

repository.ub.ac.id

**RANCANG BANGUN GAME MULTIPLAYER CERDAS CERMAT
MENGUNAKAN WEBSOCKET**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Yusuf Aji Wibowo
NIM: 115090613111004



INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016



PENGESAHAN

RANCANG BANGUN *GAME MULTIPLAYER CERDAS CERMAT* MENGGUNAKAN
WEBSOCKET

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Yusuf Aji Wibowo

NIM: 115090613111004

Skripsi ini telah diuji dan dinyatakan lulus pada
14 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Eriq Muh. Adams J, S.T, M.Kom

NIP: 198504102012121001

Drs. Marji, M.T

NIP: 196708011992031001

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T

NIP: 196708011992031001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 29 Desember 2015



Yusuf Aji Wibowo

NIM: 115090613111004

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT, karena hanya dengan rahmat dan bimbingannya penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun *Game Multiplayer* Cerdas Cermat Menggunakan Websocket” dengan baik. Tanpa rahmat dan bimbingan dari-Nya, maka niscaya penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik dan tepat waktu.

Terima kasih pula penulis sampaikan kepada pihak-pihak yang telah membantu penulis dalam penyelesaian tugas akhir ini. Pihak-pihak tersebut antara lain:

1. Orang tua penulis, Bapak Sungkono dan Ibu Suyati yang telah memberi motivasi dan kasih sayang kepada penulis. Mbak Wina yang telah memberikan semangat dari awal sampai akhir pengerjaan tugas akhir ini.
2. Bapak Drs. Marji, M.T. dan Issa Arwani, ST., MT. selaku Ketua dan Sekretaris Program Studi Informatika / Ilmu Komputer serta segenap Bapak/Ibu Dosen, Staf Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Eriq Muh. Adams J, S.T., M.Kom dan Drs. Marji, M.T. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu dan saran dalam penyusunan tugas akhir ini.
4. Bapak Yusi Tyroni Mursityo, S.Kom, MS. selaku dosen penasehat akademik yang telah memberikan bimbingan, ilmu dan saran selama penulis belajar.
5. Semua teman-teman seperjuangan mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya khususnya teman-teman Ilmu Komputer angkatan 2011.
6. Teman-teman kontrakan Yayan, Ady, Anjas, Alwi dan Rio yang selalu memberikan semangat dalam menyelesaikan tugas akhir ini.
7. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu yang membantu kelancaran penulisan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 29 Desember 2015

Yusuf Aji Wibowo

ucupper@gmail.com

ABSTRAK

Cerdas cermat merupakan salah satu permainan adu cepat dan tepat dalam menjawab suatu soal. Permainan cerdas cermat banyak diterapkan pada platform digital sebagai bentuk game edukasi. Selain sebagai media hiburan game edukasi bisa menjadi media pembelajaran yang menyenangkan. Adanya game cerdas cermat membuat pemain tidak lagi harus berada di satu ruangan lomba untuk dapat merasakan perlombaan cerdas cermat. Jika dilihat dari beberapa game cerdas cermat yang ada tersebut game dirasakan kurang menarik karena merupakan game single player hanya bisa dimainkan sendiri sehingga tidak bisa bermain dengan pemain lain dalam waktu yang sama (multiplayer). Untuk dapat membuat game multiplayer yang dapat berjalan secara real time dapat menggunakan teknologi websocket. Websocket adalah sebuah teknologi yang menyediakan dua arah komunikasi yang melalui sebuah socket Transmission Control Protocol pada sebuah web browser dan web server yang dapat digunakan oleh beberapa aplikasi client maupun server. Game ini dirancang menggunakan metode iterative and rapid prototyping. Dari hasil paper prototyping kemudian dilanjutkan dengan implementasi digital. Implementasi digital terbagi pada game server dan game client. Pada game client terdapat babak 1, 2, dan 3 dengan gameplay yang berbeda. Berdasarkan hasil pengujian unit menggunakan whitebox testing, didapatkan hasil perhitungan cyclomatic pada algoritma pencarian sebesar 3 dan algoritma penanda waktu sebesar 3. Dan hasil pengujian pada setiap jalur kasus uji hasil yang didapatkan yang sesuai dengan hasil yang diharapkan sebesar 100%. Berdasarkan hasil pengujian kombinatorial dan test flow diagram pada setiap jalur kasus uji didapatkan status valid sebesar 100% sehingga dapat disimpulkan bahwa fungsional game telah valid. Berdasarkan hasil kuesioner pengujian pengguna didapatkan hasil bahwa 70% koresponden menyatakan sangat baik dan 30% lainnya menyatakan baik pada komponen pertanyaan pertama. Sehingga didapatkan kesimpulan bahwa perancangan dan pengembangan game telah sesuai dengan tujuan yang diharapkan yaitu membangun game multiplayer cerdas cermat yang menyenangkan.

Kata kunci: *game*, cerdas cermat, *multiplayer*, websocket

ABSTRACT

Quiz is a game of fast and precise in answering a question. Quiz games are widely applied to digital platforms as a form of educational games. In addition as the entertainment media educational games can be a fun learning media. The existence quiz game makes players no longer have to be in one room can feel the race for the race quiz. If viewed from several game quiz that is the game felt less attractive because it is only a single player game can be played alone so could not play with other players in the same time (multiplayer). To be able to create multiplayer games that can be run in real time can use WebSocket technology. WebSocket is a technology that provides two-way communication through a socket Transmission Control Protocol in a web browser and web server that can be used by multiple client and server applications. This game is designed using iterative and rapid prototyping. From the results of paper prototyping is then followed by digital implementation. Digital implementation is divided on the game server and game client. In the game client there is a round of 1, 2, and 3 with different gameplay. Based on test results using whitebox testing unit, the result of calculation cyclomatic on search algorithms and Timber algorithms is 3. And the test results in each case track test results obtained in accordance with the expected results of 100%. Based on test results and test combinatorial flow diagrams on each track test cases obtained a valid status of 100% so it can be concluded that the functional game has been valid. Based on the test results of the questionnaire showed that 70% of correspondents expressed very good and another 30% stated either on the component first question. So it was concluded that the design and development of the game in accordance with the expected goals of building a multiplayer quiz game fun.

Keywords : game, quiz, multiplayer, websocket

DAFTAR ISI

RANCANG BANGUN <i>GAME MULTIPLAYER</i> CERDAS CERMAT MENGGUNAKAN WEBSOCKET	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Cerdas cermat	4
2.2 <i>Game design</i>	4
2.2.1 <i>Iterative and rapid prototyping</i>	4
2.2.2 <i>Formal element game</i>	5
2.3 Websocket	5
2.3.1 Socket.IO	6
2.4 Node.js	6
2.5 MongoDB	7
2.6 Teori pengujian	7
2.6.1 <i>White box testing</i>	7
2.6.2 <i>Black box testing</i>	9

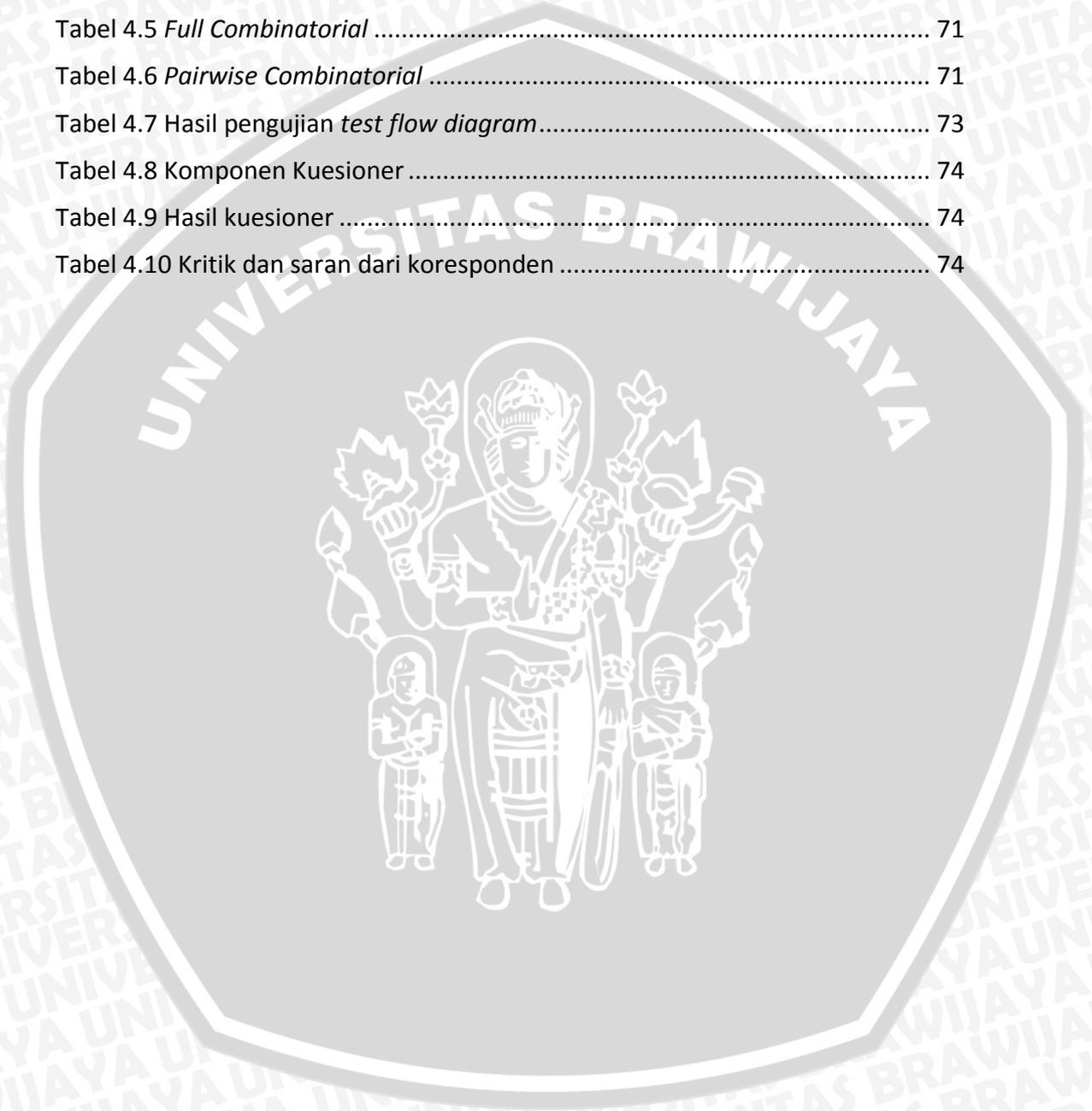


BAB 3 METODOLOGI, PERANCANGAN dan implementasi.....	11
3.1 Metodologi penelitian	11
3.1.1 Studi literatur	12
3.1.2 Perancangan <i>game</i>	12
3.1.3 Implementasi <i>game</i>	13
3.1.4 Pengujian <i>game</i>	13
3.2 Perancangan <i>game</i>	13
3.2.1 <i>Formal element game</i>	13
3.2.2 <i>Paper prototyping</i>	16
3.2.3 <i>Playtesting</i> dan evaluasi <i>paper prototyping</i>	19
3.2.4 Rancangan antarmuka	20
3.3 Implementasi <i>game</i>	25
3.3.1 Pemilihan teknologi dan platform	25
3.3.2 Implementasi <i>game server</i>	26
3.3.3 Implementasi <i>Game Client</i>	34
BAB 4 PEMBAHASAN.....	66
4.1 Pengujian unit	66
4.1.1 Pengujian unit pencarian	66
4.1.2 Pengujian Unit Penanda Waktu	68
4.2 Pengujian Kombinatorial	71
4.3 Pengujian <i>Test Flow Diagram</i> (TFD).....	72
4.4 Pengujian Terhadap Pengguna	73
BAB 5 PENUTUP	76
5.1 Kesimpulan.....	76
5.2 Saran	76
DAFTAR PUSTAKA.....	77
LAMPIRAN A KUISIONER TERHADAP PENGGUNA.....	78

DAFTAR TABEL

Tabel 3.1 Alat Bantu Implementasi.....	13
Tabel 3.2 <i>Game state</i> permainan cerdas cermat.....	15
Tabel 3.3 <i>Playtesting Paper Prototyping</i>	19
Tabel 3.4 Spesifikasi PC.....	25
Tabel 3.5 Spesifikasi Perangkat Android.....	25
Tabel 3.6 Perangkat lunak yang digunakan.....	26
Tabel 3.7 <i>Pseudocode</i> Fungsi Pencarian di <i>Server</i>	29
Tabel 3.8 <i>Pseudocode</i> Fungsi Siap Babak di <i>Server</i>	30
Tabel 3.9 <i>Pseudocode</i> Fungsi Jawab Babak di <i>Server</i>	31
Tabel 3.10 <i>Pseudocode</i> Fungsi <i>Game Selesai</i>	34
Tabel 3.11 <i>Pseudocode</i> Fungsi <i>Login</i>	35
Tabel 3.12 <i>Pseudocode</i> Fungsi <i>Challenge</i>	37
Tabel 3.13 <i>Pseudocode</i> Fungsi <i>Ready Wait</i>	37
Tabel 3.14 <i>Pseudocode</i> Fungsi <i>Pre Play</i>	38
Tabel 3.15 <i>Pseudocode</i> Fungsi <i>Go Babak 1</i>	41
Tabel 3.16 <i>Pseudocode</i> Fungsi <i>Fight Timer 1</i>	43
Tabel 3.17 <i>Pseudocode</i> Fungsi Jawab Babak 1.....	44
Tabel 3.18 <i>Pseudocode</i> Fungsi <i>Result Answered</i> Babak 1.....	45
Tabel 3.19 <i>Pseudocode</i> Fungsi Babak 1 <i>Done</i>	46
Tabel 3.20 <i>Pseudocode</i> Fungsi <i>Go Babak 2</i>	48
Tabel 3.21 <i>Pseudocode</i> Fungsi <i>Do Pilih</i>	50
Tabel 3.22 <i>Pseudocode</i> Fungsi Jawab Babak 2.....	51
Tabel 3.23 <i>Pseudocode</i> Fungsi <i>Result Answered</i> Babak 2.....	51
Tabel 3.24 <i>Pseudocode</i> Fungsi Babak 2 <i>Done</i>	53
Tabel 3.25 <i>Pseudocode</i> Fungsi <i>Go Babak 3</i>	55
Tabel 3.26 <i>Pseudocode</i> Fungsi <i>Open Question</i>	57
Tabel 3.26 <i>Pseudocode</i> Fungsi Jawab Babak 3.....	59
Tabel 3.27 <i>Pseudocode</i> Fungsi <i>Result Answered</i> Babak 3.....	60
Tabel 3.28 <i>Pseudocode</i> Fungsi Babak 3 <i>Done</i>	61
Tabel 3.29 <i>Pseudocode</i> Fungsi <i>Get History</i>	64

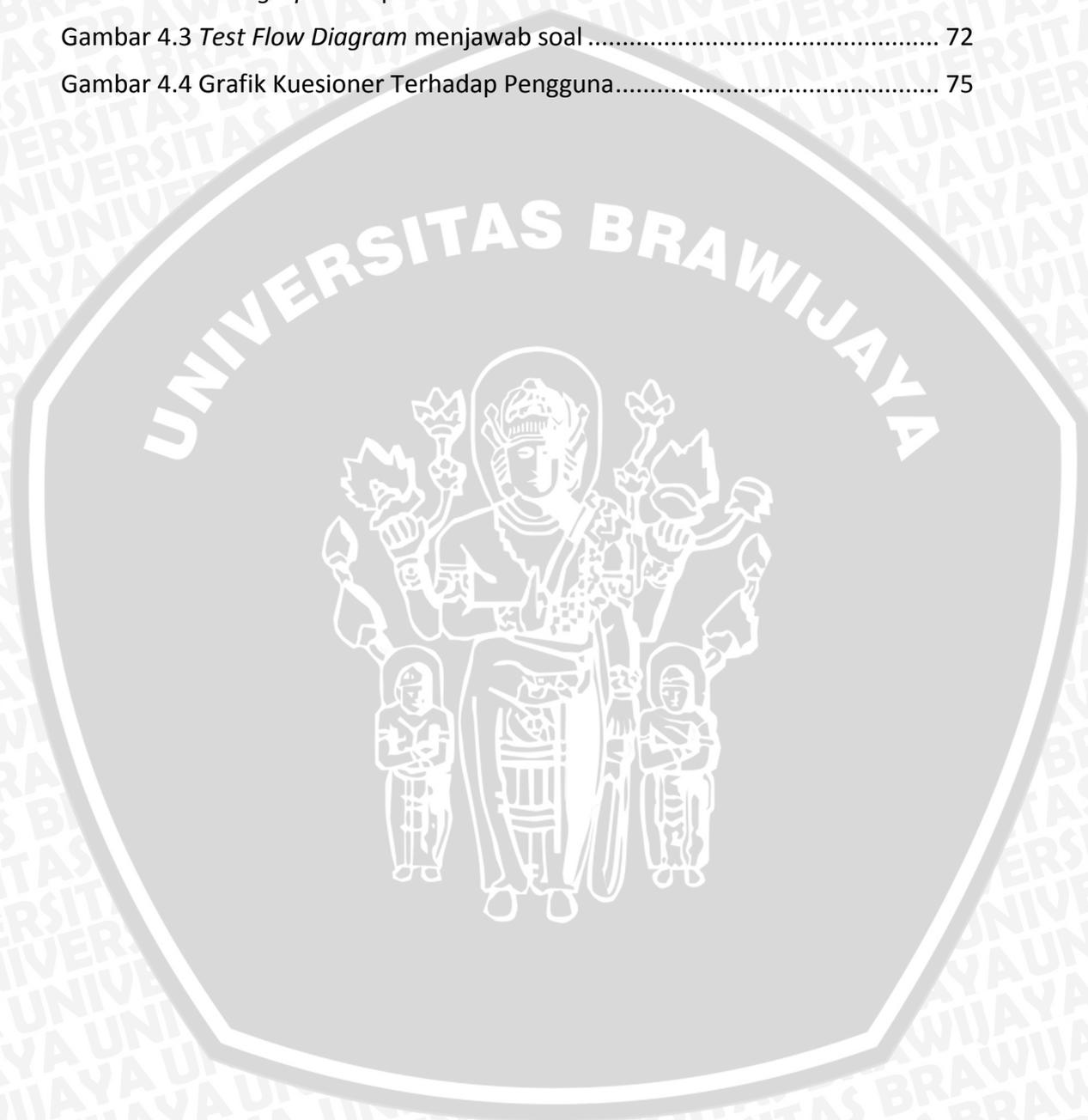
Tabel 4.1 Permodelan <i>flow graph</i> unit pencarian	66
Tabel 4.2 Tabel kasus uji unit pencarian	68
Tabel 4.3 Permodelan <i>flow graph</i> unit penanda waktu	68
Tabel 4.4 Tabel kasus uji unit penanda waktu	70
Tabel 4.5 <i>Full Combinatorial</i>	71
Tabel 4.6 <i>Pairwise Combinatorial</i>	71
Tabel 4.7 Hasil pengujian <i>test flow diagram</i>	73
Tabel 4.8 Komponen Kuesioner	74
Tabel 4.9 Hasil kuesioner	74
Tabel 4.10 Kritik dan saran dari koresponden	74



DAFTAR GAMBAR

Gambar 2.1 <i>Iterative rapid prototyping</i>	4
Tabel 2.1 Perbandingan antara <i>websocket</i> dan HTTP	6
Gambar 2.2 Logo Node.js	7
Gambar 2.3 Logo MongoDB	7
Gambar 2.4 <i>Flow graph</i>	8
Gambar 3.1 Diagram Alir Penelitian	11
Gambar 3.2 Diagram alir metode <i>iteratif and rapid prototyping</i>	12
Gambar 3.3 Papan Permainan	17
Gambar 3.4 Kartu Pemain Biru	17
Gambar 3.5 Kartu Pemain Merah	18
Gambar 3.6 Kartu Soal Tampak Depan	18
Gambar 3.7 Kartu Soal Tampak Belakang	18
Gambar 3.8 Screenflow rancangan antarmuka	21
Gambar 3.9 Rancangan Antarmuka <i>Login</i>	21
Gambar 3.10 Rancangan Antarmuka <i>Home Screen</i>	22
Gambar 3.11 Rancangan Antarmuka Babak 1	22
Gambar 3.12 Rancangan Antarmuka Babak 2	23
Gambar 3.13 Rancangan Antarmuka Babak 3	24
Gambar 3.14 Rancangan Antarmuka Riwayat	24
Gambar 3.15 Implementasi <i>Collection User</i>	27
Gambar 3.16 Implementasi <i>Collection Question</i>	28
Gambar 3.17 Implementasi <i>Collection Match</i>	28
Gambar 3.18 Hasil Antarmuka <i>Login</i>	36
Gambar 3.19 Hasil Antarmuka Halaman <i>Home</i>	40
Gambar 3.20 Antarmuka Pre Babak 1	40
Gambar 3.21 Hasil Antarmuka Babak 1	47
Gambar 3.22 Hasil Antarmuka Babak 1 Selesai	48
Gambar 3.23 Antarmuka Babak 2	54
Gambar 3.24 Antarmuka Babak 2 Selesai	55
Gambar 3.25 Antarmuka Babak 3	63

Gambar 3.26 Antarmuka Babak 3 Selesai	64
Gambar 3.27 Hasil Antarmuka Riwayat	65
Gambar 4.1 <i>Flow graph</i> unit pencarian	67
Gambar 4.2 <i>Flow graph</i> unit penanda waktu	70
Gambar 4.3 <i>Test Flow Diagram</i> menjawab soal	72
Gambar 4.4 Grafik Kuesioner Terhadap Pengguna.....	75



DAFTAR LAMPIRAN

LAMPIRAN A KUISIONER TERHADAP PENGGUNA..... 78



BAB 1 PENDAHULUAN

Bab ini akan membahas mengenai hal-hal yang akan dikerjakan beserta alasan penelitian ini. Komponen-komponen pada bab ini diantaranya latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

1.1 Latar belakang

Cerdas cermat merupakan salah satu permainan adu cepat dan tepat dalam menjawab suatu soal. Permainan cerdas cermat terdiri dari beberapa babak. Pada setiap babak para peserta diharuskan menjawab berbagai pertanyaan dengan cepat dan tepat. Pada babak yang pertama biasanya berupa pertanyaan yang standar yang kemudian pada babak berikutnya level permainan akan meningkat dengan skor yang lebih tinggi. Cerdas cermat sangat menarik karena dapat menghadirkan serunya menjawab berbagai pertanyaan, baik dari ilmu pengetahuan maupun pengetahuan umum.

