254>
- GuruBelajar.ID. (2022, September 20). *Dijamin Seru! Inilah Manfaat dan Contoh Game Puzzle untuk Meningkatkan Kemampuan Kognitif Siswa*. <https://gurubelajar.id/dijamin-seru-inilah-manfaat-dan-contoh-game-puzzle-untuk-meningkatkan-kemampuan-kognitif-siswa/>
- Harlanto, R. A. (2020, July 16). *Berkenalan dengan Fitur-Fitur Unity 3D*. <https://www.gamelab.id/news/211-berkenalan-dengan-fitur-fitur-unity-3d>
- Hartanto, Y., Sanjaya, T., & Try Windranata, dan. (2020). Pengaruh Game Berbasis Coding Terhadap Keberlanjutan Minat Belajar Programming Siswa di Batam, Indonesia. *Jurnal Teknologi Informasi*, 6. <http://ejournal.urindo.ac.id/index.php/TI>
- Khaerudin, M., Srisulistiowati, D. B., & Warta, J. (2021). GAME EDUKASI DENGAN MENGGUNAKAN UNITY 3D UNTUK MENUNJANG PROSES PEMBELAJARAN. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, 8(2).
- Nirwana, N. C., & Purwanto, A. (2022). Pengembangan Teknologi Game Indonesia “Pramuka Asik” Menggunakan Unity 2d Engine Berbasis Android. *Jurnal Pendidikan Tambusai*, 6(1).
- Pradana, A. G., & Nita, S. (2019). Rancang Bangun Game Edukasi “AMUDRA”Alat Musik Daerah Berbasis Android. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 2(1).

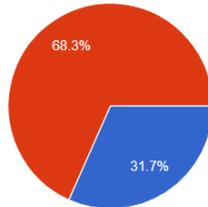
<http://prosiding.unipma.ac.id/index.php/SENATIK/article/download/1062/925>

- Ramadhan, A. R. (2023). RANCANG BANGUN GAME EXPLORE SUMATERA ISLAND MENGGUNAKAN TOOLS CONSTRUCT 2 BERBASIS ANDROID. *Teknologipintar.Org*, 3(4).
- Riyanti, & Yahfizham. (2023). Konsep Dasar Algoritma Pemrograman. *Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa Dan Matematika*, 1(6).
- Rohmawati, I., & Menarianti, I. (2019). PENGEMBANGAN GAME EDUKASI TENTANG BUDAYA NUSANTARA “TANARA” MENGGUNAKAN UNITY 3D BERBASIS ANDROID. *JURNAL SISTEM INFORMASI DAN TEKNOLOGI*, 2(2).
<http://www.jurnal.umk.ac.id/sitech>
- Sahid, D. S. S. (2020). *PENGANTAR KONSEP PEMROGRAMAN*.
<https://classroom.google.com/c/MTc1MzI1OTI3NDQ3/m/MTc4MzM3NTI3NjEx/details>
- Sarah Juliandiny. (2023, March 13). *Mengenal Lebih Lanjut tentang User Acceptance Test (UAT)*. <https://pacmann.io/blog/user-acceptance-test>
- Silvana, J., & Anistyasari, Y. (2021). Pengembangan Game “Script Labyrinth” untuk Meningkatkan Computational Thinking Siswa dalam Mata Pelajaran Pemrograman Web dan Perangkat Bergerak di SMKN 2 Surabaya PENGEMBANGAN GAME “SCRIPT LABYRINTH” UNTUK MENINGKATKAN COMPUTATIONAL THINKING SISWA DALAM PELAJARAN PEMROGRAMAN WEB DAN PERANGKAT BERGERAK DI SMKN 2 SURABAYA. *Jurnal IT-EDU*, 5(2).
www.pngdownload.id
- Syaliman, K. U., Najwa, N. F., & Kreshna, J. A. (2022). EDUCATIONAL GAME AS AN EFFORT TO ACCELERATE LEARNING AFTER THE COVID-19 PANDEMIC. *Journal of Applied Engineering and Technological Science*, 4(1).
- Wilman Juniardi. (2022, December 22). *Pengertian Post Test beserta Tujuan, Fungsi, dan Contoh Soal untuk Guru*.

LAMPIRAN A – KUESIONER PERMASALAHAN

Apakah Anda memiliki pengetahuan dasar tentang pemrograman sebelum memasuki program studi Teknik Informatika di PCR? [Copy](#)

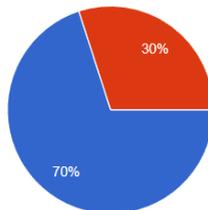
60 responses



- Ya, saya memiliki pengetahuan dasar pemrograman.
- Tidak, saya tidak memiliki pengetahuan dasar pemrograman.

Apakah Anda mengalami kesulitan dalam memahami konsep pemrograman selama kuliah? [Copy](#)

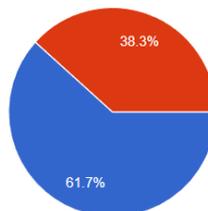
60 responses



- Ya, saya mengalami kesulitan.
- Tidak, saya tidak mengalami kesulitan.

Apakah penjelasan materi pemrograman di perkuliahan mudah dipahami? [Copy](#)

60 responses

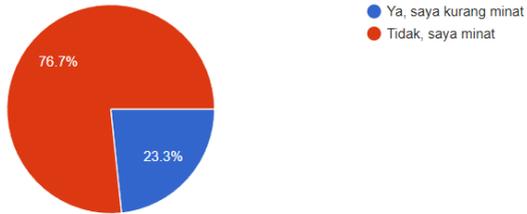


- Ya, mudah dipahami
- Tidak, susah dipahami

Apakah anda merasa kurang minat dalam mengikuti proses belajar pemrograman selama kuliah?