Permainan cerdas cermat banyak diterapkan pada platform digital sebagai bentuk *game* edukasi. Selain sebagai media hiburan *game* edukasi bisa menjadi media pembelajaran yang menyenangkan. Pemakaian *game* sangat bermanfaat pada materi-materi yang berhubungan dengan matematika, fisika dan kemampuan berbahasa. Adanya *game* cerdas cermat membuat pemain tidak lagi harus berada di satu ruangan lomba untuk dapat merasakan perlombaan cerdas cermat. *Game* cerdas cermat banyak tersedia di *store* aplikasi seperti game Cerdas Cermat yang dibuat oleh SiswaMedia (SiswaMedia, 2015). Jika dilihat dari beberapa *game* cerdas cermat yang ada tersebut game dirasakan kurang menarik karena merupakan *game single user* hanya bisa dimainkan sendiri sehingga tidak bisa bermain dengan pengguna lain dalam waktu yang sama (*multiplayer*).

Penelitian yang sudah pernah dilakukan sebelumnya berhubungan dengan pembuatan game cerdas cermat adalah Pembuatan Aplikasi *Game* Cerdas Cermat berbasis Android (Perdana, 2013). Pada penelitian tersebut peneliti berhasil membuat *game* yang dikembangkan pada platform Android dengan bantuan Macromedia Flash 8.0. Game tersebut bisa dimainkan oleh 2 orang tetapi belum mendukung untuk dimainkan secara *real time*.

Untuk dapat membuat *game multiplayer* yang dapat berjalan secara *real time* dapat menggunakan teknologi websocket. Seperti penelitian Implementasi Teknologi Websocket dan HTML5 pada Game Crazy Typing dengan Genre Typing (Iryadi, 2013). Websocket adalah sebuah teknologi yang menyediakan dua arah komunikasi yang melalui sebuah socket *Transmission Control Protocol* (TCP) pada sebuah *web browser* dan *web server* yang dapat digunakan oleh beberapa aplikasi *client* maupun server (Chen, 2011).

Berdasarkan pada permasalahan di atas penulis tertarik untuk membuat penelitian dalam pengembangan *game* cerdas cermat yang dapat dimainkan secara *multiplayer*. Diharapkan dengan adanya *game* ini, pemain mendapatkan

sarana belajar yang menyenangkan. Dari penelitian ini penulis mengangkat judul “Rancang Bangun *Game Multiplayer* Cerdas Cermat Menggunakan Webscoket”.

1.2 Rumusan masalah

1. Bagaimana merancang dan mengimplementasikan *game multiplayer* cerdas cermat menggunakan websocket?
2. Bagaimana mengetahui fungsionalitas yang valid dan melakukan pengujian *game multiplayer* cerdas cermat?
3. Bagaimana mengetahui *game multiplayer* cerdas cermat yang menyenangkan?

1.3 Tujuan

Tujuan umum:

Mengembangkan aplikasi *game multiplayer* cerdas cermat menggunakan websocket.

Tujuan khusus:

1. Merancang dan mengimplementasi *game multiplayer* cerdas cermat menggunakan websocket.
2. Mengetahui fungsionalitas yang valid dan melakukan pengujian *game multiplayer* cerdas cermat.
3. Mengetahui *game multiplayer* cerdas cermat yang menyenangkan.

1.4 Manfaat

Manfaat penelitian ini antara lain:

1. Menjadi sarana belajar yang menyenangkan dan menantang bagi pemain.
2. Mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Informatika / Ilmu Komputer Universitas Brawijaya.
3. Mendapatkan pemahaman tentang perancangan dan pengembangan permainan menggunakan websocket

1.5 Batasan masalah

1. *Game* dijalankan di lingkungan *Local Area Network* (LAN).
2. Jumlah pemain yang dapat bermain dalam satu *room* adalah 2 pemain.

1.6 Sistematika pembahasan

Laporan penelitian ini disusun berdasarkan sistematika atau alur sebagai berikut:

BAB I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB II Landasan Kepustakaan

Menguraikan tentang kajian pustaka dan dasar teori yang mendasari perancangan, pengembangan, implementasi dan pengujian dari *game*.

BAB III Metodologi dan Perancangan

Membahas metodologi yang akan digunakan dalam penelitian yang terdiri dari studi pustaka, analisis kebutuhan, metode desain, metode implementasi, metode pengujian serta pengambilan kesimpulan dan saran dari penelitian “Rancang Bangun *Game Multiplayer Cerdas Cermat* menggunakan *Websocket*”. Memuat implementasi dari *game multiplayer* cerdas cermat yang berdasarkan metodologi dan rancangan yang telah dibuat

BAB IV Pembahasan

Memuat proses hasil pengujian dan analisis terhadap *game* yang telah diimplementasi.

BAB V Penutup

Pada bab ini berisi kesimpulan yang diambil berdasarkan pembuatan dan pengujian perangkat lunak yang dikembangkan serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menjelaskan landasan kepastakaan atau dasar teori yang didapat dari studi literatur dari berbagai sumber ilmiah.

2.1 Cerdas cermat

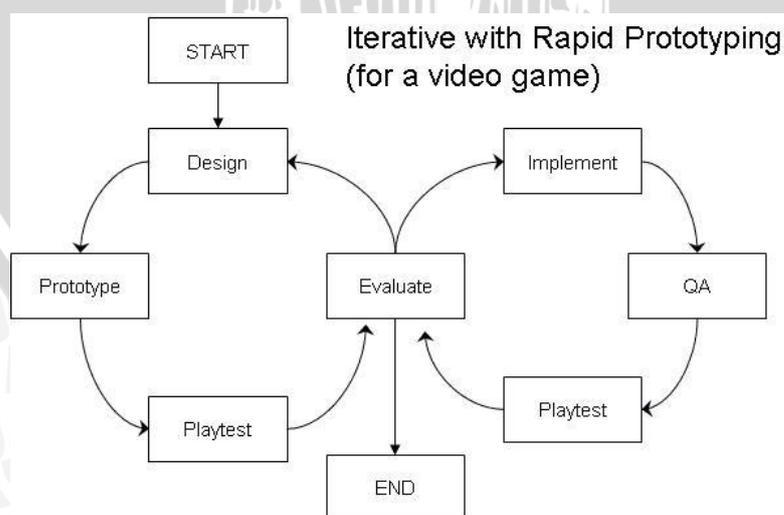
Cerdas cermat merupakan pertandingan adu ketajaman berpikir dan ketangkasan menjawab (pertanyaan-pertanyaan, soal-soal matematika, dan sebagainya) secara cepat dan tepat (Pusat Bahasa, 2008). Cerdas cermat bisa juga diartikan sebagai kuis yaitu bentuk permainan atau olahraga pikiran di mana para pemain (individu atau tim) mencoba untuk menjawab pertanyaan dengan benar. Kuis juga merupakan penilaian singkat yang digunakan dalam pendidikan dan bidang serupa mengukur peningkatan dalam pengetahuan, kemampuan, dan atau keterampilan (Annelgrim, 2015).

2.2 Game design

Game design atau perancangan *game* adalah proses pembuatan konten dan aturan permainan. Rancangan *game* yang baik merupakan proses membuat sebuah *game* yang membuat pemain merasa terdorong untuk mencapai tujuan permainan dengan mengikuti aturan permainan yang ada (Brathwaite, 2009).

2.2.1 Iterative and rapid prototyping

Iterative and rapid prototyping merupakan proses merancang *game* yang dimulai dari mencari ide game, membuat prototipe yang kemudian prototipe dibagi menjadi 2 yakni *paper prototype* dan *digital prototype*, mengimplementasi hasil prototipe dan mengevaluasinya. Proses-proses tersebut akan dilakukan berulang-ulang hingga mendapatkan hasil *game* yang diinginkan (Schreiber, 2009).



Gambar 2.1 Iterative rapid prototyping

2.2.2 Formal element game

Formal element game merupakan bagian-bagian yang berguna untuk untuk merancang suatu *game*. Bagian-bagian dari *game* dapat dipecah menjadi bagian kecil yang dapat dipelajari secara tersendiri yaitu diantaranya :

- *Players* : Banyaknya pemain yang dapat bermain dalam suatu tim atau secara individu.
- *Objectives* : Tujuan dari permainan dan hal yang harus dicoba lakukan oleh pemain, seperti *destroy* (menghancurkan), *territorial control* (menguasai wilayah), *collection* (mengumpulkan), dan lain-lain.
- *Rules* : Aturan dari *game*. Terdapat 3 kategori dari *rules* yaitu : *setup* (suatu hal yang harus dilakukan di awal permainan), *progression of play* (hal yang terjadi selama permainan berlangsung), dan *resolution* (suatu kondisi yang menyebabkan permainan berakhir, dan hasil akhir yang akan muncul).
- *Resource* : Segala sesuatu yang berada pada penguasaan seorang pemain. Serta bagaimana pemain dapat memanfaatkannya selama permainan berlangsung.
- *Information* : Ketersediaan status dari permainan untuk setiap pemain.
- *Sequencing* : Urutan alir permainan dari tindakan satu menuju tindakan yang lain.
- *Player Interaction* : Interaksi yang terjadi antar pemain dan pengaruhnya satu sama lain. Bentuk interaksi dapat berupa *direct conflict* (konflik secara langsung), *negotiation* (negosiasi), *trading* (jual beli), dan *information sharing* (berbagi informasi).
- *Theme* : Hubungan emosional pada permainan yang memudahkan untuk dipelajari dan dimainkan.
- *Game as Systems* : Hubungan antar elemen untuk memberikan pengalaman kepada pemain (Schreiber, 2009)..

2.3 Websocket

Socket adalah titik akhir dari lalu lintas komunikasi dua arah antara dua program yang berjalan pada jaringan komputer (Oracle, 2015). Websocket adalah sebuah teknologi yang menyediakan dua arah komunikasi yang melalui sebuah *socket* Transmission Control Protocol (TCP) pada sebuah *web browser* dan *web server* yang dapat digunakan oleh beberapa aplikasi *client* maupun server (Chen, 2011). Websocket merupakan fitur konektivitas baru dari HTML5. Websocket dapat melakukan perpindahan data 2 arah dalam suatu waktu atau yang dikenal dengan transmisi data *full-duplex*. Untuk menghasilkan sebuah koneksi dengan *protocol* websocket, *client* harus melakukan *request* dengan *protocol* HTTP/1.1 dan mendapat balasan dari server sehingga HTTP berubah menjadi *websocket* yang dikenal dengan *websocket opening handshake* (Wang, 2013).

Websocket memiliki efisiensi dalam transmisi data real-time dibanding protokol HTTP biasa, ini didukung oleh hasil percobaan yang dilakukan oleh Liu, Sun (2012), efisiensi merupakan faktor utama dalam transmisi data real-time. Melalui percobaannya dihasilkan data yang ditunjukkan pada **Tabel 2.1**. Dari tabel tersebut menunjukkan bahwa koneksi websocket 10 kali lebih efisien dibandingkan dengan koneksi HTTP (Iryadi, 2013).

Tabel 2.1 Perbandingan antara websocket dan HTTP

	Jumlah Paket		Jumlah Bit		Waktu (detik)	
	HTTP	Websocket	HTTP	Websocket	HTTP	Websocket
Klien ke server	83	5	33,662	372		
Server ke klien	77	8	45,600	7456		
Total	160	13	79,262	7828	~2.5	~0.25

Sumber: Iryadi (2013)

Websocket merupakan *protocol* yang sederhana, aplikasi *client* seperti *browser* biasa melalui *port* 80 atau 443 untuk mengirimkan *request* ke *server*. *Server* kemudian mengidentifikasi apakah koneksi yang terjalin merupakan *request websocket* dengan melihat dari *http header* yang diterima. Jika iya maka akan terjadi koneksi, setelah proses *handshake* berhasil, koneksi akan masuk pada fase transfer data dua arah antara *server* dan *client*.

Transfer data websocket didasarkan pada model *frame*: 0x00 yang menunjukkan bahwa awal dari transfer data, 0xff menunjukkan akhir data. Transfer data menggunakan pengodean utf-8. Jika dibandingkan HTTP *request* tradisional, proses *handshake* pada websocket menggunakan informasi *request header* yang sangat kecil antara *server* dan *client* yaitu 2 byte. Sehingga websocket dapat mengurangi jumlah permintaan *bandwidth* (Zhang, 2013).

2.3.1 Socket.IO

Socket.IO merupakan *Application Programming Interface* (API) dari websocket yang dibuat oleh Guillermo Rauch, CTO dari LearnBoost. Socket.IO ini akan membuat sambungan yang dibentuk dengan websocket untuk menciptakan aplikasi *real time* yang dapat bekerja di mana saja. Socket.IO juga menyediakan API untuk Node.js yang banyak ditemukan pada API sisi klien (Socket.IO, 2015).

2.4 Node.js

Node.js dikenalkan pada European JSConf di tahun 2009 oleh seorang *programmer* muda dengan nama Ryhan Dahl (Teixeira, 2013). Node merupakan sebuah platform yang dibangun menggunakan javascript *runtime* pada *browser* chrome. Node.js membuat pembuatan aplikasi menjadi lebih cepat dan dapat dikembangkan. Node.js menggunakan sebuah *event-driven*, model *non-blocking*

I/O sehingga menjadi ringan, efisien dan dapat mendukung aplikasi *real time* pada perangkat yang berbeda-beda (Cantelon, 2014).



Gambar 2.2 Logo Node.js

2.5 MongoDB

MongoDB yang berasal dari kata *humongous* merupakan sebuah sistem basis data lintas platform yang berorientasi pada dokumen. MongoDB adalah sebuah sistem basis data yang sangat berbeda dari sistem basis data lain yang lebih populer yaitu MySQL. Salah satu perbedaannya yaitu bahwa penulisan MySQL menggunakan *query* SQL sedangkan MongoDB menggunakan BSON (Binary JSON) sehingga banyak fungsi MongoDB dapat langsung diakses menggunakan notasi Javascript dan membuat integrasi data pada tipe tertentu pada aplikasi menjadi lebih mudah dan cepat.

MongoDB pertama dikembangkan oleh perusahaan perangkat lunak yang sekarang bernama MongoDB Inc pada bulan Oktober 2007 sebagai bagian dari rencana produk *platform as a service*. Sekarang MongoDB banyak digunakan sebagai *backend* perangkat lunak oleh beberapa situs dan layanan yang besar seperti Craigslist, eBay, Foursquare, SourceForge, Viacom dan New York Times. MongoDB merupakan sistem basis data NoSQL paling populer di dunia (DB-Engines, 2015).



Gambar 2.3 Logo MongoDB

2.6 Teori pengujian

2.6.1 White box testing

White box testing atau *glass-box testing* merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan

prosedural untuk memperoleh kasus uji (Pressman, 2010). Ada dua jenis pengujian yang termasuk *white-box testing* yaitu *basis path testing* dan *control structure testing*.

Pada penelitian ini menggunakan *basis path testing* yang diusulkan pertama kali oleh Tom McCabe (Pressman, 2010). *Basis path testing* ini memungkinkan perancang kasus uji memperoleh ukuran kompleksitas logis dari sebuah perancangan prosedural dan menggunakan pengukuran ini sebagai pedoman untuk mendefinisikan *basis set* dari jalur eksekusi (*execution path*). *Test case* yang dilakukan untuk menggunakan *basis set* tersebut dijamin untuk menggunakan setiap *statement* di dalam program paling tidak sekali selama pengujian. Sebelum metode *basis path* dapat diperkenalkan, notasi sederhana untuk representasi aliran kontrol yang disebut diagram alir (*flow graph*) harus diperkenalkan. Contoh *flow graph* dapat dilihat pada Gambar 2.2

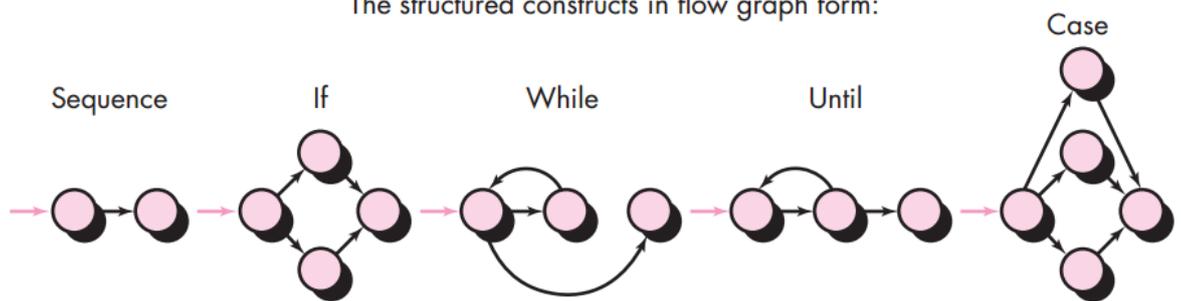
Cyclomatic complexity adalah metrik perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metrik ini digunakan dalam konteks metode pengujian *basis path*, maka nilai yang terhitung untuk *cyclomatic complexity* menentukan jumlah jalur independen (*independent path*) dalam *basis set* suatu program dan memberi batas atas bagi jumlah pengujian yang harus dilakukan untuk memastikan bahwa semua *statement* telah dieksekusi setidaknya satu kali.

Jalur independen adalah jalur yang melalui program yang mengenalkan sedikitnya satu rangkaian *statement* proses baru atau suatu kondisi baru.

- Jumlah region pada *flow graph* sesuai dengan *cyclomatic complexity*.
- *Cyclomatic complexity* $V(G)$, untuk grafik G adalah $V(G) = E - N + 2$, dimana E adalah jumlah *edge*, dan N adalah jumlah *node*.

$V(G) = P + 1$, dimana P adalah jumlah *predicate node* yaitu *node* yang merupakan kondisi (ada 2 atau lebih *edge* yang keluar dari *node* ini).

The structured constructs in flow graph form:



Where each circle represents one or more nonbranching PDL or source code statements

Gambar 2.4 Flow graph

Sumber : Pressman (2010)

2.6.2 Black box testing

Pengujian *black box* atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak (Pressman, 2010). Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut :

- Fungsi-fungsi yang tidak benar atau hilang.
- Kesalahan *interface*.
- Kesalahan dalam struktur data atau akses *database* eksternal.
- Kesalahan kinerja.
- Inisialisasi dan kesalahan terminasi.

2.6.2.1 Combinatorial testing

Combinatorial testing merupakan sebuah langkah pengujian untuk menemukan kesalahan dan mencapai kepercayaan dalam *game*. Kombinasi dalam pengujian ini menggunakan setiap *value* yang digunakan dalam pengujian akan dikombinasikan dengan setiap *value* dari parameter.

Parameter merupakan elemen-elemen dari *game* yang akan dimasukkan dalam pengujian kombinatorial. Untuk menemukan parameter dari *game* dapat dilakukan dengan melihat jenis dari elemen-elemen *game*, fungsi, dan pilihan seperti

- *Game events*
- Pengaturan *game*
- Pilihan *gameplay*
- Konfigurasi perangkat keras
- Atribut dari karakter
- Pilihan Kostumisasi

Value merupakan pilihan yang memungkinkan untuk setiap parameter. Values dapat berupa angka, teks, maupun dari sebuah daftar (Schultz, 2005).

2.6.2.2 Test flow diagram

Test Flow Diagram (TFD) merupakan bentuk grafis yang mewakili perilaku *game* dari sudut pandang pemain. Pengujian berlangsung dengan menjalankan *game* pada bentuk permainan yang biasa maupun yang tidak terduga. TFD menyediakan bentuk pendekatan untuk merancang pengujian yang

mempromosikan modularitas dan kelengkapan dari *game* yang akan diuji. Hal ini bertujuan agar mudah mudah ditinjau, dianalisis, dan mendapatkan umpan balik pada rancangan *game* yang akan diuji (Schultz, 2005).

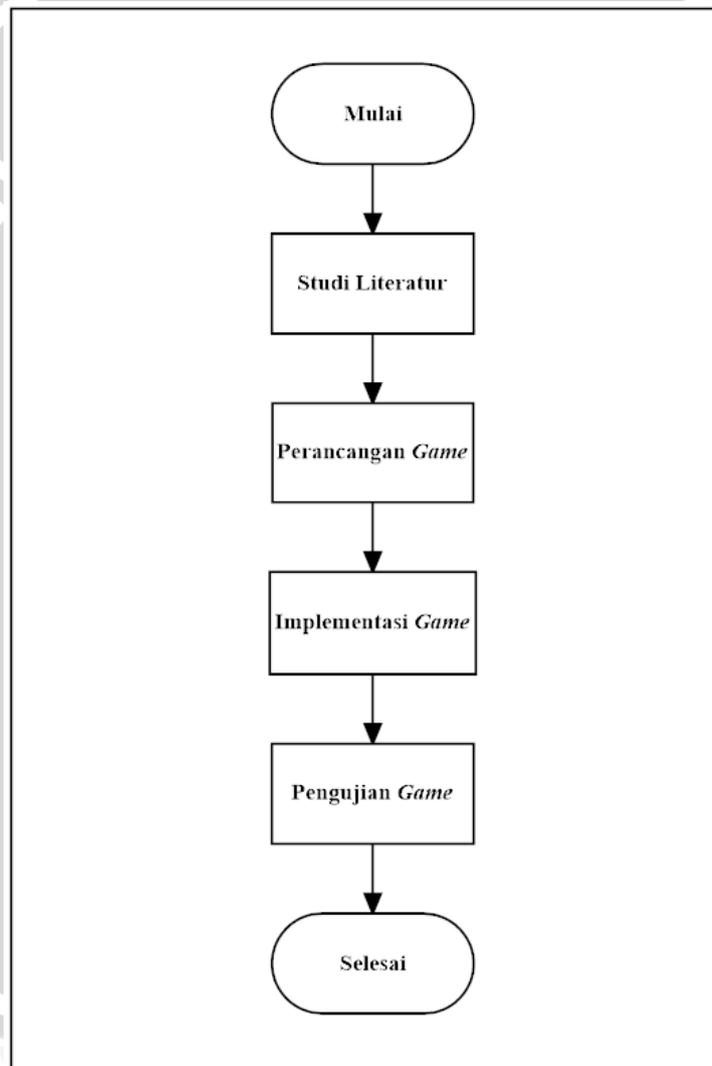


BAB 3 METODOLOGI, PERANCANGAN DAN IMPLEMENTASI

Bab ini akan menjelaskan metodologi penelitian, perancangan game beserta implementasinya.

3.1 Metodologi penelitian

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan penelitian ini, yaitu studi literatur, perancangan *game*, implementasi *game*, dan pengujian *game*. Berikut merupakan diagram alir dari pelaksanaan penelitian secara keseluruhan dapat dilihat pada **Gambar 3.1**.



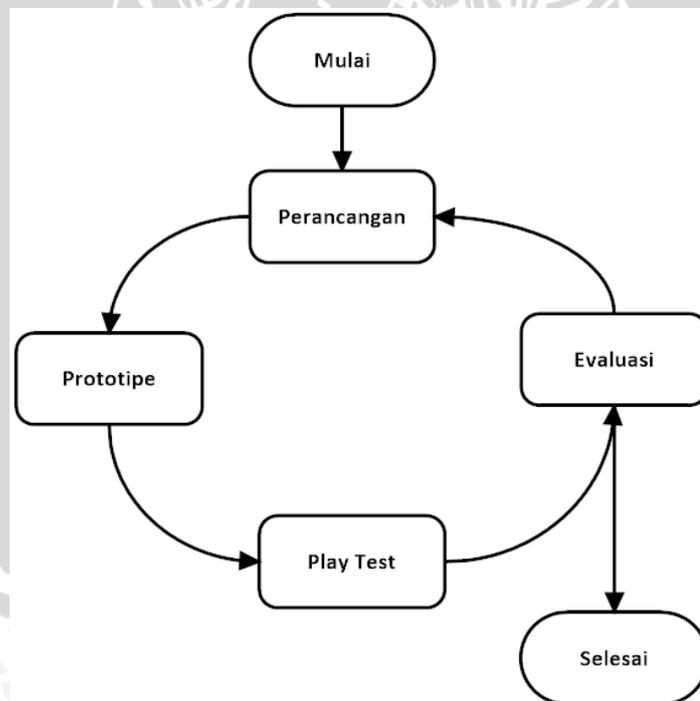
Gambar 3.1 Diagram Alir Penelitian

3.1.1 Studi literatur

Studi literatur yaitu mengumpulkan data dengan membaca dan mempelajari teori-teori dan literatur-literatur yang berkaitan dengan tema dari penelitian. Teori dan literatur tersebut didapat dari buku, jurnal, *e-book*, dan *website*. Dalam studi literatur akan dikumpulkan data-data yang valid dan dapat dipercaya sehingga aplikasi *game* yang dibuat dapat bermanfaat bagi para penggunanya. Studi literatur yang digunakan dalam penelitian ini adalah:

1. Cerdas Cermat
2. *Game Design*
 - *Iterative and Rapid Prototyping*
 - *Formal Element Game*
3. Websocket
4. Node.js
 - Socket.io
5. MongoDB
6. Teori Pengujian
 - White Box Testing
 - Combinatorial Testing
 - Test Flow Diagram

3.1.2 Perancangan *game*



Gambar 3.2 Diagram alir metode *iteratif and rapid prototyping*

Perancangan *game* pada penelitian ini menggunakan metode *Iterative and rapid prototyping* seperti yang ditunjukkan pada **Gambar 3.2**. Hal pertama yang dilakukan dalam metode ini adalah perancangan *game* dengan membuat bagian-

bagian penyusun dari game atau disebut dengan *formal element game*. *Formal element* terdiri dari *player*, *rule*, *objective*, dan beberapa lainnya. Kemudian hal selanjutnya yang dilakukan adalah membuat prototipe. Prototipe akan dibuat dalam bentuk kertas atau disebut dengan *paper prototyping*. Setelah pembuatan prototipe, hal selanjutnya yaitu melakukan *playtesting* secara iteratif atau berulang untuk mengevaluasi seberapa menyenangkan *game* yang akan dibuat. Kemudian hasil prototipe yang telah sesuai akan dilanjutkan pada tahap implementasi.

3.1.3 Implementasi game

Implementasi dalam pembuatan *game* ini merupakan penerapan dari perancangan *game* yang telah dibuat. Implementasi yang telah dilakukan di sini meliputi pemilihan teknologi dan platform dan implementasi dari prototipe yang telah dibuat. Untuk alat bantu yang akan digunakan dalam implementasi ditunjukkan pada **Tabel 3.1**.

Tabel 3.1 Alat Bantu Implementasi

No	Tugas	Nama Perangkat Lunak
1	Perancangan <i>art</i> dan aset <i>game</i>	CorelDRAW X5
2	Bahasa pemrograman dan aplikasi	JetBrains WebStorm 10.0
3	Implementasi <i>gameplay</i> , level dan <i>screen</i> .	JetBrains WebStorm 10.0

3.1.4 Pengujian game

Pengujian pada *game* ini dimaksudkan untuk menunjukkan apakah *game* dapat berjalan sesuai dengan yang diharapkan atautkah *game* masih terdapat *bug* yang perlu diperbaiki. Evaluasi dibagi ke dalam 3 bagian yaitu pengujian *white box*, *combinatorial testing*, dan *test flow diagram*.

3.2 Perancangan game

Ada beberapa tahap dalam perancangan *game* yaitu pembuatan formal elemen *game*, pembuatan *paper prototyping*, *playtesting* dan evaluasi *paper prototyping*, kemudian perancangan antarmuka *game* digital.

3.2.1 Formal element game

Formal element game penyusun *game* cerdas cermat ini ada beberapa bagian diantaranya *player*, *objectives*, *rule*, *game State*, *information*, *sequencing*, *player interaction* dan *game system*.

3.2.1.1 Player

Player yang dimaksud adalah jumlah pemain yang terdapat pada *game* cerdas cermat ini. Pada *game* ini terdapat 2 pemain yang bisa bermain. Pemain akan

bermain dengan pemain lainnya secara langsung. Jika pemain meninggalkan sesi permainan, maka pemain akan dinyatakan kalah. Interaksi yang terjadi antar pemain ini disebut *multiplayer* (1 *player* vs 1 *player*).

3.2.1.2 Objectives

Objectives yang dimaksud adalah apa yang akan pemain lakukan pada *game* cerdas cermat ini. Pada *game* ini pemain akan memilih soal untuk lawan dan juga harus menjawab soal yang diberikan secara tepat.

3.2.1.3 Rule

Rule merupakan aturan-aturan yang ada pada sebuah permainan. *Rule* dibagi menjadi 3 bagian yaitu *setup* yang dilakukan pada awal permainan dan hanya dilakukan sekali, *progression of play* yaitu apa yang terjadi selama permainan berlangsung, dan *resolution* yaitu kondisi menang atau kalah. Dari 3 bagian tersebut maka *game* cerdas cermat ini memiliki *rule*:

- *Setup* : memilih untuk memulai permainan, dan menunggu lawan
- *Progression of Play* : pada babak 1 masing-masing pemain akan diberikan soal secara bergantian, jika pemain menjawab dengan benar akan mendapatkan poin 10, jika menjawab salah maupun tidak menjawab maka akan mendapatkan poin 0. pada babak 2 pemain *game server* akan memberikan 10 soal, pemain harus memilih 1 pertanyaan untuk dijawab oleh lawan, dan ini dilakukan secara bergantian dan berulang sebanyak 5 kali untuk masing-masing pemain. Jika pemain tidak memilihkan pertanyaan untuk lawan maka lawan akan mendapatkan poin bonus sebanyak 20. jika pemain berhasil menjawab soal dengan benar akan mendapatkan poin 20 dan akan mendapatkan poin 0 jika salah maupun tidak menjawab. Pada babak 3 *game server* akan memberikan 16 kotak soal, masing-masing pemain harus memilih salah satu kotak dan menjawab pertanyaan yang muncul. Kotak pertanyaan yang sudah terjawab tidak dapat dipilih lagi. Jika pemain berhasil menjawab pertanyaan yang muncul dengan benar maka akan mendapatkan poin 30 dan akan mendapatkan poin 10 jika salah maupun tidak menjawab.
- *Resolution* : Sesi permainan akan berakhir setelah babak 3 selesai. pemain dinyatakan menang jika poin yang didapat lebih besar dari lawan dan pemain dinyatakan kalah jika poin yang didapat lebih kecil dari lawan dan keadaan seri jika poin yang didapat dari kedua pemain bernilai sama

3.2.1.4 Game state

Game state merupakan kumpulan *state* yang ada pada *game*. Pada *game* cerdas cermat ini terdapat beberapa *state* yang dimulai dari *state login* dan diakhiri dengan *state* pengumpulan poin akhir. Untuk lebih jelasnya akan ditunjukkan pada **Tabel 3.2**.

Tabel 3.2 *Game state* permainan cerdas cermat

State	Keterangan
<i>State login</i>	<i>State</i> ini merupakan proses pemain <i>login</i> ke <i>server</i>
<i>State</i> memulai bermain	<i>State</i> ini merupakan pemain memulai permainan dan menunggu lawan dari <i>server</i>
<i>State</i> menjawab soal babak 1	Di <i>state</i> ini pemain menjawab soal yang diberikan
<i>State</i> pengecekan benar atau salah babak 1	Di <i>state</i> akan menampilkan jawaban babak 1 dari pemain apakah benar atau salah.
<i>State</i> memilih soal babak 2 untuk lawan	Di <i>state</i> ini pemain memilih salah satu soal yang akan dilempar yang kemudian akan dijawab oleh lawan
<i>State</i> menjawab soal babak 2	Di <i>state</i> ini pemain akan menjawab soal babak 2 yang dilempar oleh lawan.
<i>State</i> pengecekan benar atau salah babak 2	Di <i>state</i> ini akan menampilkan jawaban babak 2 dari pemain apakah benar atau salah.
<i>State</i> memilih kotak soal babak 3	Di <i>state</i> ini pemain akan memilih salah satu kotak soal yang belum terjawab
<i>State</i> menjawab soal babak 3	Di <i>state</i> ini pemain akan menjawab soal babak 3 yang telah dipilih.
<i>State</i> pengecekan benar atau salah babak 3	Di <i>state</i> akan menampilkan jawaban babak 3 dari pemain apakah benar atau salah.
<i>State</i> pengumpulan poin akhir	Di <i>state</i> ini akan menampilkan poin akhir dari masing-masing pemain.

3.2.1.5 Information

Information merupakan informasi-informasi yang diberikan oleh *game system* kepada pemain. Pada *game* cerdas cermat ini informasi yang diberikan oleh *game system* kepada pemain berupa instruksi-instruksi seperti informasi mengenai penjelasan permainan pada tiap babak serta informasi giliran menjawab oleh masing-masing pemain

3.2.1.6 Sequencing

Pada *game* cerdas cermat ini model *sequencing* yang digunakan adalah *turn based* yaitu secara bergiliran. Pada *game* ini pemain akan menjawab maupun memilih soal secara bergantian

3.2.1.7 Player interaction

Interaksi pemain dalam *game* ini menggunakan model *direct conflict*. Dengan mode ini pemain akan berhadapan langsung dengan pemain yang lain dalam menjawab soal secara langsung

3.2.1.8 Game system

Pada *game* cerdas cermat ini jalannya permainan akan diatur oleh *game server*. Ketika satu pemain memilih untuk memulai permainan, maka *server* akan mencarikan lawan yang juga memilih untuk memulai permainan. Soal-soal yang diberikan juga akan diberikan oleh *server* kepada masing-masing pemain.

3.2.2 Paper prototyping

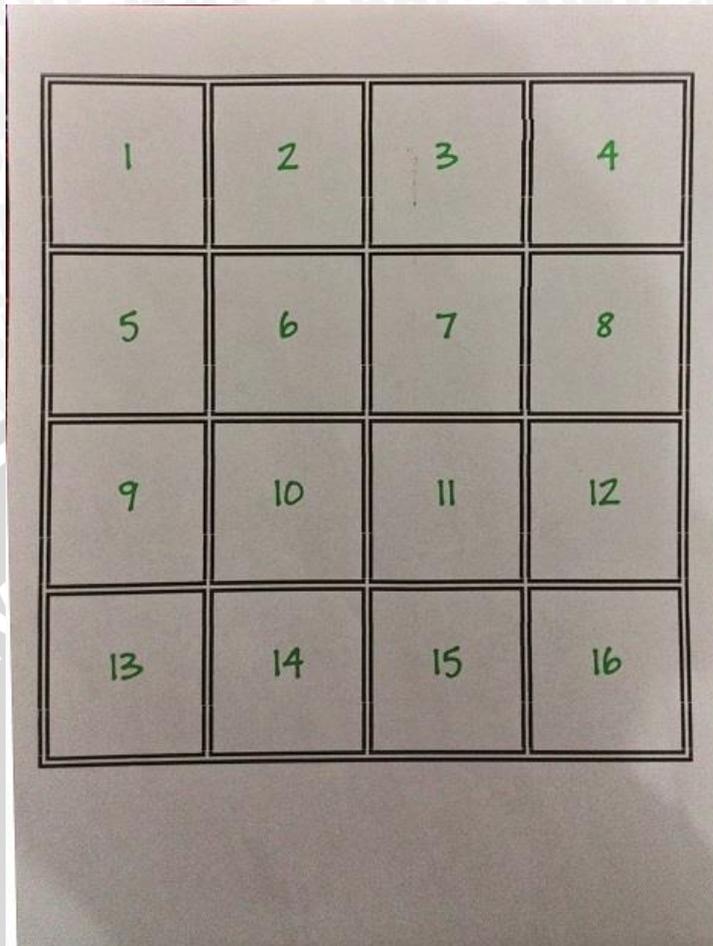
Paper prototyping merupakan pembuatan prototipe *game* dalam bentuk kertas. *Paper Prototyping* terdiri dari seperangkat perlengkapan *game* yang kemudian dilakukan *playtesting* dan evaluasi secara berulang.

3.2.2.1 Perlengkapan game

Perlengkapan *game* pada *paper prototyping* berisi komponen permainan yang terdiri dari *board game*, kartu pemain dan kartu soal.

- **Board game (Papan Permainan)**

Terdapat 16 kotak yang setiap kotak berisi nomor masing-masing. Kotak-kotak ini merupakan tempat untuk meletakkan kotak soal dan kotak pemain. Gambaran papan permainan ditunjukkan pada **Gambar 3.3**.



Gambar 3.3 Papan Permainan

- **Kartu Pemain**

Kartu pemain merupakan terdapat 2 jenis kartu yaitu kartu biru seperti yang ditunjukkan pada **Gambar 3.4** dan kartu merah yang ditunjukkan pada **Gambar 3.5**. Setiap jenis kartu untuk satu pemain. Kartu pemain ini digunakan untuk menandai dengan cara meletakkan pada kotak di papan permainan jika pemain berhasil menjawab kartu soal.



Gambar 3.4 Kartu Pemain Biru



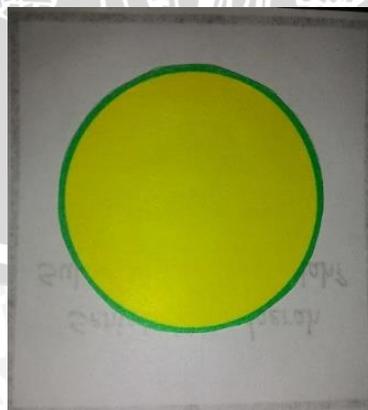
Gambar 3.5 Kartu Pemain Merah

- **Kartu Soal**

Kartu soal merupakan kartu yang berisi suatu pertanyaan beserta jawabannya. Kartu soal ini akan diletakkan pada tiap-tiap kotak di papan permainan yang kemudian akan diambil oleh pemain dan kemudian menjawabnya. Gambaran kartu soal seperti yang ditunjukkan pada **Gambar 3.6** tampak depan dan **Gambar 3.7** tampak belakang.



Gambar 3.6 Kartu Soal Tampak Depan



Gambar 3.7 Kartu Soal Tampak Belakang

3.2.2.2 Rule

- Masing-masing pemain memilih kartu biru atau merah. Kemudian meletakkan kartu masing-masing yang berjumlah 8 di samping papan permainan. Untuk menentukan giliran pertama yang mengambil kartu soal dilakukan pelemparan koin. Masing-masing pemain memilih gambar atau angka.
- Pemain yang kalah dari pelemparan koin harus mengacak kartu soal kemudian meletakkan untuk tiap satu kotak pada papan permainan dengan kartu soal. Sisa dari kartu soal kemudian diletakkan di samping papan permainan.
- Kemudian pemain dengan giliran pertama memilih salah satu kartu soal. Kemudian pemain yang lainnya mengambil dan membacakan soal yang tertera di kartu soal kepadanya. Pemain harus menjawab dengan tepat jawaban dari soal yang dibacakan. Jika pemain berhasil menjawab dengan benar maka pemain bisa meletakkan kartunya (merah atau biru) di kotak papan permainan dengan nomor yang sesuai dengan kartu soal yang diambilnya. Jika pemain gagal menjawab maka pemain yang lainnya memberi tahu jawabannya. Kemudian pemain mengambil satu kartu soal yang ada di samping papan permainan kemudian diletakkan di papan permainan untuk menggantikan kartu soal yang gagal dijawab.
- Permainan dinyatakan berakhir jika salah satu dari salah satu pemain maupun keduanya telah habis. Pemenang dari permainan adalah dihitung dari banyaknya jumlah kartu pemain yang terhubung secara horizontal maupun vertikal pada papan permainan. Pemain yang menang akan mendapat giliran pertama untuk permainan selanjutnya dan pemain yang kalah harus mengacak kartu soal untuk permainan selanjutnya.

3.2.3 Playtesting dan evaluasi *paper prototyping*

Playtesting dilakukan secara berulang-ulang untuk mengevaluasi *paper prototyping* hingga dapat menghasilkan rancangan *game* yang menyenangkan ketika dimainkan. Daftar *playtesting* akan ditunjukkan pada **Tabel 3.3**.

Tabel 3.3 Playtesting Paper Prototyping

Iterasi	Feedback	Analisis
1	Bentuk permainan terlalu monoton dengan hanya pergantian kartu soal	Membuat papan permainan dengan 25 kotak soal.
2	<i>Game</i> sudah menarik tetapi dengan jumlah kotak soal berjumlah 25 terlalu lama untuk jumlah pemain 2 orang.	Mengurangi kotak soal menjadi 16

3	Lebih baik kartu soal yang tidak terjawab, jawabannya dibacakan dan diganti dengan kartu soal yang baru.	Memperbanyak jumlah kartu soal dalam satu sesi permainan sehingga terdapat sisa kartu soal yang dapat digunakan untuk mengganti kartu soal yang tidak terjawab.
4	Pemenang ditentukan oleh banyaknya kartu soal yang terhubung.	Menerapkan dengan menentukan pemenang dari jumlah kartu soal yang terhubung.
5	Dibalik kartu soal diberi tanda agar terlihat ketika diletakkan pada papan permainan.	Membuat tanda lingkaran berwarna kuning pada kartu soal.

3.2.4 Rancangan antarmuka

Screenflow rancangan antarmuka seperti yang ditunjukkan pada **Gambar 3.8** digunakan untuk memudahkan dalam menjelaskan alur proses antarmuka *game* cerdas cermat *multiplayer*. Antarmuka *game* dimulai dari halaman *login* kemudian menuju halaman *home*, dari halaman *home* bisa menuju halaman riwayat atau menuju babak 1, dari babak 1 akan berlanjut menuju babak 2 yang berlanjut ke babak 3. Dari babak 3 yang merupakan akhir permainan, akan kembali ke halaman *home*.

3.2.4.1 Rancangan Antarmuka *Login*

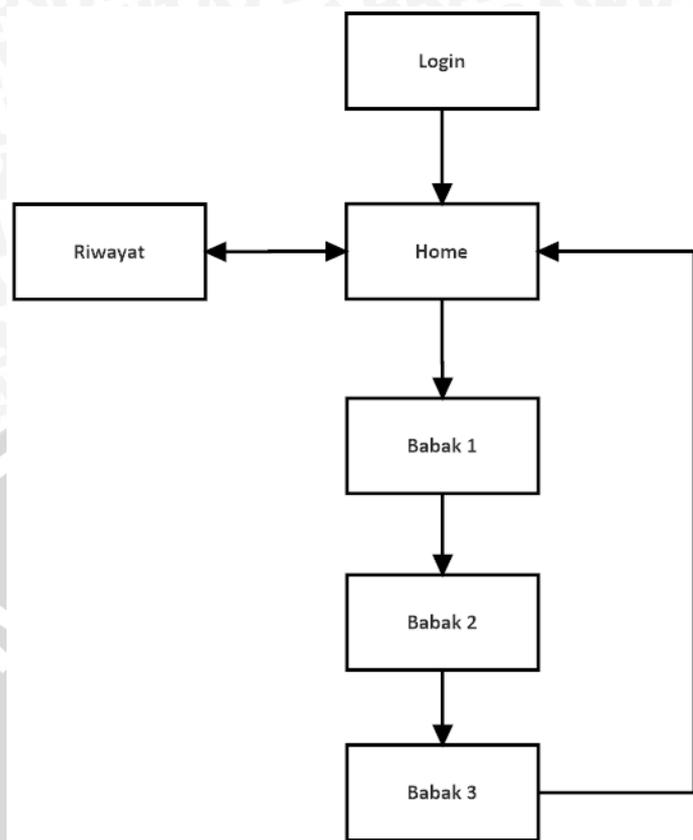
Rancangan antarmuka *login* ditunjukkan pada **Gambar 3.9**. Tampilan *login* akan berisi kotak input untuk *username* dan *password* serta terdapat satu tombol *login*.

3.2.4.2 Rancangan Antarmuka Halaman *Home*

Rancangan antarmuka halaman *home* ditunjukkan pada **Gambar 3.10**. Pada tampilan *home screen* terdapat informasi *username*, serta jumlah poin yang dimiliki, dan juga terdapat tombol *Main* untuk memulai permainan dan tombol *Logout* untuk keluar dari akun yang sedang digunakan.

3.2.4.3 Antarmuka Babak 1

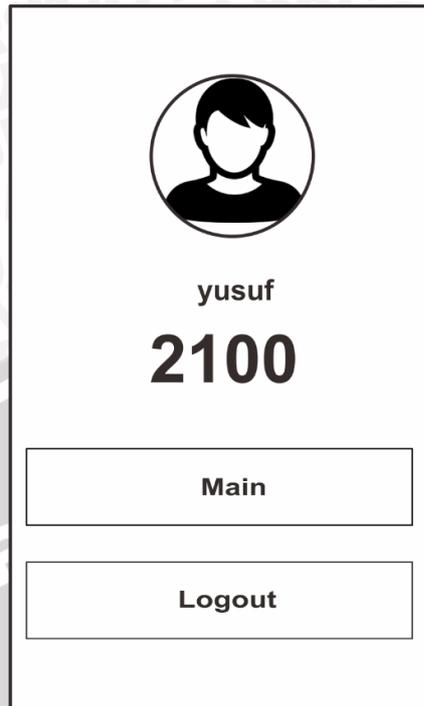
Rancangan antarmuka babak 1 ditunjukkan pada **Gambar 3.11**. Antarmuka babak 1 akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal dan 4 pilihan jawaban.