 Copy

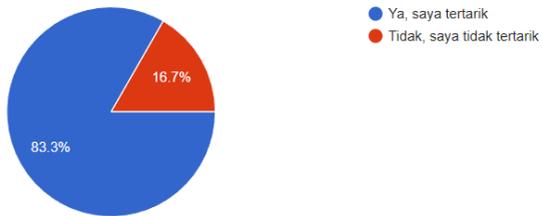
60 responses



Apakah anda minat untuk mengeksplere sendiri pemrograman diluar kuliah?

 Copy

60 responses



LAMPIRAN B – PUZZLE

1
×

Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        double y = 3.5;
        System.out.println(x + y);
    }
}
```

0.5

9

0.35

53

2
×

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int num = 10;
        System.out.println(num);
    }
}
```

int num = 10;

int main = 10;

int main += 10;

int main = 10;

3
×

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        _____ x = 5.5;
    }
}
```

int

String

float

boolean

4
×

Susun Kode Ini

int x = 5;

System.out.println()

public static void main(String[] args) {

}

Submit

1
×

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int result = 10 _____ 5; // Isi operator yang sesuai!
        System.out.println(result);
    }
}
```

*

..

/

%-

2
×

Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        boolean a = true;
        boolean b = false;
        System.out.println(a && b);
    }
}
```

true

false

0

1

3
×

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int num1 = 15;
        int num2 = 5;
        // Kode yang sudah diperbaiki:
        int result = num1 % num2;
        int result = num1 % num2; // Identifikasi bug
        System.out.println(result);
    }
}
```

Operator % hanya dapat digunakan pada tipe data integer.

Komentar barisan ketiga di dalam main() harus.

Variabel hasil dipanggil sebagai hasil.

Tidak ada bug di kode.

4
×

Susun Kode Ini

int x = 5;

System.out.println(2 + 5);

public static void main(String[] args) {

}

Submit

1
×

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in)
        // baca input pengguna
        System.out.println("hello, " + name);
    }
}
```

String name = scanner.nextLine();

"hello" + name;

String name = "Ali";

String name = args[0];

2
×

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        String sc = new Scanner(System.in)
        int num = sc.nextInt();
        System.out.println("Anda memasukkan: " + num);
    }
}
```

Tidak ada bug di kode.

scanner tidak bekerja.

% nextInt() harus dipisahkan oleh spasi.

Kode tidak menggunakan pemanggilan scanner dengan baik.

3
×

Susun Kode Ini

System.out.println("Masukkan nama Anda: ");

String name = new Scanner(System.in).nextLine();

public static void main(String[] args) {

}

Submit

4
×

Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Masukkan usia Anda: ");
        int age = new Scanner(System.in).nextInt();
        System.out.println("Usia Anda adalah: " + age);
    }
}
```

Program menginput program dan keluar.

Program meminta usia pengguna dan memantapnya.

Kode akan gagal dengan hasil lain.

Memiliki file bernama "usia.txt".

1 Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 3; i++) {
            sum += i;
        }
        System.out.println(sum);
    }
}
```

0 1

3 6

2 Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i <= 5; i++) { // Temukan bug
            System.out.println(i);
        }
    }
}
```

Lupa menambahkan kurung kurawal setelah kurawal. Lupa menambahkan kurung kurawal setelah kurawal.

Variable harus diinisialisasi sebelum diakses. Tidak ada bug di kode diatas.

3 Susun Kode Ini

```
int i = 1;      while (i <= 100) {
    System.out.println("halo");
}
```

Submit

4 Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int age = 25;
        if (age _____ 18) { // Isikan operator perbandingan yang sesuai
            System.out.println("Anda sudah dewasa.");
        }
    }
}
```

... ...

... ...

1 Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        String[] names = {"Alice", "Bob", "Charlie"};
        System.out.println(names[1]);
    }
}
```

Program diampatkan dengan karaktir. Hal memuat array element dari array.

Hal memuat array. Hal memuat element kedua (Bob).

2 Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        System.out.println(numbers[6]); // Temukan bug
    }
}
```

array yang diinisialisasi dengan ukuran 5, tetapi diakses pada index ke-6. Element element array di urutan terbalik.

Lupa melubuhkan kurva atau kudi diatas. Tidak ada bug di kode.

3 Susun Kode Ini

```
int[] matrix = {1, 2, 3, 4, 5, 6};
System.out.println(matrix[0]);
```

public static void main(String[] args) { for (int i = 0; i < matrix.length; i++)

Submit

4 Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = new int[5]; // Deklarasikan sebuah array numbers[2] = ____; // Berikan nilai pada elemen ke-3
        System.out.println(numbers[2]);
    }
}
```

10 2.5

{1, 2, 5} numbers.length

1 Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int num = Integer.parseInt("hello"); // Baris ini mungkin menyebabkan pengecualian
        System.out.println(num);
    }
}
```

Program diatas tidak diinput. Mencoba parse nilai yang dapat diinput ke dalam variabel.

int adalah tipe data yang digunakan sebagai string. Tidak ada bug di kode.

2 Susun Kode Ini

```
System.out.println("valid input");
```

try { try {
 Integer.parseInt("10");
 NumberFormatException e;
}

Submit

3 Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        int result = 10 / 0;
        catch (ArithmeticException e) {
            System.out.println("Pembagian dengan nol!");
        }
    }
}
```

Program diampatkan dengan karaktir. Integer adalah tipe data yang digunakan sebagai variabel result.

Hal menerima hasil pengacakan yang tidak diinginkan. Hal memuat pembagian dengan nol.

4 Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = new int[2];
        try {
            numbers[2] = 10; // Baris ini mungkin menyebabkan pengecualian
        } catch (_____ e) { // Isikan jenis pengecualian yang sesuai
            System.out.println("Indeks array di luar batas!");
        }
    }
}
```