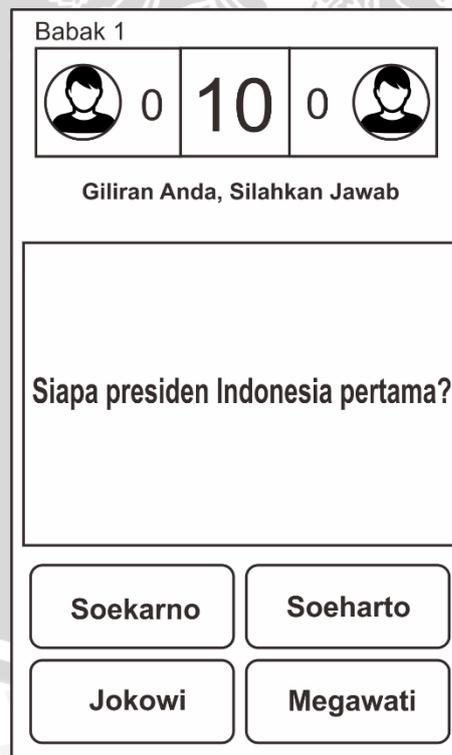
Gambar 3.8 Screenflow rancangan antarmuka

The login form consists of three main components arranged vertically within a rectangular frame. At the top is the label 'username' above a horizontal input field. Below this is the label 'password' above another horizontal input field. At the bottom is a rectangular button labeled 'Login'.

Gambar 3.9 Rancangan Antarmuka Login



Gambar 3.10 Rancangan Antarmuka Home Screen



Gambar 3.11 Rancangan Antarmuka Babak 1

3.2.4.4 Antarmuka Babak 2

Berikut ini adalah rancangan antarmuka babak 2 yang ditunjukkan pada **Gambar 3.12**. Antarmuka babak 2 akan dibuat dua tampilan. Pada antarmuka yang pertama akan menampilkan beberapa pilihan soal untuk dipilih pemain. Kemudian pada antarmuka kedua akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal yang telah dipilih sebelumnya dan 4 pilihan jawaban.



Gambar 3.12 Rancangan Antarmuka Babak 2

3.2.4.5 Antarmuka Babak 3

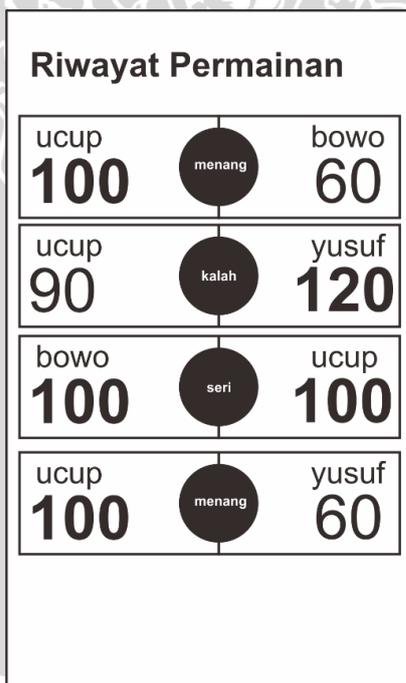
Rancangan antarmuka babak 3 ditunjukkan pada **Gambar 3.13**. Antarmuka babak 3 akan dibuat dua jenis tampilan. Pada antarmuka yang pertama akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, serta 16 kotak yang berisi soal-soal yang berbeda. Untuk antarmuka yang kedua akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal yang telah dipilih sebelumnya dan 4 pilihan jawaban.

3.2.4.6 Antarmuka Riwayat

Rancangan antarmuka riwayat ditunjukkan pada **Gambar 3.14**. Antarmuka riwayat permainan akan menampilkan daftar riwayat permainan dari pemain yang bersangkutan. Dalam daftar akan menampilkan detail *username* dari kedua pemain dan poin yang didapat.



Gambar 3.13 Rancangan Antarmuka Babak 3



Gambar 3.14 Rancangan Antarmuka Riwayat

3.3 Implementasi *game*

Implementasi *game* akan dibagi ke dalam beberapa bagian di antaranya adalah pemilihan teknologi dan platform, implementasi *game server*, dan implementasi *game client*.

3.3.1 Pemilihan teknologi dan platform

Pemilihan teknologi dan platform akan dibagi ke dalam dua bagian yaitu pemilihan teknologi perangkat keras dan pemilihan teknologi perangkat lunak.

3.3.1.1 Pemilihan teknologi perangkat Keras

Teknologi perangkat keras untuk membuat *game* ini adalah sebuah *personal computer* dan perangkat android sebagai alat pengujian *game*.

1. *Personal computer* (PC)

PC digunakan sebagai alat pembuatan *game* dan sebagai *game server*. Spesifikasi PC yang dipakai akan dijelaskan pada **Tabel 3.4**.

Tabel 3.4 Spesifikasi PC

PC	
<i>Processor</i>	Intel® Core™ i3 2330M Processor
<i>Memory</i> (RAM)	2 x SO-DIMM socket for expansion up to 8 GB SDRAM, DDR3 1333 MHz SDRAM
<i>Hardisk</i>	500GB 5400
<i>Graphic Card</i>	NVIDIA® GeForce® GT 520M dengan 1GB DDR3 VRAM
<i>Monitor</i>	14.0" 16:9 HD (1366x768) LED Backlight

2. Perangkat android

Perangkat android digunakan sebagai alat pengujian *game*. Spesifikasi perangkat android akan dijelaskan pada **Tabel 3.5**.

Tabel 3.5 Spesifikasi Perangkat Android

Perangkat Android	
<i>Processor</i>	Dual-core 1 GHz Cortex-A9
<i>Memory</i> (RAM)	8 GB (5 GB <i>user available</i>), 512 MB RAM
<i>Screen Display</i>	LED-backlist LCD, capacitive touchscreen, 16 M colors

3.3.1.2 Pemilihan Teknologi Perangkat Lunak

Dalam pengembangan *game* ini dibutuhkan berbagai macam perangkat lunak pendukung dalam pembuatan *asset* dari *game*. Pada **Tabel 3.6** akan dijelaskan berbagai macam perangkat lunak yang digunakan.

Tabel 3.6 Perangkat lunak yang digunakan

No	Perangkat Lunak	Keterangan
1	Ubuntu 15.04	Sebagai sistem operasi dari PC yang digunakan dalam pengembangan <i>game</i> dan sebagai <i>game server</i> .
2	Windows 8.1	Sebagai sistem operasi dari PC yang digunakan dalam pembuatan <i>game asset</i> .
3	Android 5.0 Lollipop	Sebagai sistem operasi dari perangkat android yang digunakan untuk pengujian <i>game</i> .
4	JetBrains WebStorm 10.0	Sebagai aplikasi pemrograman (HTML, CSS, Javascript) dan pembuatan <i>game</i> .
5	Adobe Illustrator CS6	Sebagai aplikasi pembuat dan pengolah <i>game asset</i> .
6	NodeJS versi 0.10.35	Sebagai mesin <i>server</i> dari <i>game</i> yang dibuat.
7	MongoDB versi 2.9.6	Sebagai aplikasi penyimpanan dan pengolah data <i>game</i> .

3.3.2 Implementasi *game server*

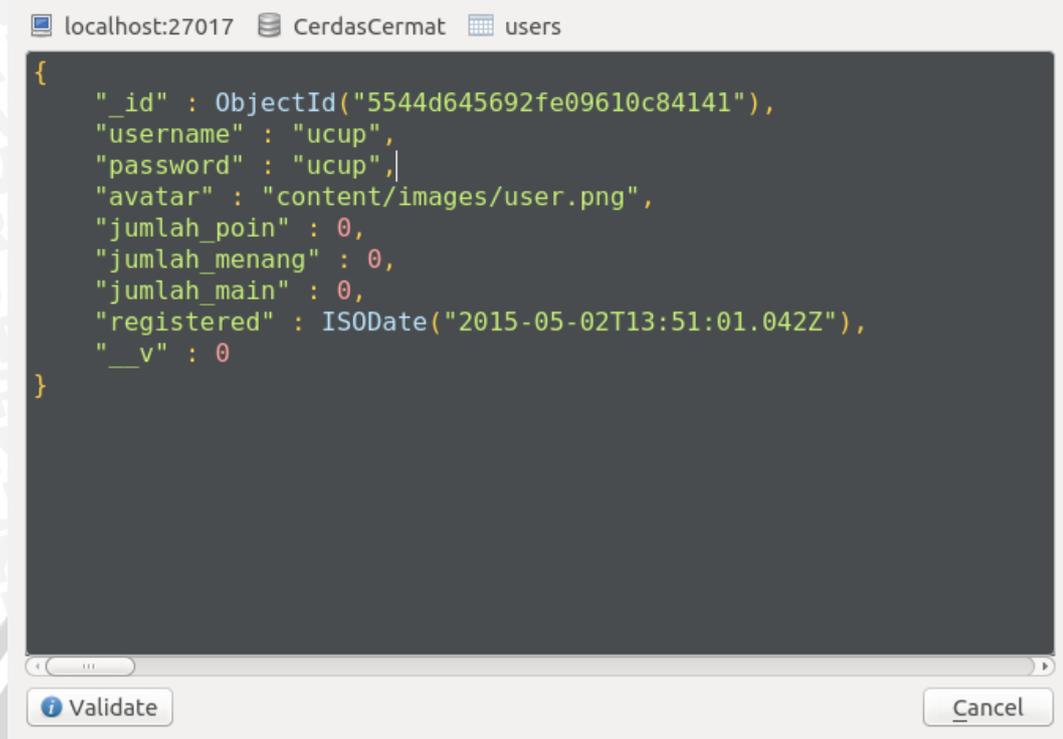
Implementasi *game server* terbagi menjadi dua bagian yaitu implementasi basis data dan implementasi program *game server*.

3.3.2.1 Implementasi basis data

Basis data yang digunakan dalam pembuatan *game* ini adalah MongoDB versi 2.9.6. MongoDB berguna untuk menyimpan dan mengolah data *game* yang berisi data pemain dan data soal. Untuk membantu dalam pengolahan data tersebut digunakan aplikasi basis data klien yaitu Robomongo.

- **Implementasi *collection user***

Collection user merupakan kumpulan data pemain yang tersimpan dalam satu dokumen. *Collection user* berisi beberapa komponen seperti *username*, *password*, *avatar*, *jumlah_poin*, *jumlah_menang*, *jumlah_main*, dan *tanggal_daftar*. Berikut ini adalah bentuk satu data *Collection user* yang ditunjukkan pada **Gambar 3.15**.



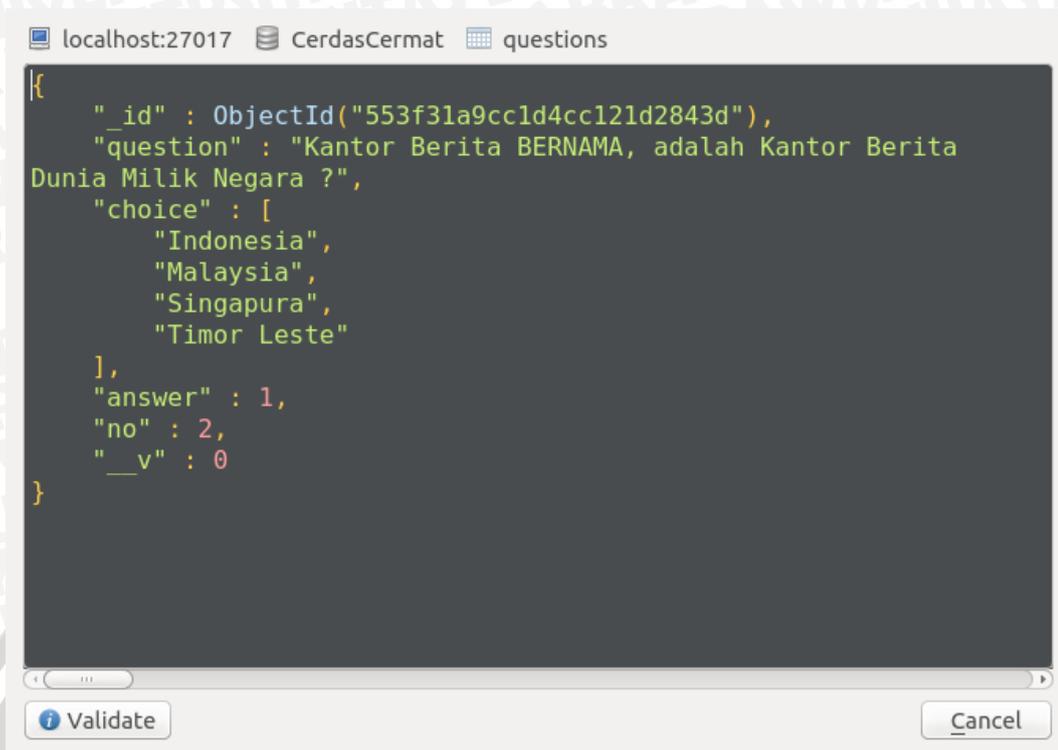
Gambar 3.15 Implementasi *Collection User*

- **Implementasi *Collection Question***

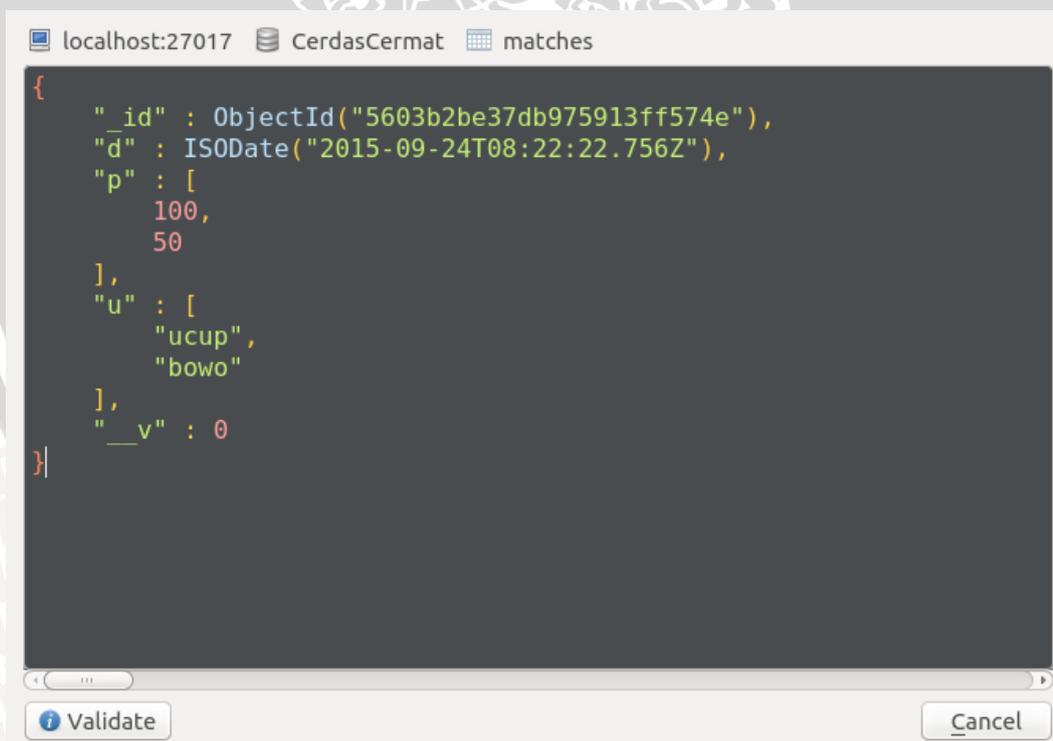
Collection Question merupakan kumpulan data pemain yang tersimpan dalam satu dokumen. *Collection Question* berisi beberapa komponen seperti *question* yang berisi data pertanyaan, *choice* yang berisi data pilihan jawaban, *answer* yang berisi data jawaban, serta data nomor. Berikut ini adalah bentuk *Collection Question* yang ditunjukkan pada **Gambar 3.16**.

- **Implementasi *Collection Match***

Collection Match merupakan kumpulan data permainan yang tersimpan dalam satu dokumen. *Collection Question* berisi beberapa komponen seperti tanggal, *username* dan poin masing-masing. Berikut ini adalah bentuk *Collection Question* yang ditunjukkan pada **Gambar 3.17**.



Gambar 3.16 Implementasi *Collection Question*



Gambar 3.17 Implementasi *Collection Match*

3.3.2.2 Implementasi Program *Game Server*

Program *game server* dijalankan pada NodeJS. NodeJS akan menjalankan kode-kode program yang ditulis dalam javascript. Ada beberapa jenis fungsi yang

diimplementasikan pada program *game server* yaitu fungsi pencarian, fungsi siap babak, fungsi jawab babak, dan fungsi *game* selesai.

- **Implementasi Websocket**

Tahap Implementasi websocket merupakan tahap penambahan library socket.IO pada kode program game server. socket.IO akan menyediakan API yang digunakan dalam pembuatan fungsi-fungsi websocket.

- **Implementasi Fungsi Pencarian**

Fungsi pencarian digunakan untuk mempertemukan dua pemain dalam satu ruangan (*room*) permainan. Untuk membedakan digunakan tipe untuk setiap pemain yaitu X dan Y. Berikut ini adalah *pseudocode* fungsi pencarian yang ditunjukkan pada **Tabel 3.7**.

Tabel 3.7 Pseudocode Fungsi Pencarian di Server

Baris	Pseudocode
1	DECLARATION
2	pemainX
3	pemainY
4	socket
5	
6	DESCRIPTION
7	IF socket.username = undefined THEN
8	CALL socket.emit with 'logout'
9	ELSE
10	IF pemainX > pemainY THEN
11	ADD socket to pemainY
12	SET opponent to pemainX.shift
13	SET socket.musuh to opponent
14	SET socket.poin to 0
15	SET opponent.musuh to socket
16	SET socket.musuh.poin to 0
17	SET name room to opponent
18	CALL socket.join with name_room
19	SET socket.tipe to Y
20	pemainY.shift
21	SET pertanyaan to GET FROM database
22	ELSE
23	ADD socket to pemainX
24	SET socket.tipe to X
25	CALL socket.join with socket.username

26	SET socket.name_room to socket.username
27	ENDIF
28	ENDIF

Penjelasan *pseudocode* dari fungsi pencarian pada **Tabel 3.7** adalah

- Variabel yang digunakan dalam fungsi ini yaitu fungsi pemainX, pemainY dan socket. Kedua variabel pemainX dan pemainY tersebut digunakan untuk menampung pemain-pemain yang melakukan pencarian lawan. Variabel socket merupakan hubungan jaringan yang terjalin dengan *game* klien. Variabel socket berisi informasi terkait detail *game* klien dari pemain.
 - Baris 7-8 digunakan untuk pengondisian jika *username* dari pemain tidak terdefinisi.
 - Baris 10 digunakan jika banyaknya pemainX lebih banyak dari pemainY maka pemain akan ditambahkan pada pemainY. Kemudian akan diambil satu pemain dari pemainX yang dijadikan lawan bagi pemain yang baru dari pemainY.
 - Selanjutnya akan diinisialisasi detail dari tiap pemain termasuk lawan dan poin masing-masing. Kemudian pemain akan bergabung dengan *room* dengan nama *room* adalah *username* pemain lawan.
 - Pada baris 21 dilakukan pengambilan data pertanyaan dari *database*. Data pertanyaan tersebut akan digunakan selama permainan.
 - Jika pemainX kurang dari pemainY maka pemain akan ditambahkan pada pemainX. Kemudian akan dibuat satu *room* dengan nama diambil dari *username* dari pemain.
- **Implementasi Fungsi Siap Babak**

Fungsi Siap Babak digunakan untuk menyiapkan data sebelum permainan pada setiap babak dimulai. Berikut ini adalah *pseudocode* fungsi siap babak yang ditunjukkan pada **Tabel 3.8**.

Tabel 3.8 Pseudocode Fungsi Siap Babak di Server

Baris	Pseudocode
1	DECLARATION
2	socket
3	current_soal
4	
5	DESCRIPTION
6	IF socket.tipe = Y THEN
7	SET data.user = socket.username

8	SET data.tipe = Y
9	SET data.soal = current_soal
10	CALL io.to socket.name_room with data
11	ENDIF

Penjelasan *pseudocode* dari fungsi siap babak pada **Tabel 3.8** adalah

- Variabel yang digunakan dalam fungsi ini yaitu socket dan current soal. Current_soal merupakan variabel yang berisi data soal yang siap dikirim.
- Jika pemain adalah pemain dengan tipe Y maka akan dilakukan inialisasi data *user* yaitu *username*, tipe yaitu Y, data soal yaitu *current_soal*.
- Kemudian pada baris 10 akan data akan dikirim pada setiap pemain pada satu nama *room* yang sama.

• **Implementasi Fungsi Jawab Babak**

Fungsi Jawab Babak berguna untuk menangani jawaban atas soal yang dipilih oleh pemain. Jawaban akan dicocokkan dengan data jawaban yang tersimpan di basis data, jika jawaban benar akan dilakukan penambahan poin untuk pemain tersebut dan kemudian dilakukan pergantian giliran pemain. Berikut ini adalah *pseudocode* fungsi jawab babak yang ditunjukkan pada **Tabel 3.9**.