IndexOutOfBoundsException ArrayIndexOutOfBoundsException

IntegerException NumberFormatException

1

Apa Outputnya?

```
import java.util.Scanner;
public class Utama {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Masukkan sebuah angka: ");
        int x = scanner.nextInt();
        System.out.println("Anda memasukkan: " + x);
    }
}
```

Masukkan sebuah angka. Anda memasukkan 5

Masukkan sebuah angka: 5

Masukkan sebuah angka. Anda memasukkan

Submit

2

Susun Kode Ini

```
public class Utama {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Masukkan sebuah angka: ");
        int x = scanner.nextInt();
        System.out.println("Anda memasukkan: " + x);
    }
}
```

public static void main(String[] args) {

System.out.println("Masukkan sebuah angka: ");

Submit

3

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 5;
        int z = x // y;
        System.out.println(z);
    }
}
```

//

*

/

System.out.println

Submit

4

Susun Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int result = 10 / 0;
    }
}
```

System.out.println("Error by user.");

try {

int result = 10 / 0;

catch (ArithmeticException e) {

Submit

5

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(x);
    }
}
```

int

String

double

float

6

Apa Outputnya?

```
public class Utama {
    public static void main(String[] args) {
        int x = 5;
        if (x > 2) {
            System.out.println("Lebih Besar?");
        } else {
            System.out.println("Lebih Kecil?");
        }
    }
}
```

Lebih Besar

Lebih Kecil

Error

Tidak Ada Output

7

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        String y = "Hello";
        System.out.println(x * y);
    }
}
```

* y

String y

int * 5;

System.out.println

8

Temukan Bug di Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};
        System.out.println(arr[5]);
    }
}
```

arr[5]

int[] arr;

System.out.println

public static void main

9

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        int c = a _____ b;
        System.out.println(c);
    }
}
```

..

..

..

..

10

Lengkapi Kode Ini

```
public class Main {
    public static void main(String[] args) {
        try {
            int[] arr = {1, 2, 3};
            System.out.println(arr[3]);
        } catch (_____ e) {
            System.out.println("Array index out of bounds.");
        }
    }
}
```

IOException

ArrayIndexOutOfBoundsException

NullPointerException

Exception

11

Apa Outputnya?

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};
        System.out.println(arr[1] + arr[4]);
    }
}
```

30

50

60

70

12

Susun Kode Ini

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Masukkan sebuah angka: ");
        int x = scanner.nextInt();
    }
}
```

Scanner scanner = new Scanner(System.in);

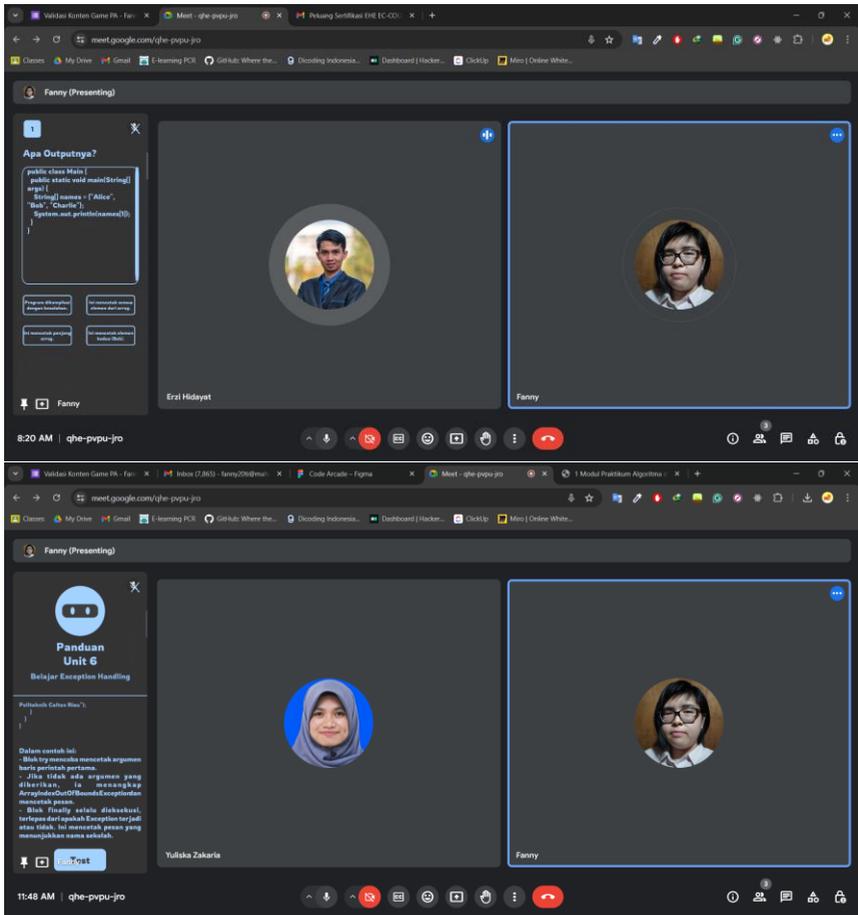
public class Main {

System.out.println("Masukkan sebuah angka: ");

int x = scanner.nextInt();

Submit

LAMPIRAN C – VALIDASI KONTEN



Who has responded?

Email

erzi@pcr.ac.id

yuliska@pcr.ac.id

Apakah section variable sudah sesuai?

2 responses

 Copy

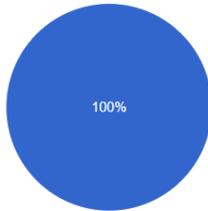


● Ya
● Tidak

Apakah materi di section variable sudah sesuai?

2 responses

 Copy



● Ya
● Tidak

Apakah soal tipe "apa outputnya" section variable sudah sesuai?

2 responses

 Copy

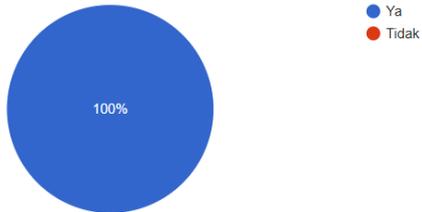


● Ya
● Tidak

Apakah soal tipe "lengkapi kode ini" section variable sudah sesuai?

 Copy

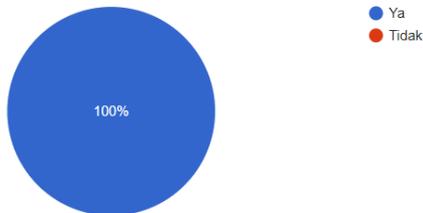
2 responses



Apakah soal tipe "temukan bug di kode ini" section variable sudah sesuai?

 Copy

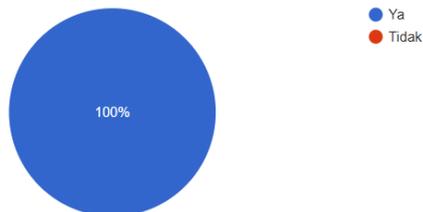
2 responses



Apakah soal tipe "susun kode ini" section variable sudah sesuai?

 Copy

2 responses



Apakah section operator sudah sesuai?

2 responses

 Copy



● Ya
● Tidak

Apakah materi di section operator sudah sesuai?

2 responses

 Copy

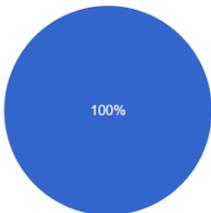


● Ya
● Tidak

Apakah soal tipe "lengkapi kode ini" section operator sudah sesuai?

2 responses

 Copy

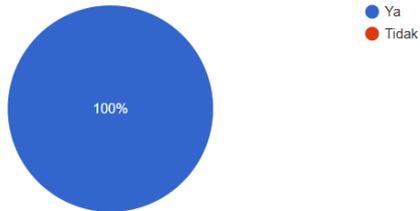


● Ya
● Tidak

Apakah soal tipe "apa outputnya" section operator sudah sesuai?

 Copy

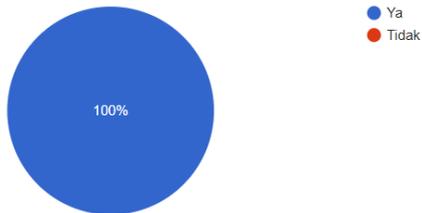
2 responses



Apakah soal tipe "temukan bug di kode ini" section operator sudah sesuai?

 Copy

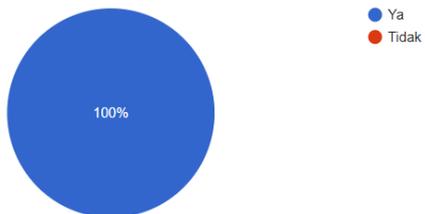
2 responses



Apakah soal tipe "susun kode ini" section operator sudah sesuai?

 Copy

2 responses



Apakah section input/output sudah sesuai?

2 responses

 Copy

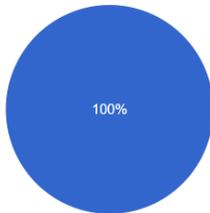


● Ya
● Tidak

Apakah materi di section input/output sudah sesuai?

2 responses

 Copy

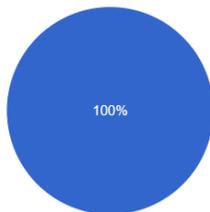


● Ya
● Tidak

Apakah soal tipe "lengkapi kode ini" section input/output sudah sesuai?

2 responses

 Copy

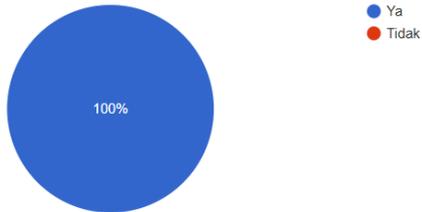


● Ya
● Tidak

Apakah soal tipe "temukan bug di kode ini" section input/output sudah sesuai?

 Copy

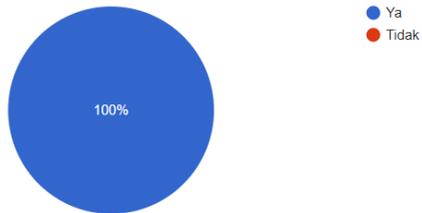
2 responses



Apakah soal tipe "susun kode ini" section input/output sudah sesuai?

 Copy

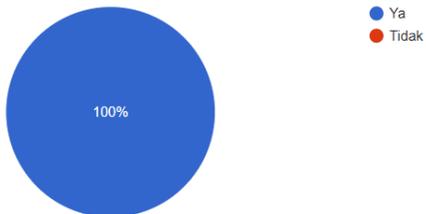
2 responses



Apakah soal tipe "apa outputnya" section input/output sudah sesuai?

 Copy

2 responses



Apakah section control statement sudah sesuai?

 Copy

2 responses



● Ya
● Tidak

Apakah materi di section control statement sudah sesuai?

 Copy

2 responses

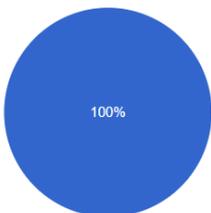


● Ya
● Tidak

Apakah soal tipe "apa outputnya" section control statement sudah sesuai?

 Copy

2 responses

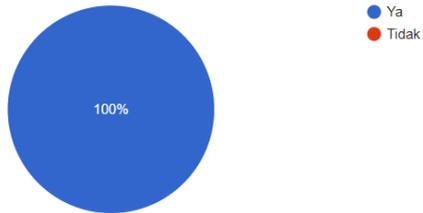


● Ya
● Tidak

Apakah soal tipe "temukan bug di kode ini" section control statement sudah sesuai?