Tabel 3.9 Pseudocode Fungsi Jawab Babak di Server

Baris	Pseudocode
1	DECLARATION
2	data, benar
3	socket
4	question
5	
6	DESCRIPTION
7	IF socket.tipe = Y THEN
8	IF data.time > 1 AND data.answer != 'noFight' AND data.answer != 'noAnswer' THEN
9	SET soal where id = data.id
10	IF soal.answer = data.answer THEN
11	SET benar = true
12	IF data.username = socket.username THEN
13	SET poin AND socket.poin to socket.poin + penambahanBabak1
14	ELSE
15	SET poin AND socket.musuh.poin to socket.musuh.poin + penambahanBabak1



```
16         ENDIF
17     ELSE
18         SET benar = false
19         IF data.username = socket.username THEN
20             SET poin to socket.poin
21         ELSE
22             SET poin to socket.musuh.poin
23         ENDIF
24     ENDIF
25     SET data.isBenar to benar
26     SET data.username to data.username
27     SET data.answer to data.answer
28     SET data.poin to poin
29     CALL io.to socket.name_room with data
30 ELSE
31     CALL io.to socket.name_room with 'noAnswer' AND
data.username
32     ENDIF
33     IF data.tipe = Y THEN
34         SET tipe to X
35         SET shiftUser to socket.musuh
36     ELSE
37         SET tipe to Y
38         SET shiftUser to socket
39     ENDIF
40     IF LENGTH question >= 1 THEN
41         SET current_soal to question.last
42         SET data.user to shiftUser.username
43         SET data.tipe to tipe
44         SET data.soal to current_soal
45         CALL io.to socket.name_room with data
46     ELSE
47         SET data.usernameKu = socket.username
48         SET data.poinKu = socket.poin
49         SET data.usernameMusuh = socket.musuh.username
50         SET data.poinMusuh = socket.musuh.poin
51         CALL io.to socket.name_room with data
52     ELSE
53         CALL socket.musuh.emit with data
```

Penjelasan *pseudocode* dari fungsi jawab babak pada **Tabel 3.9** adalah

- Variabel yang digunakan dalam fungsi ini yaitu data, benar, socket, dan *question*. Socket berisi detail informasi dari setiap pemain yang terhubung. *Question* berisi informasi daftar soal yang digunakan dalam permainan.
- Fungsi ini akan berjalan salah satu pemain dengan tipe Y seperti yang ditunjukkan pada baris 7. Jika pemain bertipe X maka akan mengirim informasi kepada pemain lawan dengan tipe Y seperti yang ditunjukkan pada baris 53.
- Pada baris 8 dilakukan pengecekan kondisi jika waktu yang dikirimkan lebih dari 1 dan informasi jika pemain benar menjawab soal.
- Kemudian pada baris 9 program akan mengambil data soal dari *database* dengan id yang sama dengan informasi yang diterima.
- Pada baris 10-17 jika jawaban yang diterima sama dengan jawaban yang didapat dari *database*, maka variabel benar akan bernilai true, dan dilakukan penambahan poin ada pemain yang menjawab soal tersebut.
- Baris 18-22 tidak ada penambahan poin dan variabel benar akan bernilai false.
- Pada baris 29 dilakukan pemanggilan fungsi untuk mengirim informasi mengenai apakah jawaban benar, *username* pemain, jawaban yang dikirim dan poin yang didapat.
- Baris 31 akan mengirim informasi bahwa pemain tidak menjawab soal.
- Baris 33-39 digunakan untuk mengganti giliran pemain untuk menjawab soal yang diberikan.
- Baris 40-45 merupakan kondisi jumlah jika pertanyaan masih tersedia maka program akan mengirimkan lagi pertanyaan selanjutnya kepada pemain untuk melanjutkan permainan.
- Namun jika pertanyaan habis maka permainan selesai dan program akan mengirimkan informasi poin akhir dari setiap pemain seperti ditunjukkan pada baris 47-51.

- **Implementasi Fungsi *Game* Selesai**

Fungsi *game* selesai digunakan untuk menghitung poin akhir masing-masing pemain. Kemudian data permainan akan disimpan pada basis data sebagai data riwayat permainan. Berikut ini adalah *pseudocode* fungsi *game* selesai yang ditunjukkan pada **Tabel 3.10**.

Tabel 3.10 Pseudocode Fungsi Game Selesai

Baris	Pseudocode
1	DECLARATION
2	socket
3	jumlahSoal
4	
5	DESCRIPTION
6	IF counter > jumlahSoal THEN
7	SET match.username to socket.username AND socket.musuh.username
8	SET match.poin to socket.poin AND socket.musuh.poin
9	SAVE match
10	SET data.username to socket.username
11	SET data.poin to socket.poin
12	SET data.musuh.username to socket.musuh.username
13	SET data.musuh.poin to socket.musuh.poin
14	CALL io.to socket.name_room with data

Penjelasan pseudocode dari fungsi *game* selesai pada **Tabel 3.10** adalah

- Variabel yang digunakan dalam fungsi ini yaitu *socket*, dan *jumlahSoal*. *socket* merupakan informasi detail dari pemain yang terhubung, dan *jumlahSoal* merupakan jumlah total yang digunakan selama permainan.
- Pada baris 6 akan dilakukan pengondisian dengan *counter* yang merupakan penghitung soal lebih dari *jumlahSoal*, maka akan menjalankan baris 7-14.
- Baris 7-8 dilakukan penambahan nilai pada variabel *match* yaitu *username* dan *poin* dari setiap pemain.
- Kemudian pada baris 9 data *match* akan disimpan pada basis data.
- Baris 10-13 dilakukan penambahan nilai data *username* dan *poin* akhir dari setiap pemain yang kemudian akan dikirim pada masing-masing pemain tersebut seperti yang ditunjukkan pada baris 14.

3.3.3 Implementasi Game Client

Game Client dibuat sebagai program yang berwujud aplikasi *game* yang dapat berjalan pada sistem operasi Android. Untuk dapat menjalankan aplikasi *game* diperlukan hubungan koneksi dengan *game server*. Dalam aplikasi *game* ini terdapat beberapa halaman yang dalam implementasinya dibagi dalam beberapa tahap yaitu implementasi halaman *login*, implementasi halaman *home*, implementasi babak 1, implementasi babak 2, dan implementasi babak 3 serta implementasi halaman riwayat.

3.3.3.1 Implementasi Halaman Login

Halaman *login* digunakan untuk masuk ke aplikasi *game* menggunakan akun pemain. Untuk dapat masuk dibutuhkan *username* dan *password* yang dimasukkan oleh pemain. Pada implementasi halaman *login* terdapat tahap implementasi fungsi *login* dan implementasi antarmuka halaman *login*.

- **Implementasi Fungsi Login**

Fungsi *login* digunakan untuk melakukan pengecekan *username* dan *password* telah dimasukkan. Pengecekan dilakukan dengan cara mengirimkan informasi ke *game server*. **Tabel 3.11** akan ditunjukkan bagaimana proses tersebut dijalankan.

Tabel 3.11 Pseudocode Fungsi Login

Baris	Pseudocode
1	DECLARATION
2	username
3	password
4	
5	DESCRIPTON
6	IF CALL loginUser with username, password = success THEN
7	IF CALL socket.emit with 'successLogin', username
8	THEN
9	CALL pindahHalaman with 'home'
10	ELSE
11	CALL alert with 'user sudah login'
12	ENDIF
13	ELSE
14	CALL alert with 'login gagal'
15	ENDIF

Penjelasan *pseudocode* dari fungsi *login* pada **Tabel 3.11** adalah

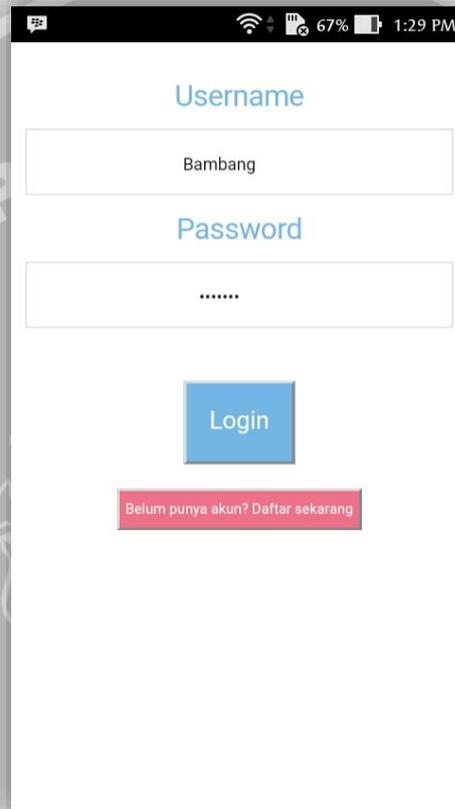
- Terdapat dua buah variabel utama yang digunakan dalam fungsi ini yaitu *username* dan *password*.
- Pada baris 6 menjelaskan bahwa fungsi akan memanggil fungsi *loginUser* yang kemudian akan dikirim ke *game server*. Parameter yang digunakan adalah *username*, dan *password*.
- Jika respon dari *game server* adalah bahwa *username* dan *password* diterima maka akan menjalankan proses baris no 7 - 11. Yaitu program *game* akan memanggil fungsi *socket.emit* untuk mengirim ke *server* untuk mengecek bahwa *user* belum *login*. Jika pemain dinyatakan sukses *login* maka *game* akan berpindah halaman menuju halaman *home*. Jika tidak maka akan menampilkan peringatan bahwa akun pemain sedang digunakan.



- Jika respon dari *game server* menyatakan bahwa *login* tidak berhasil maka akan menampilkan informasi *login* gagal seperti yang ditunjukkan pada baris 13.

- **Implementasi Antarmuka Halaman *Login***

Hasil antarmuka halaman *login* berisi kolom untuk input *username* dan *password* dan tombol *login*. Berikut ini adalah hasil antarmuka *login* yang ditunjukkan pada **Gambar 3.18**.



Gambar 3.18 Hasil Antarmuka *Login*

3.3.3.2 Implementasi Halaman *Home*

Halaman *Home* merupakan halaman yang berisi detail pemain seperti foto profil, *username*, dan poin serta terdapat tombol untuk memulai permainan. Pada implementasi Halaman *Home* terdapat beberapa tahap implementasi yaitu implementasi fungsi *challenge*, implementasi fungsi *ready wait*, implementasi fungsi *pre play* dan implementasi antarmuka halaman *home*.

- **Implementasi Fungsi *Challenge***

Fungsi *Challenge* digunakan ketika pemain menekan tombol mulai main untuk memulai permainan. Berikut ini adalah fungsi *challenge* yang ditunjukkan pada **Tabel 3.12**.

Tabel 3.12 Pseudocode Fungsi Challenge

Baris	Pseudocode
1	DECLARATION
2	localUsername
3	
4	DESCRIPTION
5	IF localUsername = null THEN
6	CALL pindahHalaman with 'login'
7	ELSE
8	CALL socket.emit 'search'
9	ENDIF
10	
11	IF CALL socket.on with halo THEN
12	CALL play
13	ENDIF

Penjelasan *pseudocode* dari fungsi *login* pada **Tabel 3.12** adalah

- Variabel yang digunakan dalam fungsi ini yaitu localUsername.
- Pada baris 5 dilakukan pengondisian jika localUsername kosong maka akan *logout* dari *game* dengan berpindah menuju halaman *login*.
- Jika localUsername tidak kosong maka akan memanggil fungsi socket.emit dengan parameter 'search'. Fungsi socket.emit digunakan untuk mengirim data menuju *game server*.
- Pada baris 11-13 ditunjukkan ketika program memanggil fungsi socket.on dengan paramater 'halo'. Fungsi socket.on digunakan untuk menerima perintah dari *game server*. Jika perintah berhasil diterima maka program akan memanggil fungsi *play*.

• **Implementasi Fungsi Ready Wait**

Fungsi ini digunakan untuk menginisialisasi data detail pemain lawan yang akan dihadapi. Berikut ini adalah *pseudocode* fungsi *ready wait* yang ditunjukkan pada **Tabel 3.13**.

Tabel 3.13 Pseudocode Fungsi Ready Wait

Baris	Pseudocode
1	DECLARATION
2	localUsername
3	localPoin
4	
5	DESCRIPTION

6	CALL socket.emit with 'ready wait'
7	CALL socket.on with 'ready wait other' RETURN data
8	CALL socket.emit with 'all ready wait'
9	SET detailMusuh.username to data.username
10	SET detailMusuh.poin to data.poin
11	SET detailMusuh.poinBabak1 to 0
12	SET detailMusuh.poinBabak2 to 0
13	SET detailMusuh.poinBabak3 to 0
14	SET detailKu.username to localUsername
15	SET detailKu.poin to localPoin
16	SET detailKu.poinBabak1 to 0
17	SET detailKu.poinBabak2 to 0
18	SET detailKu.poinBabak3 to 0

Penjelasan *pseudocode* dari fungsi *ready wait* pada **Tabel 3.13** adalah

- Variabel yang digunakan dalam fungsi ini adalah *localUsername* dan *localPoin*.
 - Pada baris 6 ditunjukkan proses pemanggilan fungsi *socket.emit* untuk mengirim informasi 'ready wait' ke *server*.
 - Jika *game* menerima informasi 'ready with other', maka akan menjalankan program pada baris 9-18.
 - Baris 9 program akan mengirim informasi 'all ready wait' ke *server*.
 - Pada baris 10-18 dilakukan inisialisasi nilai detail pemain dan detail musuh. Detail berisi *username* dan *poin* pada tiap babak.
- **Implementasi Fungsi Pre Play**

Fungsi ini digunakan untuk menampilkan halaman menuju permainan babak 1. Pada fungsi ini juga menjalankan penanda waktu dan dilengkapi dengan informasi penjelasan mengenai permainan pada babak 1. Berikut ini adalah *pseudocode* fungsi *pre play* yang ditunjukkan pada **Tabel 3.14**.

Tabel 3.14 Pseudocode Fungsi Pre Play

Baris	Pseudocode
1	DECLARATION
2	timerPrePlay = 10
3	stopPrePlay = defined
4	
5	DESCRIPTION
6	IF stopPrePlay != undefined THEN
7	RETURN

8	ELSE
9	IF timerPrePlay > 0 THEN
10	DECREMENT timerPrePlay
11	ELSE
12	SET stopPrePlay to undefined
13	CALL goBabak1

Penjelasan *pseudocode* dari fungsi *pre play* pada **Tabel 3.14** adalah

- Variabel `timerPrePlay` digunakan sebagai penanda waktu menuju babak 1 yaitu bernilai 10 detik. Dan variabel `stopPrePlay` digunakan untuk menghentikan penanda waktu.
- Baris 6-7 jika `stopPreplay` terdefinisi maka proses tidak akan dilanjutkan. Jika tidak maka `timerPrePlay` akan berkurang nilainya sampai bernilai 1 ditunjukkan pada baris 9-10.
- Jika `timerPrePlay` bernilai 1 maka timer akan berhenti dan memanggil fungsi `goBabak1` ditunjukkan pada baris 12-13.

- **Implementasi Antarmuka Halaman *Home***

Hasil antarmuka halaman *Home* berisi foto profil pemain, *username*, poin pemain dan tombol “Mulai Main” untuk menjalankan fungsi *challenge* serta tombol “Keluar” untuk *logout* atau keluar dari akun *game* yang digunakan. Hasil antarmuka halaman *Home* ditunjukkan pada **Gambar 3.19**.

- **Implementasi Antarmuka Pre Babak 1**

Hasil antarmuka pre babak 1 menampilkan penanda waktu selama 10 detik dan informasi penjelasan mengenai babak 1. Kemudian pada antarmuka ini juga terdapat informasi dari kedua pemain yaitu foto profil, *username* dan poin. Hasil antarmuka pre babak 1 ditunjukkan pada **Gambar 3.20**.



Gambar 3.19 Hasil Antarmuka Halaman *Home*



Gambar 3.20 Antarmuka Pre Babak 1

3.3.3.3 Implementasi Babak 1

Pada babak 1 pemain akan bergantian menjawab soal yang diterima dari *game server*. Batas waktu menjawab untuk setiap soal adalah 10 detik yang ditandai dengan penanda waktu yang berjalan mundur. Pada implementasi babak 1 dibagi dalam beberapa tahap yaitu implementasi fungsi seperti fungsi *go babak 1*, *fight timer 1*, *jawab babak 1*, *result answered babak 1*, dan babak 1 *done*. Kemudian dilanjutkan dalam tahap implementasi antarmuka babak 1.

- **Implementasi Fungsi Go Babak 1**

Fungsi ini digunakan untuk memulai permainan pada babak 1. Program akan mengirim informasi ke *game server* bahwa program telah siap dan kemudian juga akan menerima informasi bahwa program lawan juga telah siap yang selanjutnya permainan babak 1 akan dimulai. Pada babak 1 ini setiap pemain akan mendapatkan giliran menjawab. Soal dan jawaban akan ditampilkan sesuai informasi yang diterima dari *game server*. Berikut ini adalah *pseudocode* fungsi *go babak 1* yang ditunjukkan pada **Tabel 3.15**.

Tabel 3.15 Pseudocode Fungsi Go Babak 1

Baris	Pseudocode
1	DECLARATION
2	detailKu.username
3	tipeUser
4	textShift = null
5	isShift = false
6	soal = null
7	jawaban = null
8	opponentAct = null
9	secondBabak1 = 10
10	
11	DESCRIPTION
12	CALL socket.emit with 'ready babak 1'
13	CALL socket.on with 'ready other babak 1' RETURN
14	CALL socket.emit with 'all ready babak 1'
15	CALL socket.on with 'soal babak 1' RETURN data
16	SET waktu to secondBabak1
17	IF data.user = detailKu.username THEN
18	SET tipeUser to data.tipe
19	SET isShift to true
20	SET textShift to 'Giliran Anda, silahkan jawab'
21	ELSE
22	SET isShift to false

23	SET textShift to 'Giliran' data.user
24	ENDIF
25	CALL fightTimer
26	SET soal to data.soal.question
27	SET jawaban to data.soal.choice

Penjelasan *pseudocode* dari fungsi *go* babak 1 pada **Tabel 3.15** adalah

- Ada beberapa variabel yang digunakan dalam fungsi ini yaitu `detailKu.username` yang digunakan sebagai *username* dari pemain, `tipeUser` digunakan untuk membedakan jenis pemain yaitu tipe X atau Y, `textShift` berisi informasi yang akan ditampilkan dalam giliran pemain, `isShift` digunakan untuk mengetahui bahwa saat itu adalah giliran pemain, `soal` merupakan informasi yang menampilkan soal, `jawaban` merupakan informasi pilihan jawaban yang dapat dipilih pemain, `opponentAct` merupakan informasi yang digunakan untuk menampilkan aktivitas dari lawan dan kemudian `secondBabak 1` merupakan jumlah waktu lama pemain harus menjawab soal yaitu 10 detik.
- Pada baris 12 program mengirimkan informasi '*ready* babak 1' menuju *game server* untuk menyatakan bahwa program telah siap.
- Pada baris 13-14 program akan menerima informasi bahwa pemain lawan telah siap untuk bermain yang kemudian program akan mengirim informasi bahwa semua pemain telah siap menuju *game server*.
- Pada baris 15 program akan menerima informasi soal dan pilihan jawaban dari *game server*. Baris 16 waktu akan ditentukan dengan lama 10 detik.
- Baris 17-20 menyatakan bahwa jika data *username* yang diterima sama dengan *username* pemain maka tipe *user* akan ditentukan sama dengan informasi tipe yang diterima, kemudian program akan menentukan bahwa itu adalah giliran pemain yaitu dengan menampilkan informasi kepada pemain "Giliran Anda, silakan menjawab".
- Pada baris 21-23 merupakan pernyataan jika data *username* tidak sama dengan *username* pemain dengan kata lain bahwa saat itu bukanlah giliran pemain melainkan giliran dari lawan.
- Pada baris 25 program akan memanggil fungsi `fightTimer` untuk menjalankan penanda waktu.
- Pada baris 26 dan 27 program menampilkan informasi soal dan pilihan jawaban yang diterima dari *game server*.

- Implementasi Fungsi *Fight Timer 1*

Fungsi ini digunakan untuk menjalankan penanda waktu sebagai informasi batas waktu dalam menjawab soal pada babak 1. Berikut ini adalah *pseudocode* fungsi *fight timer 1* yang ditunjukkan pada **Tabel 3.16**.

Tabel 3.16 Pseudocode Fungsi Fight Timer 1

Baris	Pseudocode
1	DECLARATION
2	stopTimer1 = defined
3	waktu = 10
4	isShift
5	data
6	
7	DESCRIPTION
8	IF waktu > 0 THEN
9	DECREASE waktu
10	ELSE
11	IF isShift THEN
12	SET isShift to false
13	SET dataJawaban.username to data.user
14	SET dataJawaban.time to waktu
15	SET dataJawaban.answer to 'noAnswer'
16	SET dataJawaban.tipe to data.tipe
17	CALL socket.emit with 'answer babak 1', dataJawaban
18	ENDIF
19	SET stopTimer1 to undefined
20	ENDIF

Penjelasan *pseudocode* fungsi *fight timer 1* pada **Tabel 3.16** adalah :

- Variabel yang digunakan dalam fungsi ini yaitu stopTimer1 yang digunakan untuk menghentikan penanda waktu, kemudian lama waktu, isShift yang digunakan untuk mengetahui giliran, dan data yang didapat dari *game server*.
- Pada baris 8-10 merupakan pernyataan jika waktu masih tersedia maka akan dilakukan pengurangan lama waktu.
- Pada baris 10-20 merupakan pernyataan jika waktu sudah habis, pernyataan akan berlaku jika saat itu adalah giliran dari pemain. Baris 12 giliran akan beralih kepada lawan. Baris 13-15 waktu akan bernilai waktu saat itu yaitu 0, jawaban akan bernilai 'noAnswer', tipe *user* akan bernilai dari data tipe yang diterima. Kemudian program akan

mengirimkan data jawaban menuju *game server*. dan penanda waktu telah berhenti.

- **Implementasi Fungsi Jawab Babak 1**

Fungsi ini digunakan untuk mengirimkan informasi pilihan jawaban dari pemain menuju ke *game server*. Detail informasi jawaban yang dikirimkan yaitu *id*, *username*, waktu, jawaban dan tipe *user*. Berikut ini adalah *pseudocode* fungsi jawab babak 1 yang ditunjukkan pada **Tabel 3.17**.

Tabel 3.17 Pseudocode Fungsi Jawab Babak 1

Baris	Pseudocode
1	DECLARATION
2	userAnswer
3	stopTimer1 = defined
4	idSoalBabak1
5	detailKu.username
6	tipeUser
7	
8	DESCRIPTION
9	SET stopTimer1 to undefined
10	SET dataJawaban._id to idSoalBabak1
11	SET dataJawaban.username to detailKu.username
12	SET dataJawaban.time to waktu
13	SET dataJawaban.answer to userAnswer
14	SET dataJawaban.tipe to tipeUser
15	SET isShift to false
16	CALL socket.emit with 'answer babak 1', dataJawaban

Penjelasan *pseudocode* fungsi jawab babak 1 dari **Tabel 3.17** adalah:

- Ada beberapa variabel yang digunakan dalam fungsi ini yaitu *userAnswer* yang merupakan jawaban dari *user*, *stopTimer* untuk menghentikan penanda waktu, *idSoalBabak1* merupakan id dari soal yang ditampilkan, *username* dari pemain, dan waktu dari pemain.
- Pada baris 9 penanda waktu akan berhenti.
- Baris 10-14 merupakan pernyataan untuk mengisi nilai detail jawaban seperti id soal, *username*, waktu, jawaban serta tipe pemain.
- Pada baris 15 giliran beralih kepada pemain lawan.
- Pada baris 16 program akan mengirim informasi detail jawaban ke *game server*.