 Copy

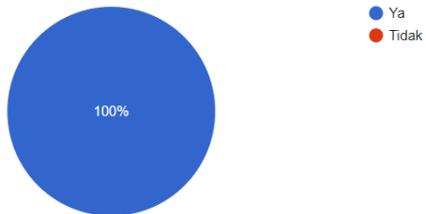
2 responses



Apakah soal tipe "susun kode ini" section control statement sudah sesuai?

 Copy

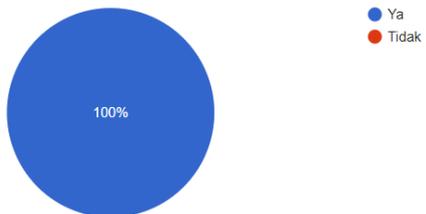
2 responses



Apakah soal tipe "lengkapi kode ini" section control statement sudah sesuai?

 Copy

2 responses



Apakah section array sudah sesuai?

 Copy

2 responses



● Ya
● Tidak

Apakah materi di section array sudah sesuai?

 Copy

2 responses

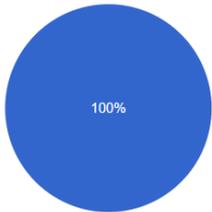


● Ya
● Tidak

Apakah soal tipe "apa outputnya" section array sudah sesuai?

 Copy

2 responses

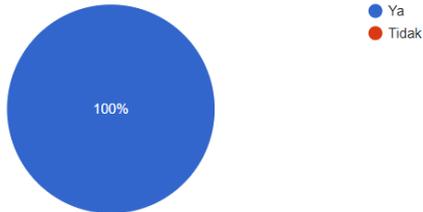


● Ya
● Tidak

Apakah soal tipe "temukan bug di kode ini" section array sudah sesuai?

 Copy

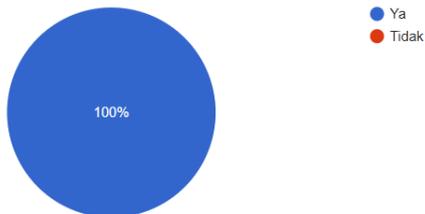
2 responses



Apakah soal tipe "susun kode ini" section array sudah sesuai?

 Copy

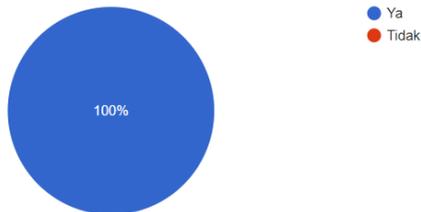
2 responses



Apakah soal tipe "lengkapi kode ini" section array sudah sesuai?

 Copy

2 responses



Apakah section exception handling sudah sesuai?

 Copy

2 responses



● Ya
● Tidak

Apakah materi di section exception handling sudah sesuai?

 Copy

2 responses

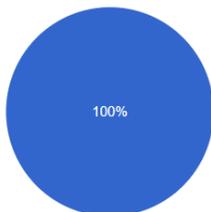


● Ya
● Tidak

Apakah soal tipe "temukan bug di kode ini" section exception handling sudah sesuai?

 Copy

2 responses

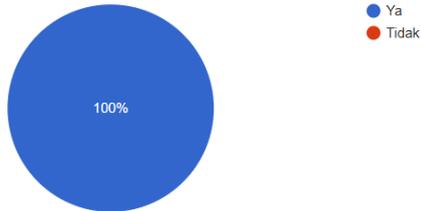


● Ya
● Tidak

Apakah soal tipe "susun kode ini" section exception handling sudah sesuai?

2 responses

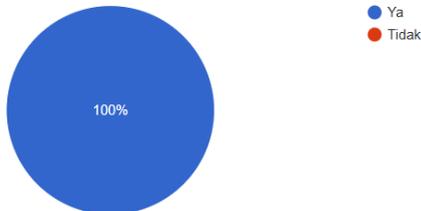
 Copy



Apakah soal tipe "apa outputnya" section exception handling sudah sesuai?

2 responses

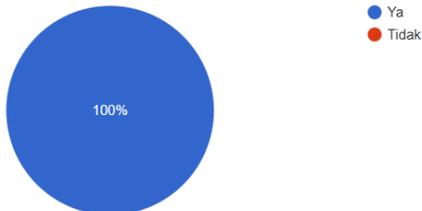
 Copy



Apakah soal tipe "lengkapi kode ini" section exception handling sudah sesuai?

2 responses

 Copy



LAMPIRAN D – PENGUJIAN UAT, RESPONDEN, KUESIONER KEPUASAN

Who has responded?

Email

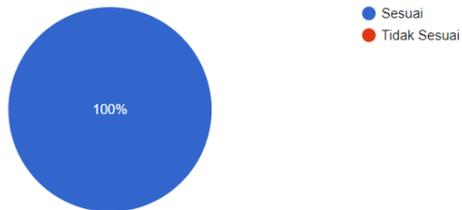
erzi@pcr.ac.id

Pengujian Halaman Awal

Game menampilkan halaman mode kesulitan ketika menekan tombol mulai ketika pemain baru pertama kali bermain

 Copy

1 response



Game menampilkan halaman menu module ketika menekan tombol mulai ketika pemain sudah memilih mode kesulitan

 Copy

1 response



Game menampilkan informasi tentang game ketika menekan tombol tentang

 Copy

1 response



● Sesuai
● Tidak Sesuai

Game berhenti berjalan ketika menekan tombol keluar

 Copy

1 response



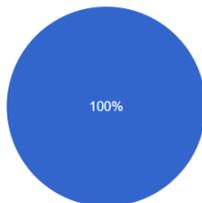
● Sesuai
● Tidak Sesuai

Pengujian Halaman Mode Kesulitan

Game menampilkan halaman module ketika memilih opsi "I'm New/Saya Pemula"

 Copy

1 response



● Sesuai
● Tidak Sesuai

Game menampilkan halaman pre-test ketika memilih opsi "I know basic/Saya tahu dasar" atau "I'm intermediate/Saya tingkat menengah"

 Copy

1 response



Pengujian Halaman Module

Game menampilkan halaman materi ketika menekan tombol "Learn/Pelajari"

 Copy

1 response



Game kembali ke halaman awal ketika menekan tombol kembali

 Copy

1 response



Pengujian Halaman Materi

Game menampilkan halaman test ketika menekan tombol test

 Copy

1 response



Game kembali ke halaman module ketika menekan tombol kembali

 Copy

1 response

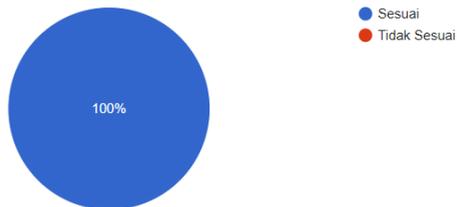


Pengujian Halaman Practice

Game menampilkan halaman test ketika menekan tombol test

 Copy

1 response



Game kembali ke halaman module ketika menekan tombol kembali

 Copy

1 response

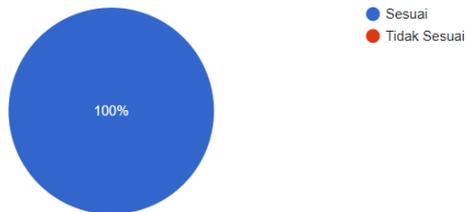


Pengujian Halaman Test/Pre-test

Game lanjut ke soal selanjutnya ketika menekan jawaban yang benar/jawaban yang salah

 Copy

1 response



Game kembali ke halaman materi ketika menekan tombol kembali

 Copy

1 response



Pengujian Halaman Result

Game menampilkan halaman pre-test/test ketika menekan tombol "retry/ulang"

 Copy

1 response



Game menampilkan halaman module ketika menekan tombol "continue/selanjutnya"

 Copy

1 response



Pengujian Halaman Tentang

Game kembali ke halaman awal ketika menekan tombol kembali

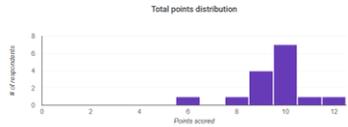
 Copy

1 response



Insights

Average	Median	Range
9.53 / 12 points	10 / 12 points	6 - 12 points



Frequently missed questions

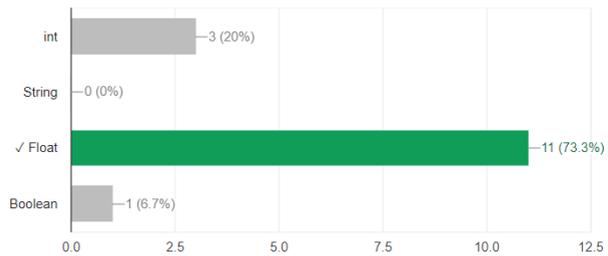
Pengujian Pre-test

Complete This Code

Copy

```
public class Main {  
    public static void main(String[] args) {  
        _____ y = 1.0;  
        System.out.println(y);  
    }  
}
```

11 / 15 correct responses

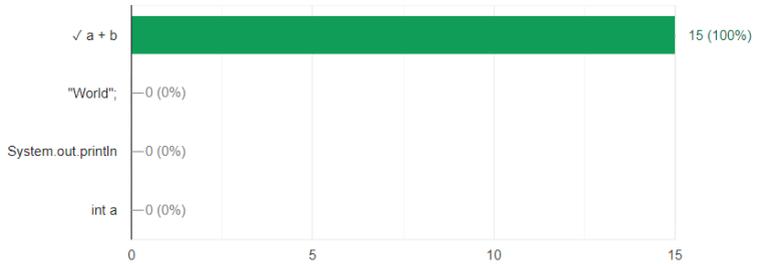


Find a bug in this code

 Copy

```
public class Main {  
    public static void main(String[] args) {  
        int a = 10;  
        String b = "World";  
        System.out.println(a + b);  
    }  
}
```

15 / 15 correct responses

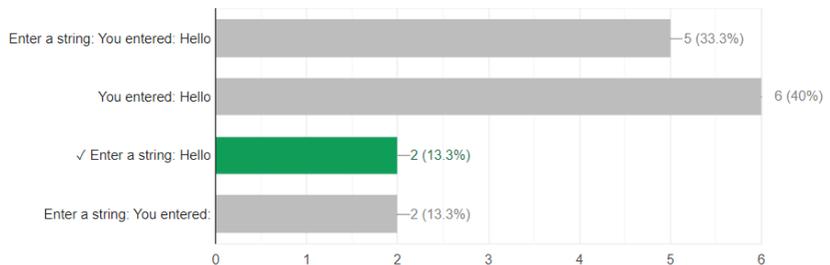


What is the output?

 Copy

```
import java.util.Scanner;  
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter a string: ");  
        String str = scanner.nextLine();  
        System.out.println("You entered: " + str);  
    }  
}
```

2 / 15 correct responses

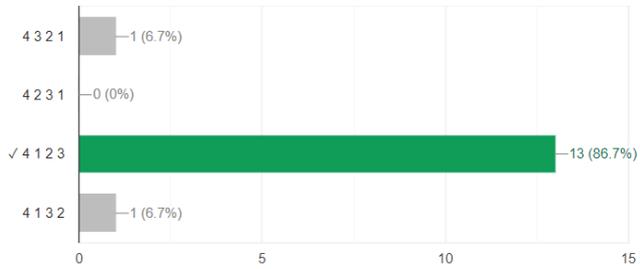


Arrange this code



1. public static void main(String[] args) {
2. String name = "John";
3. System.out.println("Hello, " + name);
4. public class Main