- **Implementasi Fungsi Result Answered Babak 1**

Fungsi ini digunakan untuk menampilkan informasi hasil jawaban yang dipilih oleh pemain apakah benar atau tidak. Kemudian akan dilakukan penambahan poin jika jawaban adalah benar. Berikut ini adalah *pseudocode* fungsi *result answered* babak 1 yang ditunjukkan pada **Tabel 3.18**.

Tabel 3.18 Pseudocode Fungsi Result Answered Babak 1

Baris	Pseudocode
1	DECLARATION
2	data
3	
4	DESCRIPTION
5	SET opponentAct to data.username 'menjawab' data.answer
6	IF data.username = detailKu.username THEN
7	detailKu.poin = data.poin
8	ELSE
9	detailMusuh.poin = data.poin
10	ENDIF
11	IF data.isBenar THEN
12	SET opponentAct to 'Benar'
13	ELSE
14	SET opponentAct to 'Salah'
15	ENDIF

Penjelasan *pseudocode* fungsi *result answered* babak 1 dari **Tabel 3.18** adalah:

- Pada baris 5 merupakan pernyataan untuk menampilkan informasi bahwa pemain maupun lawan memilih jawaban tertentu.
- Baris 6-10 jika data *username* sama dengan *username* pemain maka poin merupakan milik pemain itu sendiri namun jika tidak maka poin adalah milik dari pemain lawan.
- Baris 11-15 jika jawaban bernilai benar maka program akan menampilkan informasi benar namun jika salah maka informasi yang ditampilkan adalah salah.

- **Implementasi Fungsi Babak 1 Done**

Fungsi ini digunakan ketika permainan babak 1 sudah selesai. Fungsi akan menghitung poin babak 1 dari setiap pemain dan menampilkan informasi penjelasan permainan babak 2. Berikut ini adalah *pseudocode* fungsi babak 1 *done* yang ditunjukkan pada **Tabel 3.19**.

Tabel 3.19 Pseudocode Fungsi Babak 1 Done

Baris	Pseudocode
1	DECLARATION
2	data
3	detailKu.poin
4	detailMusuh.poin
5	time2Babak2 = 10
6	stopTimerKeBabak2 = defined
7	
8	DESCRIPTION
9	SET detailKu.poinBabak1 to detailKu.poin
10	SET detailMusuh.poinBabak1 to detailMusuh.poin
11	IF username = detailKu.username THEN
12	SET detailKu.poin to data[0].poin
13	SET detailMusuh.poin to data[1].poin
14	ELSE
15	SET detailKu.poin to data[1].poin
16	SET detailMusuh.poin to data[0].poin
17	ENDIF
18	SET waktu2Next to time2Babak2
19	IF stopTimerKeBabak2 = defined THEN RETURN
20	ELSE
21	IF waktu2Next > 0 THEN
22	DECREASE waktu2Next
23	ELSE
24	IF stopTimerKeBabak2 = defined THEN
25	SET stopTimerKeBabak2 to undefined
26	ENDIF
27	CALL goBabak2
28	ENDIF
29	ENDIF

Penjelasan *pseudocode* dari fungsi babak 1 *done* pada Tabel 3.19 adalah

- Baris 9-10 digunakan untuk menampilkan poin babak 1 dari masing-masing pemain berdasarkan data poin yang diterima oleh *game server*.
- Baris 11-17 digunakan jika *username* sama dengan *username* pemain maka data poin merupakan milik pemain namun jika tidak sama maka data poin adalah milik pemain lawan.

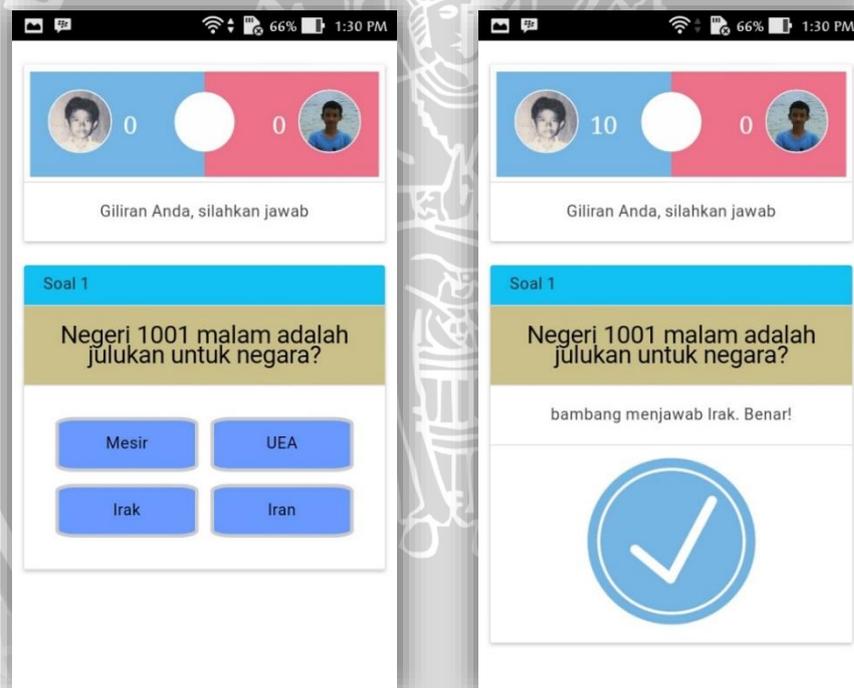
- Baris 18 digunakan untuk mengisi nilai waktu dengan time2Babak2 yaitu 10 detik.
- Baris 19-29 digunakan untuk menjalankan penanda waktu jika waktu masih tersedia maka dilakukan pengurangan waktu, namun jika waktu habis maka penanda waktu akan berhenti dan memanggil fungsi goBabak2.

- **Implementasi Antarmuka Babak 1**

Berikut ini adalah rancangan antarmuka babak 1 yang ditunjukkan pada **Gambar 3.21** dan **Gambar 3.22**.

Gambar 3.21 menunjukkan hasil antarmuka permainan babak 1 yang menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal dan 4 pilihan jawaban.

Gambar 3.22 menunjukkan hasil antarmuka ketika permainan babak 1 selesai. Dalam antarmuka tersebut menampilkan foto profil pemain dan poin yang didapat selama permainan babak 1. Selain itu pada antarmuka ini juga menampilkan penanda waktu dan informasi penjelasan mengenai permainan babak 2.



Gambar 3.21 Hasil Antarmuka Babak 1



Gambar 3.22 Hasil Antarmuka Babak 1 Selesai

3.3.3.4 Implementasi Babak 2

Pada babak 2 ini pemain akan bergantian memilih satu soal untuk dilempar kepada pemain lawan. Dan kemudian pemain lawan harus menjawab soal yang dipikirkan. Waktu untuk memilih soal dan menjawab masing-masing yaitu selama 10 detik. Pada implementasi babak 2 ini dibagi dalam beberapa tahap yaitu implementasi fungsi seperti fungsi *go* babak 2, *do* pilih, *jawab* babak 2, *result answered* babak 2, dan babak 2 *done*. Kemudian dilanjutkan dalam tahap implementasi antarmuka babak 2.

- **Implementasi Fungsi Go Babak 2**

Fungsi ini digunakan untuk memulai permainan pada babak 2. Program akan mengirim informasi ke *game server* bahwa program telah siap. Kemudian program akan menerima informasi bahwa program lawan juga telah siap. Selanjutnya permainan babak 2 akan dimulai. Pada babak 2 ini setiap pemain akan mendapatkan giliran memilihkan soal untuk pemain lawan. Soal dan jawaban akan ditampilkan sesuai informasi yang diterima dari *game server*. Berikut ini adalah *pseudocode* fungsi *go* babak 2 yang ditunjukkan pada **Tabel 3.20**.

Tabel 3.20 Pseudocode Fungsi Go Babak 2

Baris	Pseudocode
1	DECLARATION
2	secondFight2 = 10
3	

4	DESCRIPTION
5	CALL socket.emit with 'ready babak 2'
6	CALL socket.on with 'ready other babak 2' RETURN
7	CALL socket.emit with 'all ready babak 2'
8	CALL socket.on with 'pilihan soal' RETURN data
9	SET tipeUser2 to data.tipe
10	SET waktu to secondFight2
11	IF stopTimer2 = defined THEN
12	stopTimer2 = undefined
13	ENDIF
14	IF stopTimer2 = defined THEN RETURN
15	ELSE
16	IF waktu > 0 THEN
17	DECREASE waktu
18	ELSE
19	IF stopTimer2 = defined THEN
20	stopTimer2 = undefined
21	ENDIF
22	CALL jawabBabak2 with 'noChoose'
23	ENDIF
24	ENDIF

Penjelasan *pseudocode* dari fungsi *go babak 2* pada **Tabel 3.20** adalah

- Variabel yang digunakan dalam fungsi ini adalah *secondFight2* yang digunakan sebagai penanda waktu pada fungsi ini yaitu 10 detik.
- Pada baris 5 program mengirimkan informasi 'ready babak 2' menuju *game server* untuk menyatakan bahwa program telah siap.
- Pada baris 6-7 program akan menerima informasi bahwa pemain lawan telah siap untuk bermain yang kemudian program akan mengirim informasi bahwa semua pemain telah siap menuju *game server*.
- Pada baris 8-24 program akan menerima informasi pilihan soal. Kemudian tipe *user* akan bernilai sesuai data yang diterima. Dan kemudian dijalankan penanda waktu. waktu akan terus berkurang hingga waktu habis. jika waktu habis maka akan memanggil fungsi *jawabBabak2* dengan 'noChoose'.

• Implementasi Fungsi *Do Pilih*

Fungsi ini digunakan untuk menampilkan informasi soal yang dipikirkan oleh pemain lawan. Fungsi ini juga akan menjalankan penanda waktu sebagai informasi batas waktu untuk menjawab. Berikut ini adalah *pseudocode* fungsi *do pilih* yang ditunjukkan pada **Tabel 3.21**.

Tabel 3.21 Pseudocode Fungsi Do Pilih

Baris	Pseudocode
1	DECLARATION
2	pilihan
3	mandek2 = defined
4	
5	DESCRIPTION
6	CALL socket.emit with 'pertanyaan pilihan', pilihan
7	CALL socket.on with 'jawaben rek' RETURN data
8	IF mandek2 = defined THEN
9	SET mandek2 to undefined
10	ENDIF
11	IF mandek2 = defined THEN RETURN
12	ELSE
13	IF waktu > 0 THEN
14	DECREASE waktu
15	ELSE
16	CALL jawabBabak2 with 'noAnswer'
17	ENDIF
18	ENDIF
19	SET soal to data.soal.question
20	SET jawaban to data.soal.choice
21	SET idSoal to data.soal._id

Penjelasan *pseudocode* dari fungsi *go* babak 1 pada Tabel 3.21 adalah

- Variabel yang digunakan dalam fungsi ini yaitu pilihan merupakan soal yang dipilih oleh pemain, kemudian mandek2 untuk menghentikan penanda waktu.
- Pada baris 6 program akan mengirim informasi pilihan soal pemain menuju *game server*.
- Pada baris 7-21 program akan menerima informasi pilihan soal yang dipikirkan oleh pemain lawan. Kemudian penanda waktu akan berjalan dan pemain harus menjawab soal dengan pilihan jawaban yang tersedia. Jika sampai waktu habis pemain tidak menjawab maka program akan memanggil fungsi jawabBabak2 dengan 'noAnswer'.

• **Implementasi Fungsi Jawab Babak 2**

Fungsi ini digunakan ketika pemain menjawab soal. Kemudian informasi jawaban akan dikirim ke *game server*. Berikut ini adalah *pseudocode* fungsi jawab babak 2 yang ditunjukkan pada **Tabel 3.22**.



Tabel 3.22 Pseudocode Fungsi Jawab Babak 2

Baris	Pseudocode
1	DECLARATION
2	userAnswer
3	stopTimer2 = defined
4	DESCRIPTION
5	SET dataJawaban.username to detailKu.username
6	SET dataJawaban.time to waktu
7	SET dataJawaban.id to idSoal
8	SET dataJawaban.answer to userAnswer
9	SET dataJawaban.tipe to tipeUser
10	CALL socket.emit with 'answer babak 2', dataJawaban

Penjelasan *pseudocode* dari fungsi jawab babak 2 pada **Tabel 3.22** adalah

- Ada beberapa variabel yang digunakan dalam fungsi ini yaitu userAnswer yang merupakan jawaban dari user dan stopTimer untuk menghentikan penanda waktu.
- Pada baris 5-9 program akan mengisi informasi detail jawaban seperti username, waktu, id soal, jawaban, dan tipe user.
- Pada baris 10 program akan mengirim informasi detail jawaban ke game server.

• **Implementasi Fungsi Result Answered Babak 2**

Fungsi ini digunakan untuk menampilkan informasi hasil jawaban yang dipilih oleh pemain apakah benar atau tidak. Kemudian akan dilakukan penambahan poin jika jawaban adalah benar. Berikut ini adalah *pseudocode* fungsi *result answered* babak 2 yang ditunjukkan pada **Tabel 3.23**.

Tabel 3.23 Pseudocode Fungsi Result Answered Babak 2

Baris	Pseudocode
1	DECLARATION
2	userAnswer
3	data
4	opponentAct
5	
6	DESCRIPTION
7	SET opponentAct to data.username 'menjawab' data.answer
8	IF data.username = detailKu.username THEN
9	detailKu.poin = data.poin
10	ELSE



11	detailMusuh.poin = data.poin
12	ENDIF
13	IF data.isBenar THEN
14	SET opponentAct to 'Benar'
15	ELSE
16	SET opponentAct to 'Salah'
17	ENDIF
18	CALL socket.on 'noChoose' RETURN data
19	SET textShift to 'waktu habis'
20	IF data.username = detailKu.username THEN
21	SET detailKu.poin to data.poin
22	ELSE
23	SET detailMusuh.poin to data.poin
24	ENDIF

Penjelasan *pseudocode* dari fungsi *result answerd* babak 2 pada **Tabel 3.23** adalah

- Variabel yang digunakan dalam fungsi ini adalah jawaban dari *user* dan data yang diterima dari *game server*.
- Pada baris 7 program akan menampilkan informasi bahwa pemain menjawab suatu pilihan jawaban.
- Pada baris 8-12 menunjukkan pernyataan jika data *username* sama dengan *username* pemain maka poin adalah milik pemain tersebut namun jika tidak maka poin adalah milik pemain lawan.
- Baris 13-16 menunjukkan bahwa jika jawaban adalah benar maka program akan menampilkan informasi benar jika salah maka akan menampilkan informasi bahwa jawaban adalah salah.
- Baris 18-24 program akan menerima informasi dari *game server* jika pemain tidak memilih pilihan soal. Maka *game* akan menampilkan informasi bahwa waktu habis. Kemudian dilakukan penambahan poin dengan melihat *username* dari *server* dan *username* pemain.

- **Implementasi Fungsi Babak 2 Done**

Fungsi ini digunakan ketika permainan babak 2 sudah selesai. Kemudian program akan menghitung poin babak 2 dari setiap pemain dan menampilkan informasi penjelasan permainan pada babak 3. Berikut ini adalah *pseudocode* fungsi babak 2 *done* yang ditunjukkan pada **Tabel 3.24**.

Tabel 3.24 Pseudocode Fungsi Babak 2 Done

Baris	Pseudocode
1	DECLARATION
2	data
3	time2Babak3 = 5
4	stopTimerKeBabak3 = defined
5	
6	DESCRIPTION
7	SET detailKu.poinBabak2 to detailKu.poin - detailKu.poinBabak1
8	SET detailMusuh.poinBabak2 to detailMusuh.poin - detailMusuh.poinBabak1
9	IF username = detailKu.username THEN
10	SET detailKu.poin to data[0].poin
11	SET detailMusuh.poin to data[1].poin
12	ELSE
13	SET detailKu.poin to data[1].poin
14	SET detailMusuh.poin to data[0].poin
15	ENDIF
16	SET waktu2Next to time2Babak3
17	IF stopTimerKeBabak3 = defined THEN RETURN
18	ELSE
19	IF waktu2Next > 0 THEN
20	DECREASE waktu2Next
21	ELSE
22	IF stopTimerKeBabak3 = defined THEN
23	SET stopTimerKeBabak3 to undefined
24	ENDIF
25	CALL goBabak3
26	ENDIF
27	ENDIF

Penjelasan *pseudocode* dari fungsi babak 2 done pada Tabel 3.24 adalah

- Variabel yang digunakan dalam fungsi ini adalah data yang diterima oleh *game server*, *time2Babak3* sebanyak 5 detik kemudian *stopTimerKeBabak3* untuk menghentikan penanda waktu.
- Pada baris 7-8 dilakukan pengisian nilai poin babak 2 untuk setiap pemain dengan cara poin total dikurangi poin dari babak 1.
- Baris 9-15 merupakan pernyataan untuk mengisi nilai untuk pemain maupun lawan dengan data yang diterima dari *game server*.

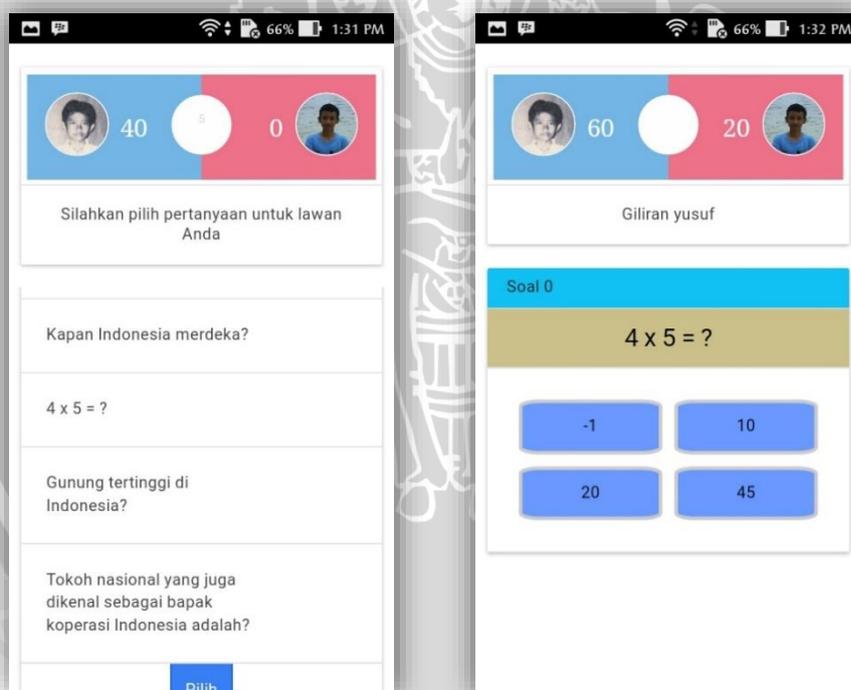
- Pada baris 16 memulai penanda waktu dengan mengisi waktu sama dengan variabel time2Babak3.
- Kemudian baris 17-27 merupakan pernyataan untuk menjalankan penanda waktu sampai waktu habis yang kemudian program akan memanggil fungsi goBabak3.

- **Implementasi Antarmuka Babak 2**

Berikut ini adalah hasil antarmuka babak 2 yang ditunjukkan pada **Gambar 3.23** dan **Gambar 3.24**.

Gambar 3.23 menunjukkan hasil antarmuka babak 2 yang menampilkan beberapa pilihan soal untuk dipilih pemain. Kemudian pada hasil antarmuka kedua menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal yang telah dipilih sebelumnya dan empat pilihan jawaban.

Gambar 3.24 menunjukkan hasil antarmuka ketika permainan babak 2 selesai. Dalam antarmuka tersebut menampilkan foto profil pemain dan poin yang didapat selama permainan babak 1 dan 2 serta poin total dari kedua babak. Selain itu pada antarmuka ini juga menampilkan penanda waktu dan informasi penjelasan mengenai permainan babak 3.



Gambar 3.23 Antarmuka Babak 2



Gambar 3.24 Antarmuka Babak 2 Selesai

3.3.3.5 Implementasi Babak 3

Pada babak 3 ini pemain akan bergantian memilih satu dari 16 kotak soal. Kemudian akan muncul soal dari kotak yang dipilih. Pemain harus menjawab soal yang muncul secara tepat dengan memilih pilihan jawaban yang tersedia. Waktu untuk memilih soal dan menjawab masing-masing yaitu selama 10 detik. Pada implementasi babak 3 ini terdapat beberapa tahap yaitu implementasi fungsi seperti fungsi *go* babak 3, *open question*, jawab babak 3, *result answered* babak 3, dan babak 3 *done*. Kemudian dilanjutkan dalam tahap implementasi antarmuka babak 3.

- **Implementasi Fungsi Go Babak 3**

Fungsi ini digunakan untuk memulai babak 3. Program akan mengirim informasi ke *game server* bahwa program telah siap. Kemudian program juga akan menerima informasi bahwa program lawan juga telah siap. Selanjutnya permainan babak 3 akan dimulai. Pada babak 3 ini tiap pemain akan mendapatkan giliran memilihkan soal untuk pemain lawan. Soal dan jawaban akan ditampilkan sesuai informasi yang diterima dari *game server*. Berikut ini adalah *pseudocode* fungsi *go* babak 3 yang ditunjukkan pada **Tabel 3.25**.

Tabel 3.25 Pseudocode Fungsi Go Babak 3

Baris	Pseudocode
1	DECLARATION
2	kotakSoal = [16]

```

3 secondFight3 = 10
4 isSelesai = false
5 stopTimer3 = defined
6 isShift = false
7
8 DESCRIPTION
9 CALL socket.emit with 'ready babak 3'
10 CALL socket.on with 'ready other babak 2' RETURN
11     CALL socket.emit with 'all ready babak 2'
12 CALL socket.on with 'grid soal' RETURN data
13     SET isSelesai to false
14     FOR i = 0, i < LENGTH data.soal, INCREASE i
15         SET kotakSoal.id to i
16         SET kotakSoal._id to data.soal[i]._id
17         SET kotakSoal.soal to data.soal[i].question
18         SET kotakSoal.pilihan to data.soal[i].choice
19         SET kotakSoal.terjawab to false
20     ENDFOR
21     SET waktu to secondFight3
22     IF stopTimer3 = defined THEN RETURN
23     ELSE
24         IF waktu > 0 THEN
25             DECREASE waktu
26         ELSE
27             IF isShift THEN
28                 SET stopTimer3 to undefined
29                 CALL jawabBabak3 with 'noFight'
30             ENDIF
31         ENDIF
32         IF data.user = detailKu.username THEN
33             SET isShift to true
34             SET textShift to 'Giliran anda silahkan
jawab'
35         ELSE
36             SET isShift to false
37             SET textShift to 'Giiran' data.user
38         ENDIF
39     ENDIF

```

Penjelasan *pseudocode* dari fungsi *go babak 3* pada **Tabel 3.25** adalah

- Variabel yang digunakan dalam fungsi ini adalah kotak soal yang berjumlah 16, `secondFight3` sebagai penanda waktu, `isSelesai` untuk mengetahui apakah permainan sudah selesai atau belum, `stopTimer3` untuk menghentikan penanda waktu, dan `isShift` untuk mengetahui giliran pada permainan.
 - Pada baris 9 program akan mengirim informasi ke *game server* bahwa pemain telah siap.
 - Pada baris 10-11 program akan menerima informasi bahwa pemain lawan telah siap kemudian program akan mengirim informasi ke *game server* bahwa semua pemain telah siap.
 - Pada baris 12 program akan menerima informasi bahwa kotak soal berjumlah 16. Kemudian pada baris 13 variabel `isSelesai` akan bernilai false. Pada baris 14-20 dilakukan perulangan untuk mengambil informasi dari *game server* yang kemudian akan ditampilkan.
 - Pada baris 21 program akan memulai penanda waktu dengan 10 detik. Dan pada baris 22-31 program akan menjalankan penanda waktu hingga waktu habis. Jika waktu habis maka penanda waktu akan berhenti dan program akan memanggil fungsi `jawabBabak3` dengan 'noFight'.
 - Pada baris 32-39 jika data *username* sama dengan *username* pemain maka `isShift` akan bernilai true dan program akan menampilkan informasi bahwa saat itu adalah giliran dari pemain begitu sebaliknya jika *username* tidak sama.
- **Implementasi Fungsi *Open Question***

Fungsi ini digunakan untuk membuka kotak soal yang dipilih oleh pemain. Ketika terbuka, *game* akan menampilkan soal beserta pilihan jawaban yang tersedia. Berikut ini adalah *pseudocode* fungsi *open question* yang ditunjukkan pada **Tabel 3.26**.

Tabel 3.26 Pseudocode Fungsi *Open Question*

Baris	Pseudocode
1	DECLARATION
2	data
3	ok
4	<code>secondAnswer = 10</code>
5	<code>mandek3 = defined</code>
6	<code>isShift = true</code>
7	
8	DESCRIPTION
9	IF <code>isShift</code> AND kotakSoal NOT terjawab THEN



10	SET noSoal to ok
11	CALL socket.emit 'open babak 3'
12	SET stopTimer3 to undefined
13	SET waktu = secondAnswer3
14	SET soal to kotakSoal.soal
15	SET jawaban to kotakSoal.pilihan
16	IF mandek3 = defined THEN RETURN
17	ELSE
18	IF waktu > 0 THEN
19	DECREASE waktu
20	ELSE
21	IF isShift THEN
22	SET mandek3 to undefined
23	CALL jawabBabak3 with 'noAnswer'
24	ENDIF
25	ENDIF
26	ENDIF
	ENDIF

Penjelasan *pseudocode* dari fungsi *open question* pada **Tabel 3.26** adalah

- Variabel yang digunakan dalam fungsi ini yaitu data yang berisi informasi dari server, ok yang berisi no soal yang dipilih oleh pemain, secondAnswer merupakan penanda waktu yaitu selama 10 detik, mandek3 yang digunakan ketika penanda waktu telah habis, dan isShift yang digunakan untuk melihat giliran menjawab.
- Pada baris 9 jika saat itu adalah giliran pemain dan kotak soal yang dipilih belum terjawab maka noSoal akan bernilai ok yang merupakan no soal yang dipilih, kemudian program akan mengirim informasi ke *game server* untuk membuka kotak soal. Penanda waktu akan berhenti dan *game* akan menampilkan soal dan pilihan jawaban yang dipilih.
- Pada baris 16-27 program akan menjalankan penanda waktu untuk menjawab soal yang dipilih. Jika waktu habis maka penanda waktu akan berhenti dan program akan memanggil fungsi jawabBabak3 dengan 'noAnswer'.

• Implementasi Fungsi Jawab Babak 3

Fungsi ini digunakan ketika pemain menjawab soal pada babak 3. Kemudian informasi jawaban akan dikirim ke *game server*. Berikut ini adalah *pseudocode* fungsi jawab babak 3 yang ditunjukkan pada **Tabel 3.26**.

Tabel 3.26 *Pseudocode* Fungsi Jawab Babak 3

Baris	<i>Pseudocode</i>
1	DECLARATION
2	userAnswer
3	kotakSoal
4	noSoal
5	mandek3 = defined
6	
7	DESCRIPTION
8	IF isShift THEN
9	SET dataJawaban.username to detailKu.username
10	SET dataJawaban.time to waktu
11	SET dataJawaban.answer to userAnswer
12	SET dataJawaban.id to kotakSoal.id
13	SET dataJawaban.no to noSoal
14	CALL socket.emit with 'answer babak 3', dataJawaban
15	SET mandek3 to undefined
16	ENDIF

Penjelasan *pseudocode* dari fungsi jawab babak 3 pada **Tabel 3.26** adalah

- Variabel yang digunakan dalam fungsi ini adalah userAnswer yang merupakan jawaban dari pemain, kotakSoal merupakan variabel dari kotak soal, noSoal merupakan no dari kotak soal yang dipilih dan mandek3 yang digunakan ketika penanda waktu berhenti.
 - Fungsi ini hanya akan berjalan jika saat itu adalah giliran pemain seperti yang ditunjukkan pada baris 8.
 - Pada baris 9-13 detail data jawaban akan diisi seperti *username*, waktu, jawaban, id, dan no soal.
 - Kemudian pada baris 14 data jawaban yang telah terisi akan dikirim ke *game server*.
 - Dan kemudian penanda waktu akan berhenti seperti ditunjukkan pada baris 15.
- **Implementasi Fungsi *Result Answered* Babak 3**

Fungsi ini digunakan untuk menampilkan informasi hasil jawaban yang dipilih oleh pemain apakah benar atau tidak. Kemudian akan dilakukan penambahan poin jika jawaban adalah benar. Berikut ini adalah *pseudocode* fungsi *result answerd* babak 3 yang ditunjukkan pada **Tabel 3.27**.

Tabel 3.27 Pseudocode Fungsi Result Answered Babak 3

Baris	Pseudocode
1	DECLARATION
2	mandek3 = defined
3	stopTimer3 = undefined
4	data
5	
6	DESCRIPTION
7	SET mandek3 to undefined
8	SET stopTimer3 to undefined
9	SET opponentAct to data.username 'menjawab' jawaban
10	IF data.isBenar THEN
11	SET opponentAct to 'Benar'
12	SET kotakSoal.terjawab to true
13	ELSE
14	SET opponentAct to 'Salah'
15	ENDIF
16	IF data.username = detailKu.username THEN
17	IF data.isBenar THEN
18	detailKu.poin = data.poin
19	ENDIF
20	ELSE
21	IF data.isBenar THEN
22	detailMusuh.poin = data.poin
23	ENDIF
24	ENDIF

Penjelasan *pseudocode* dari fungsi *result answered* babak 3 pada **Tabel 3.27** adalah

- Variabel yang digunakan dalam fungsi ini adalah *mandek3* dan *stopTimer3* yang digunakan ketika penanda waktu berhenti serta *data* yang diterima dari *game server*.
- Baris 7-8 digunakan untuk membuat penanda waktu berhenti.
- Baris 9 untuk menampilkan informasi bahwa pemain menjawab soal dengan suatu jawaban.
- Baris 10-15 merupakan pernyataan jika pilihan jawaban adalah benar maka akan menampilkan informasi pemain menjawab benar dan kotak soal akan ditandai sebagai telah terjawab begitu sebaliknya jika pilihan jawaban adalah salah.

- Baris 16-24 merupakan kondisi jika data *username* sama dengan *username* pemain dan jawaban bernilai benar maka poin akan didapat oleh pemain, namun jika *username* tidak sama maka poin akan didapat oleh pemain lawan.

- **Implementasi Fungsi Babak 3 Done**

Fungsi ini digunakan ketika permainan babak 3 sudah selesai. Kemudian *game client* menghitung poin babak 3 dari setiap pemain dan menampilkan informasi menuju halaman *home*. Berikut ini adalah *pseudocode* fungsi babak 3 *done* yang ditunjukkan pada **Tabel 3.28**.

Tabel 3.28 Pseudocode Fungsi Babak 3 Done

Baris	Pseudocode
1	DECLARATION
2	secondDone = 10
3	stopKeHome = undefined
4	
5	DESCRIPTION
6	IF data.username = detailKu.username THEN
7	SET detailKu.poinBabak3 to data[0].poin
8	SET detailMusuh.poinBabak3 to data[1].poin
9	ELSE
10	SET detailKu.poinBabak3 to data[1].poin
11	SET detailMusuh.poinBabak3 to data[0].poin
12	ENDIF
13	SET detailKu.poin to detailKu.poinBabak1 + detailKu.poinBabak2 + detailKu.poinBabak3
14	SET detailMusuh.poin to detailMusuh.poinBabak1 + detailMusuh.poinBabak2 + detailMusuh.poinBabak3
16	SET textShift to 'SELESAI'
17	SET isSelesai to true
18	IF detailKu.poin > detailMusuh.poin THEN
19	SET isMenang to 'Selamat Anda Menang'
20	ELSE
21	SET isMenang to 'Anda Kalah'
22	ENDIF
23	SET stopTimer3 to undefined
24	SET dataGame.tanggal to date now
25	SET dataGame.poinKu to detailKu.poin
26	SET dataGame.usernameMusuh to detailMusuh.username
27	SET dataGame.poinMusuh to detailMusuh.poin

28	SAVE dataGame
29	SET waktu to secondDone
30	IF stopKeHome = defined THEN RETURN
31	ELSE
32	IF waktu > 0 THEN
33	DECREASE waktu
34	ELSE
35	IF isShift THEN
36	SET stopKeHome to undefined
37	CALL socket.emit with 'game done'
38	CALL socket.emit with 'wis bar'
39	ENDIF
40	ENDIF
	ENDIF

Penjelasan *pseudocode* dari fungsi babak 3 *done* pada **Tabel 3.28** adalah

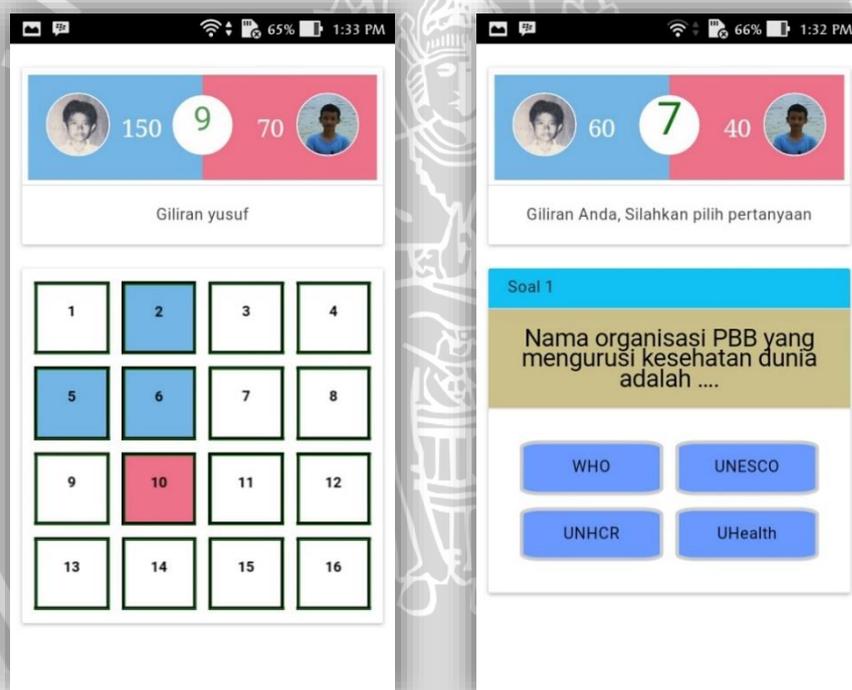
- Variabel yang digunakan dalam fungsi ini adalah *secondDone* yaitu 10 detik digunakan untuk lama penanda waktu dan *stopKeHome* digunakan ketika penanda waktu berhenti.
- Pada baris 6-12 merupakan pernyataan untuk mengisi nilai untuk pemain maupun lawan dengan data yang diterima dari *game server*.
- Pada baris 13-14 nilai total poin dari pemain dan pemain lawan adalah penjumlahan dari poin babak 1, babak 2 dan babak 3.
- Pada baris 16 program akan menampilkan informasi bahwa permainan selesai.
- Pada baris 17 *isSelesai* akan bernilai *true* menandakan bahwa permainan telah selesai.
- Baris 18-22 merupakan pernyataan jika poin dari pemain lebih besar dari lawan maka pemain dinyatakan menang, namun jika lebih kecil maka pemain dinyatakan kalah, dan pemain lawanlah yang menang.
- Baris 23 untuk menghentikan penanda waktu.
- Baris 24-27 detail data *game* akan bernilai seperti tanggal, poin, *username* pemain lawan, poin pemain lawan. Kemudian data *game* akan disimpan untuk digunakan sebagai riwayat permainan seperti ditunjukkan pada baris 28.
- Baris 29-41 menunjukkan penanda waktu untuk kembali ke *home* akan dimulai dan kemudian berjalan sampai penanda waktu berhenti dan program akan mengirim informasi bahwa permainan telah selesai.

- Implementasi Antarmuka Babak 3

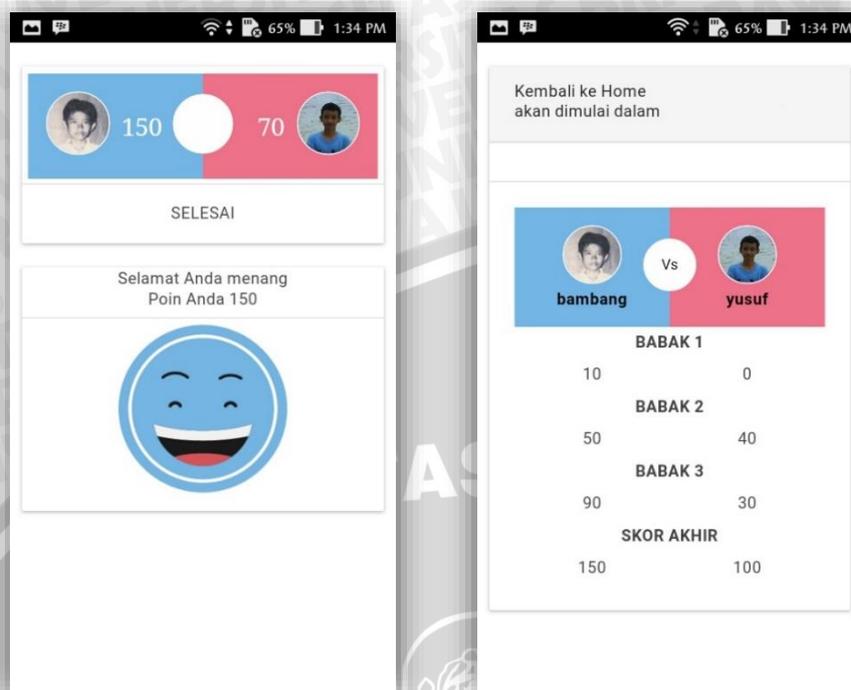
Berikut ini adalah hasil antarmuka babak 3 yang ditunjukkan pada **Gambar 3.25** dan **Gambar 3.26**.

Pada hasil antarmuka babak 3 yang pertama akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, serta 16 kotak yang berisi soal-soal yang berbeda. Untuk antarmuka yang kedua akan menampilkan foto profil dari kedua pemain beserta poin yang didapat, penanda waktu, soal yang telah dipilih sebelumnya dan 4 pilihan jawaban.

Gambar 3.26 menunjukkan hasil antarmuka ketika permainan babak 3 selesai. Pada antarmuka yang pertama menampilkan informasi bahwa pemain kalah atau menang dengan melihat total poin. Dalam antarmuka yang kedua menampilkan foto profil pemain dan poin yang didapat selama permainan babak 1, 2 dan 3 serta poin total dari ketiga babak. Selain itu pada antarmuka ini juga menampilkan penanda waktu kembali ke halaman *home*.



Gambar 3.25 Antarmuka Babak 3



Gambar 3.26 Antarmuka Babak 3 Selesai

3.3.3.6 Implementasi Halaman Riwayat

Halaman riwayat akan menampilkan riwayat permainan dari pemain. Detail riwayat seperti tanggal, *username* pemain lawan dan poin akan ditampilkan. Pada implementasi halaman riwayat ada beberapa tahap yang dilakukan yaitu implementasi fungsi *get history* dan implementasi antarmuka halaman riwayat.

- **Implementasi Fungsi *Get History***

Fungsi ini digunakan untuk membaca riwayat permainan dengan mengirimkan *request* ke *game server*. Kemudian *game server* akan merespon dengan data riwayat yang kemudian akan ditampilkan pada halaman riwayat. Berikut ini adalah *pseudocode* fungsi *get history* yang ditunjukkan pada **Tabel 3.29**.

Tabel 3.29 *Pseudocode* Fungsi *Get History*

Baris	<i>Pseudocode</i>
1	DECLARATION
2	riwayat
3	
4	DESCRIPTION
5	CALL <i>getHistoryServer</i> RETURN data
6	SET riwayat to data

Penjelasan *pseudocode* dari fungsi *get history* pada **Tabel 3.29** adalah

- Variabel yang digunakan dalam fungsi ini adalah riwayat.
 - Program akan mengirimkan *request* ke *game server* dengan memanggil fungsi *getHistoryServer*. Kemudian *game server* akan merespon dengan data seperti yang ditunjukkan pada baris 5.
 - Pada baris 6 data akan dimasukkan pada variabel riwayat yang kemudian akan tampil pada halaman riwayat.
- **Implementasi Antarmuka Riwayat**

Hasil antarmuka riwayat ditunjukkan pada **Gambar 3.27**. Hasil antarmuka riwayat permainan menampilkan daftar riwayat permainan dari pemain yang bersangkutan. Dalam daftar akan menampilkan detail *username* dari kedua pemain dan poin yang didapat serta waktu permainan.



Gambar 3.27 Hasil Antarmuka Riwayat

BAB 4 PEMBAHASAN

Bab ini akan membahas mengenai pengujian dan analisis *game multiplayer* cerdas cermat yang telah dibuat.

4.1 Pengujian unit

Pada pengujian unit digunakan metode *white box testing* dengan teknik *basis path testing*. Pada teknik *basis path testing* pengujian dilakukan dengan memodelkan algoritma pada sebuah *flow graph*, menentukan *cyclometric complexity* dan melakukan uji kasus untuk setiap *path* yang ada.

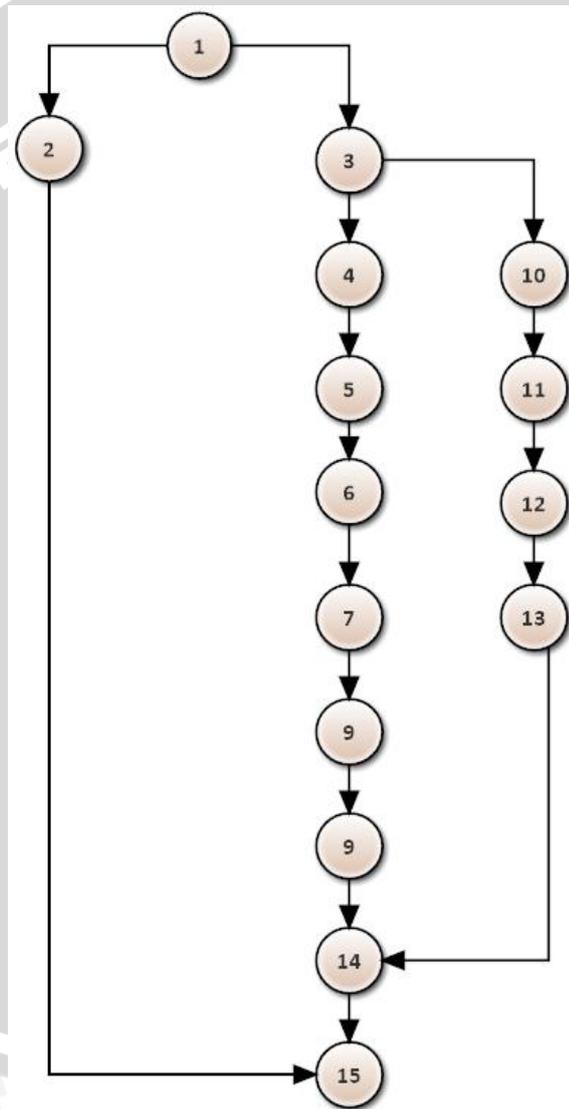
4.1.1 Pengujian unit pencarian

Unit pencarian merupakan unit algoritma pada *game server* yang digunakan untuk mempertemukan dua pemain dalam satu ruangan permainan seperti yang ditunjukkan sebelumnya pada **Tabel 3.7**. Pada pengujian unit pencarian akan dimodelkan dengan *flow graph* dengan proses pada **Tabel 4.1** dan ditunjukkan pada **Gambar 4.1**.

Tabel 4.1 Permodelan *flow graph* unit pencarian

No	Pseudocode
1	DECLARATION
2	pemainX
3	pemainY
4	socket
5	
6	DESCRIPTION
7	IF socket.username = undefined THEN.....1
8	CALL socket.emit with 'logout'.....2
9	ELSE
10	IF pemainX > pemainY THEN.....3
11	ADD socket to pemainY.....4
12	SET opponent to pemainX.shift
13	SET socket.musuh to opponent
14	SET socket.poin to 0
15	SET opponent.musuh to socket
16	SET socket.musuh.poin to 0
17	SET name room to opponent.....5
18	CALL socket.join with name_room.....6
19	SET socket.tipe to Y.....7
20	pemainY.shift.....8

21	SET pertanyaan to GET FROM database.....	9
22	ELSE	
23	ADD socket to pemainX.....	10
24	SET socket.tipe to X.....	11
25	CALL socket.join with socket.username.....	12
26	SET socket.name_room to socket.username.....	13
27	ENDIF.....	14
28	ENDIF.....	15



Gambar 4.1 Flow graph unit pencarian

Pemodelan ke dalam *flow graph* yang telah dilakukan pada unit pencarian menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas



siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 16 - 15 + 2 \\ &= 3 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 3 buah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1-2-15

Jalur 2 : 1-3-4-5-6-7-8-9-14-15

Jalur 3 : 1-3-10-11-12-13-14-15

Tabel 4.2 Tabel kasus uji unit pencarian

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Pemain tidak dalam status <i>login</i> dan mengirim permintaan permainan.	Program mengirim perintah untuk <i>logout</i>	Program mengirim perintah untuk <i>logout</i>
2	Pemain sudah <i>login</i> dan jumlah pemain tipe X kurang atau sama dengan pemain tipe Y	Pemain ditambahkan pada daftar pemain X	Pemain ditambahkan pada daftar pemain X
3	Pemain sudah <i>login</i> dan jumlah pemain X lebih dari pemain Y	Pemain ditambahkan pada daftar pemain Y	Pemain ditambahkan pada daftar pemain Y

Dari hasil pengujian unit pencarian pada setiap jalur kasus uji seperti ditunjukkan pada **Tabel 4.2** didapatkan hasil yang sesuai dengan hasil yang diharapkan sebesar 100%.