13 / 15 correct responses

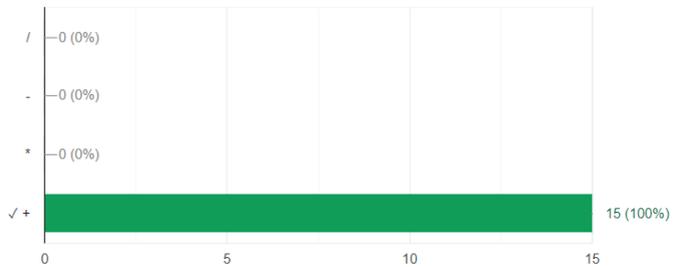


Complete This Code



```
public class Main {  
    public static void main(String[] args) {  
        int m = 15;  
        int n = 25;  
        int sum = m _____ n;  
        System.out.println(sum);  
    }  
}
```

15 / 15 correct responses

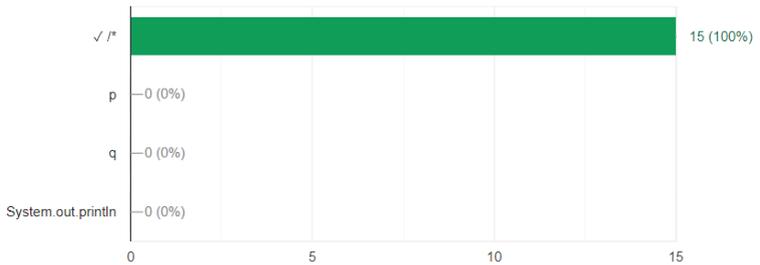


Find a bug in this code

[Copy](#)

```
public class Main {  
    public static void main(String[] args) {  
        int p = 20;  
        int q = 10;  
        int r = p /* q;  
        System.out.println(r);  
    }  
}
```

15 / 15 correct responses

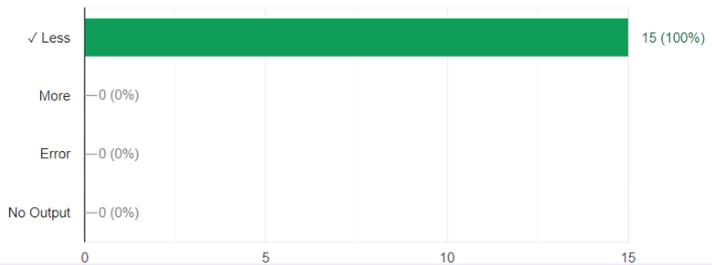


What is the output?

[Copy](#)

```
public class Main {  
    public static void main(String[] args) {  
        int y = 10;  
        if (y < 20) {  
            System.out.println("Less");  
        } else {  
            System.out.println("More");  
        }  
    }  
}
```

15 / 15 correct responses

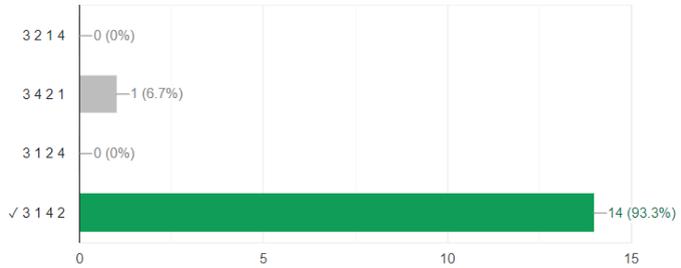


Arrange this code

 Copy

1. int number = 100;
2. System.out.println("Number is greater");
3. public static void main(String[] args) {
4. if (number > 50) {

14 / 15 correct responses

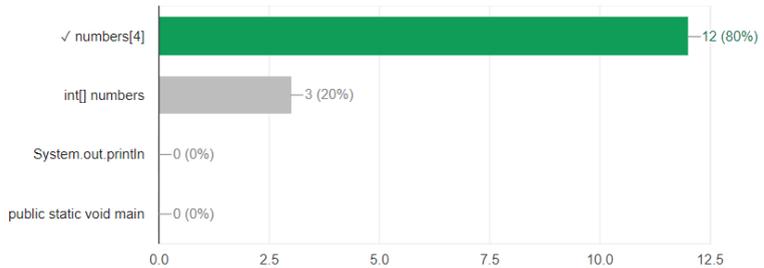


Find a bug in this code

 Copy

```
public class Main {  
    public static void main(String[] args) {  
        int[] numbers = {1, 2, 3, 4};  
        System.out.println(numbers[4]);  
    }  
}
```

12 / 15 correct responses

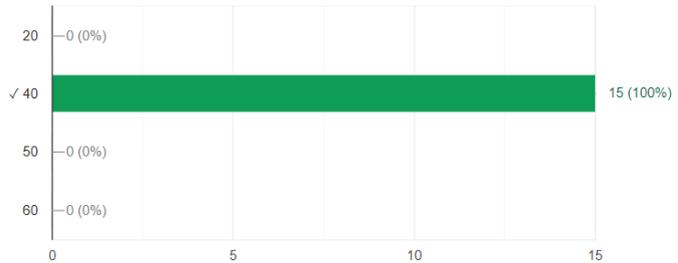


What is the output?

 Copy

```
public class Main {  
    public static void main(String[] args) {  
        int[] nums = {5, 15, 25, 35, 45};  
        System.out.println(nums[0] + nums[3]);  
    }  
}
```

15 / 15 correct responses

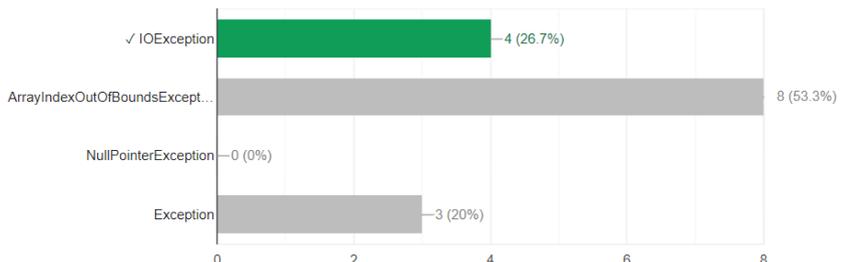


Complete This Code

 Copy

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            int[] array = {2, 4, 6};  
            System.out.println(array[5]);  
        } catch (_____ e) {  
            System.out.println("Index out of bounds");  
        }  
    }  
}
```

4 / 15 correct responses

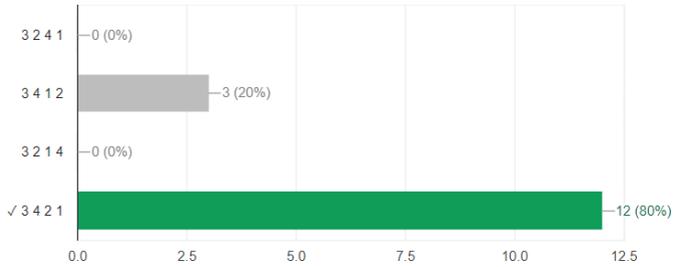


Arrange this code

 Copy

1. System.out.println("File not found");
2. } catch (FileNotFoundException e) {
3. try {
4. File file = new File("test.txt");

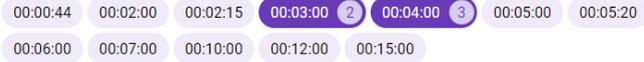
12 / 15 correct responses



Berapa lama kamu menyelesaikan pre-test?

15 responses

00 : :

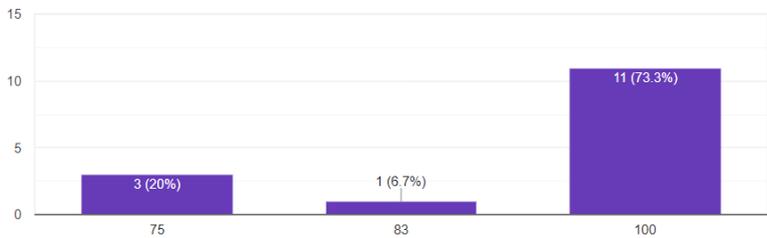


Pengujian Post-test

Akurasi (ex: 100)

 Copy

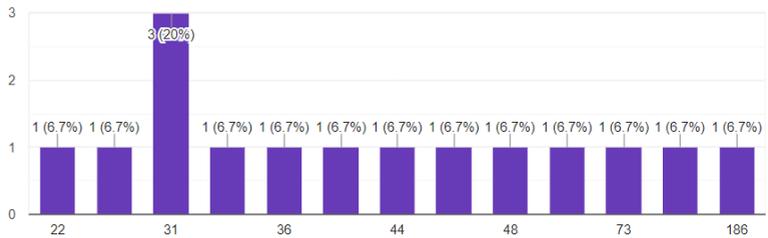
15 responses



Durasi (ex: 60)

 Copy

15 responses



Lampirkan Screenshot Hasil Practice

 View folder

15 responses

-  Screenshot_20240517_122040 - SYAFRISAR PUTRI CADASTERIA.jpg
-  WhatsApp Image 2024-05-17 at 12.42.49 PM - Chintya Ang.jpeg
-  sshasi - PINUS HENDRAWAN.jpg
-  Screenshot_2024-05-17-17-43-58-211_com.DefaultCompany.com.unity.template.mobile2D - Febriana.jpg
-  b3768f44-efd2-4a7d-aecc-3b58e260101d - AGNES FELICIA.jpeg
-  WhatsApp Image 2024-05-18 at 01.54.10 - Egy Dya Hermawan.jpeg
-  WhatsApp Image 2024-05-18 at 10.42.27_105ecbd3 - Oky Firnanda.jpg
-  WhatsApp Image 2024-05-18 at 13.54.48_4cd6e7f1 - Rasyid Tarmizi.jpg
-  Screenshot_20240518-205619 - Daniel John Tomsanit.jpg
-  Screenshot 2024-05-19 124726 - Bobby Hartanto.png

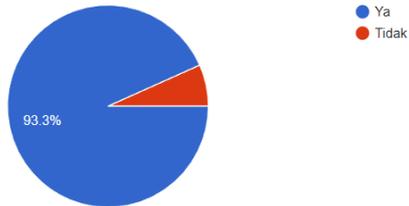
[5 more files](#)

Kuesioner Kepuasan Pengguna

Game sudah memiliki tampilan yang mudah dipahami

 Copy

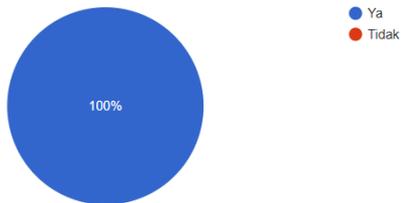
15 responses



Pembelajaran pemrograman lebih menarik melalui bermain game

 Copy

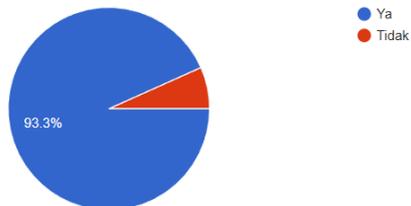
15 responses



Materi mudah dipahami dengan menggunakan game

 Copy

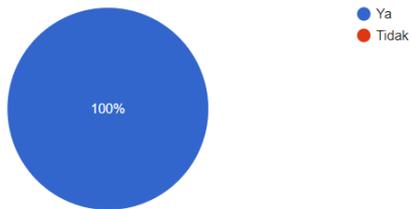
15 responses



Apakah kamu lebih suka belajar pemrograman melalui game?

 Copy

15 responses



Apakah kamu mau menggunakan game ini untuk belajar pemrograman dan membiasakan nya dirumah?

 Copy

15 responses