4.1.2 Pengujian Unit Penanda Waktu

Unit penanda waktu merupakan unit algoritma yang digunakan untuk menjalankan penanda waktu sebagai informasi batas waktu dalam menjawab soal pada setiap babak seperti yang ditunjukkan sebelumnya pada **Tabel 3.16**. Pada pengujian unit penanda waktu akan dimodelkan dengan *flow graph* dengan proses pada **Tabel 4.3** dan ditunjukkan pada **Gambar 4.2**.

Tabel 4.3 Permodelan *flow graph* unit penanda waktu

Baris	<i>Pseudocode</i>
1	DECLARATION

2	stopTimer1 = defined	
3	waktu = 10	
4	isShift	
5	data	
6		
7	DESCRIPTION	
8	IF waktu > 0 THEN.....	1
9	DECREASE waktu.....	2
10	ELSE	
11	IF isShift THEN.....	3
12	SET isShift to false	
13	SET dataJawaban.username to data.user	
14	SET dataJawaban.time to waktu	
15	SET dataJawaban.answer to 'noAnswer'	
16	SET dataJawaban.tipe to data.tipe.....	4
17	CALL socket.emit with 'answer babak 1', dataJawaban.....	5
18	ENDIF.....	6
19	SET stopTimer1 to undefined.....	7
20	ENDIF.....	8

Pemodelan ke dalam *flow graph* yang telah dilakukan pada unit *timer* menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 3 buah basis set dari jalur *independent*, yaitu :

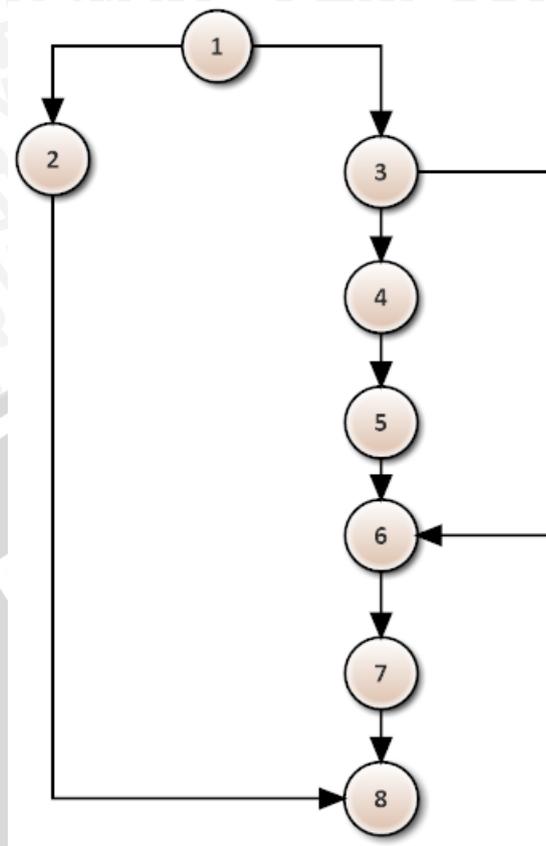
Jalur 1 : 1-2-8

Jalur 2 : 1-3-4-5-6-7-8

Jalur 3 : 1-3-6-7-8

Dari hasil pengujian unit penanda waktu pada setiap jalur kasus uji seperti ditunjukkan pada **Tabel 4.3** didapatkan hasil yang sesuai dengan hasil yang diharapkan sebesar 100%.





Gambar 4.2 Flow graph unit penanda waktu

Tabel 4.4 Tabel kasus uji unit penanda waktu

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Nilai waktu lebih dari 0 detik atau waktu masih tersedia	Penanda waktu akan berjalan dengan pengurangan nilai waktu.	Penanda waktu akan berjalan dengan pengurangan nilai waktu.
2	Waktu habis dan giliran pemain untuk menjawab.	Penanda waktu akan berhenti dan game akan mengirim informasi ke game server bahwa pemain tidak menjawab.	Penanda waktu akan berhenti dan game akan mengirim informasi ke game server bahwa pemain tidak menjawab.
3	Waktu habis dan bukan giliran pemain untuk menjawab	Penanda waktu akan berhenti dan tidak mengirim informasi	Penanda waktu akan berhenti dan tidak mengirim informasi

		apa-apa ke <i>game</i> server.	apa-apa ke <i>game</i> server.
--	--	--------------------------------	--------------------------------

4.2 Pengujian Kombinatorial

Kombinasi dalam pengujian ini menggunakan setiap *value* yang digunakan dalam pengujian akan dikombinasikan dengan setiap *value* dari parameter. Parameter-parameter yang digunakan dalam pengujian ini adalah jumlah pemain, pilih soal, dan pilih jawaban. Parameter jumlah pemain memiliki value 2 dan 4, parameter pilih soal memiliki value pilih soal dan tidak pilih, kemudian parameter pilih jawaban memiliki value jawaban benar, jawaban salah dan tidak menjawab. Kombinasi lengkap dari parameter dan *value* tersebut dijelaskan pada **Tabel 4.5**.

Tabel 4.5 Full Combinatorial

Jumlah Pemain	Pilih Soal	Pilih Jawaban
2	Pilih Soal	Jawaban Benar
2	Pilih Soal	Jawaban Salah
2	Pilih Soal	Tidak Menjawab
2	Tidak Pilih	Jawaban Benar
2	Tidak Pilih	Jawaban Salah
2	Tidak Pilih	Tidak Menjawab
4	Pilih Soal	Jawaban Benar
4	Pilih Soal	Jawaban Salah
4	Pilih Soal	Tidak Menjawab
4	Tidak Pilih	Jawaban Benar
4	Tidak Pilih	Jawaban Salah
4	Tidak Pilih	Tidak Menjawab

Dari **Tabel 4.5** akan diambil beberapa baris untuk mewakili beberapa baris dengan *value* yang sama. Beberapa baris tersebut akan menjadi *pairwise combinatorial* yang akan menjadi kasus uji.

Tabel 4.6 Pairwise Combinatorial

Jumlah Pemain	Pilih Soal	Pilih Jawaban	Status
2	Pilih Soal	Jawaban Benar	Valid
2	Tidak Pilih	Jawaban Salah	Valid
4	Pilih Soal	Tidak Menjawab	Valid

Berdasarkan hasil pengujian pada *pairwise combinatorial* seperti yang ditunjukkan pada **Tabel 4.6** didapatkan status valid sebesar 100% pada setiap kombinasi.

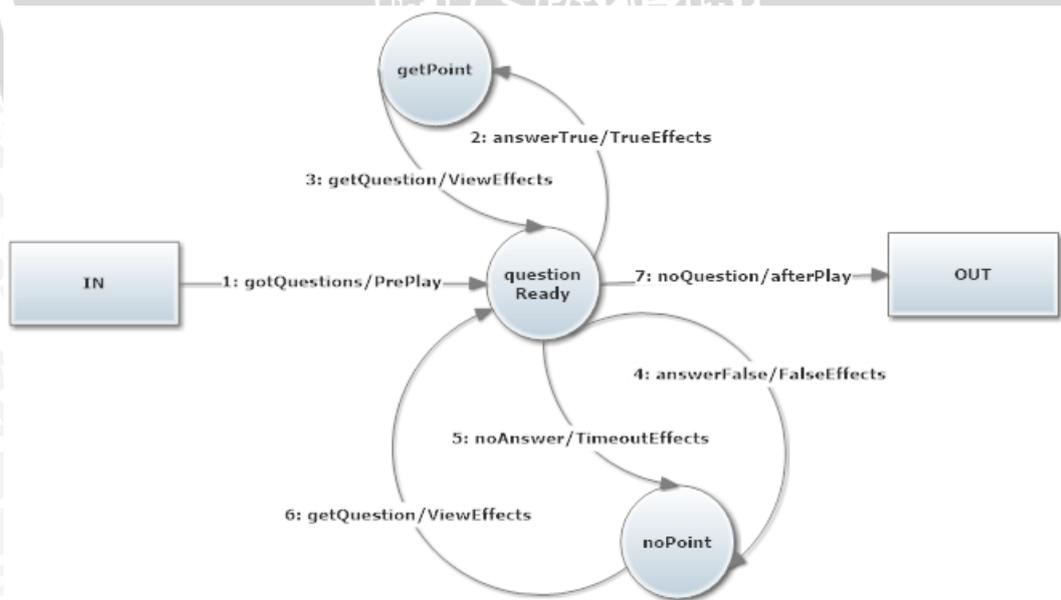
4.3 Pengujian Test Flow Diagram (TFD)

Pengujian *test flow diagram* dilakukan dengan menjalankan setiap jalur *gameplay* yang ada pada *game*. Terdapat 1 kasus uji dalam pengujian ini yaitu *gameplay* menjawab soal yang ditampilkan pada **Gambar 4.3**.

Dari **Gambar 4.3** didapatkan beberapa jalur pada alur *gameplay* diantaranya:

- 1-2-3-4-6-7
- 1-4-6-2-3-7
- 1-2-3-5-6-7
- 1-5-6-1-2-7
- 1-4-6-5-6-7
- 1-5-6-4-6-7

Pengujian dilakukan dengan menjalankan *gameplay* dari *game* pada lingkungan perangkat yang digunakan pada tahap implementasi seperti yang ditunjukkan pada **Tabel 3.5**. Pengujian dilakukan sesuai jalur yang telah terbentuk. Alur yang berjalan dengan baik mendapatkan status valid, dan alur yang tidak berjalan dengan baik mendapatkan status tidak valid. Berdasarkan hasil pengujian *test flow diagram* didapatkan status valid pada setiap jalur kasus uji sebesar 100%.



Gambar 4.3 Test Flow Diagram menjawab soal

Tabel 4.7 Hasil pengujian *test flow diagram*

Jalur	Alur	Status
1-2-3-4-6-7	Pemain menjawab pertanyaan pertama dengan benar	Valid
	Pemain salah menjawab pertanyaan berikutnya.	Valid
	Pertanyaan habis	Valid
1-4-6-2-3-7	Pemain salah menjawab pertanyaan pertama.	Valid
	Pemain menjawab pertanyaan berikutnya dengan benar.	Valid
	Pertanyaan habis	Valid
1-2-3-5-6-7	Pemain menjawab pertanyaan pertama dengan benar	Valid
	Pemain tidak menjawab pertanyaan berikutnya.	Valid
	Pertanyaan habis.	Valid
1-5-6-1-2-7	Pemain tidak menjawab pertanyaan pertama.	Valid
	Pemain menjawab pertanyaan berikutnya dengan benar.	Valid
	Pertanyaan habis.	Valid
1-4-6-5-6-7	Pemain salah menjawab pertanyaan berikutnya.	Valid
	Pemain tidak menjawab pertanyaan berikutnya.	Valid
	Pertanyaan habis.	Valid
1-5-6-4-6-7	Pemain tidak menjawab pertanyaan berikutnya.	Valid
	Pemain salah menjawab pertanyaan berikutnya.	Valid
	Pertanyaan habis.	Valid

4.4 Pengujian Terhadap Pengguna

Pengujian kepada pengguna dilakukan untuk memastikan bahwa pembuatan *game* telah sesuai dengan tujuan yang ingin dicapai yaitu *game multiplayer* cerdas cermat yang menyenangkan dan interaktif. Pengujian dilakukan dengan membuat kuesioner yang ditujukan kepada 20 mahasiswa Fakultas Ilmu Komputer UB seperti yang ditunjukkan pada **Lampiran A**. Pertanyaan-pertanyaan yang merupakan komponen dalam kuesioner ditunjukkan pada **Tabel 4.8**.

Tabel 4.8 Komponen Kuesioner

No	Isi Komponen Kuesioner
1	Apakah game ini menyenangkan untuk dimainkan secara keseluruhan?
2	Apakah tampilan dan penyajian <i>game</i> ini sudah bagus?
3	Apakah prosedur dan aturan untuk memainkan game mudah untuk dimengerti?
4	Apakah game ini lebih menyenangkan dibanding game cerdas cermat <i>single player</i> ?

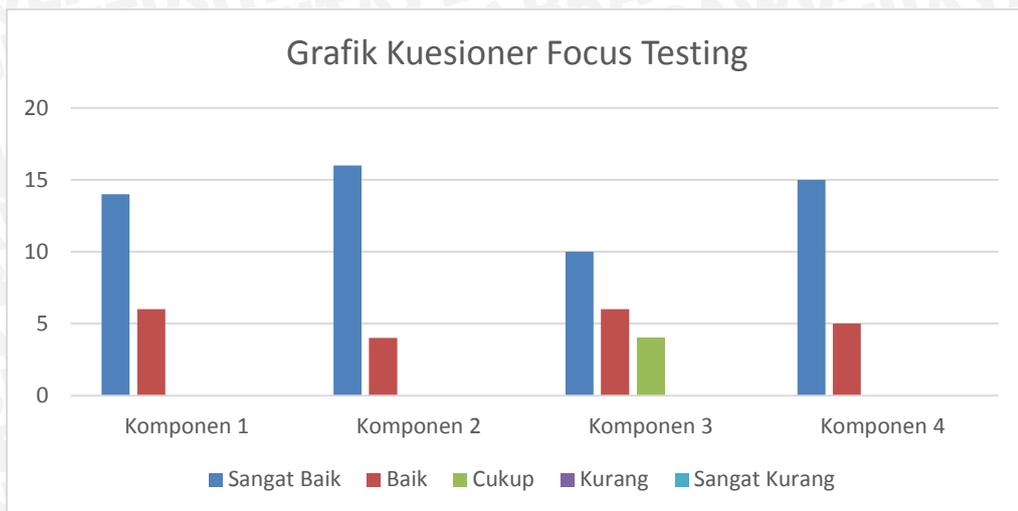
Tabel 4.9 menunjukkan hasil kuesioner dari setiap komponen pertanyaan pada setiap kriteria jawaban. Dari hasil kuesioner didapatkan informasi bahwa 70% koresponden menyatakan bahwa secara keseluruhan game sangat baik dan 30% lainnya menyatakan baik. Untuk komponen tampilan dan penyajian game 20% koresponden menyatakan bahwa sangat baik dan 80% lainnya menyatakan baik. Untuk komponen kemudahan prosedur dan aturan dalam game 50% koresponden menyatakan sangat baik, 30% menyatakan baik dan 20% menyatakan cukup. Untuk komponen perbandingan kesenangan dengan game cerdas cermat *singelplayer* 75% menyatakan sangat baik dan 25% lainnya menyatakan baik. **Tabel 4.10** menunjukkan kritik dan saran dari koresponden terhadap pengembangan *game* cerdas cermat *multiplayer*.

Tabel 4.9 Hasil kuesioner

Komponen	Hasil				
	Sangat Baik	Baik	Cukup	Kurang	Sangat Kurang
1	14	6	0	0	0
2	4	16	0	0	0
3	10	6	4	0	0
4	15	5	0	0	0

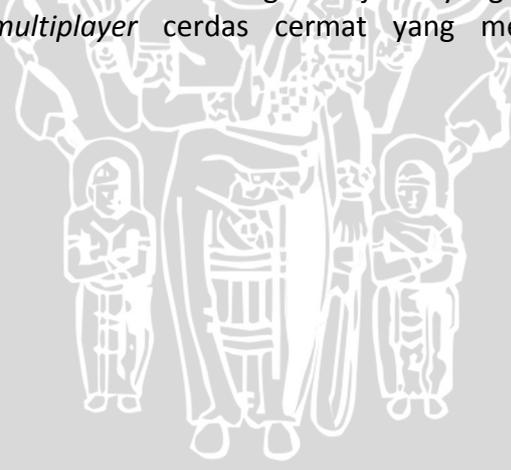
Tabel 4.10 Kritik dan saran dari koresponden

No	Kritik dan Saran
1	Sebaiknya jika jawaban salah terdapat jawaban yang benar.
2	Penanda waktu dalam penjelasan aturan permainan terlalu cepat.
3	Untuk pertanyaannya lebih baik ada kategorinya.
4	Soal-soalnya lebih baik diperbanyak.



Gambar 4.4 Grafik Kuesioner Terhadap Pengguna

Dari hasil pengisian kuesioner yang telah dilakukan kemudian didapat diagram grafik yang ditunjukkan pada **Gambar 4.4**. Dari hasil kuesioner pada komponen pertanyaan pertama mengenai apakah secara keseluruhan game menyenangkan untuk dimainkan didapatkan informasi bahwa 70% koresponden menyatakan sangat baik dan 30% lainnya menyatakan baik. Sehingga dari keseluruhan pengujian terhadap pengguna didapatkan kesimpulan bahwa perancangan dan pengembangan *game* telah sesuai dengan tujuan yang diharapkan yaitu membangun *game multiplayer* cerdas cermat yang menyenangkan bagi pemainnya.



BAB 5 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap penelitian ini yang disajikan secara terpisah.

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Game *multiplayer* Cerdas Cermat berhasil diimplementasikan menggunakan teknologi websocket.
2. Berdasarkan hasil pengujian unit menggunakan *whitebox testing*, didapatkan hasil perhitungan *cyclomatic* pada algoritma pencarian sebesar 3 dan algoritma penanda waktu sebesar 3. Dan hasil pengujian pada setiap jalur kasus uji hasil yang didapatkan yang sesuai dengan hasil yang diharapkan sebesar 100%.
3. Berdasarkan hasil pengujian kombinatorial dan *test flow diagram* pada setiap jalur kasus uji didapatkan status valid sebesar 100% sehingga dapat disimpulkan bahwa fungsional game telah valid.
4. Berdasarkan hasil kuesioner pengujian pengguna didapatkan hasil bahwa 70% koresponden menyatakan sangat baik dan 30% lainnya menyatakan baik pada komponen pertanyaan pertama. Sehingga didapatkan kesimpulan bahwa perancangan dan pengembangan *game* telah sesuai dengan tujuan yang diharapkan yaitu membangun *game multiplayer* cerdas cermat yang menyenangkan.

5.2 Saran

Saran untuk pengembangan *game multiplayer* cerdas cermat lebih lanjut antara lain:

1. Dapat dilakukan pengembangan menjadi banyak pemain yaitu lebih dari dua pada satu *room* permainan.
2. Penambahan kategori soal-soal yang akan dimainkan.
3. Dapat dilakukan pengembangan dengan dapat menantang satu pemain tertentu secara manual.
4. Penambahan fitur pemain dapat menambahkan donasi soal.

DAFTAR PUSTAKA

- Annelgrim, 2015. *Kuis*. Tersedia di : < <http://id.termwiki.com/ID/quiz> > [Diakses 15 Oktober 2015]
- Brathwaite, B. & Schreiber, I., 2009. *Challenges For Game Designers*. Boston : Course Technology.
- Cantelon, M., dkk., 2014. *Node.js in Action*. New York : Manning
- Chen, B. & Xu*, Z., 2011. *A Framework for Browser-based Multiplayer Online Games using WebGL and Websocket*. Guangzhou : South China of University.
- DB-Engines, 2015. *DB-Engines Ranking*. [online] Tersedia di <<http://db-engines.com/en/ranking>> [Diakses tanggal 16 Januari 2015].
- Iryadi, A., dkk. 2013 Implementasi Teknologi Websocket dan HTML 5 pada Game Crazy Typing dengan Genre Typing. Bina Nusantara University.
- Perdana, A. 2013. Pembangunan Aplikasi Game Cerdas Cermat Berbasis Android. Yogyakarta: Fakultas Teknologi Industri Universitas Atma Jaya
- Pressman, R., 2010. *Software Engineering. A Practitioner's Approach*. 7th ed. Mc Graw Hill.
- Pusat Bahasa, 2008. *Kamus Bahasa Indonesia*. Jakarta: Pusat Bahasa Departemen Pendidikan Nasional.
- Teixeira, P., 2013. *Hands-On Node.js*. British Coloumbia, Canada : Leanpub
- Schreiber, I. & Sohn, B., 2009. *Game Design Consept*. USA : Creative Commons Attribution.
- Schultz, C. P., Bryant, R. & Langdell, T. 2005. *Game Testing All in One*. Boston : Course Technology.
- SiswaMedia, 2015. *Cerdas Cermat*. [aplikasi Android] Google Play Store. Tersedia di: <https://play.google.com/store/apps/details?id=com.game.anak.indonesia.cerdas.cermat> [Diakses 9 Oktober 2015]
- Socket.IO, 2015. *Socket.io*. [online] Tersedia di : <<http://socket.io>> [Diakses tanggal 15 Oktober 2015]
- Oracle, 2015. *What Is a Socket?*. [online] Tersedia di : <<https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>> [Diakses tanggal 15 Oktober 2015]
- Wang, V., dkk. 2013. *The Definitive Guide to HTML5 Websocket*. Apress : New York.
- Zhang, L. & Xiaoxiao, S., 2013. *Research and Development of Real-time Monitoring System Based on Websocket Technology*. Shenyang : International Conference on Mechatronic Sciences, Electric Engineering and Computer.

LAMPIRAN A KUISIONER TERHADAP PENGGUNA

G
P₃

Kuesioner Game Cerdas Cermat Multiplayer

File apk tersedia di <http://crevion.net/cerdascermat/cc.apk>
 Screenshot tampilan game tersedia di <http://bit.ly/1lac1k>

Jawaban berupa pilihan ganda dengan angka antara 1 sampai 5.
 1 untuk sangat kurang
 2 untuk kurang
 3 untuk cukup
 4 untuk baik
 5 untuk sangat baik

Video Game Cerdas Cermat Multiplayer



Apakah game ini menyenangkan untuk dimainkan secara keseluruhan?

1
 2
 3
 4
 5

Apakah tampilan dan penyajian game ini sudah bagus?

1
 2
 3
 4
 5

Apakah prosedur dan aturan untuk memainkan game ini mudah dimengerti?

1
 2
 3
 4
 5

Apakah game ini lebih menyenangkan dibanding game cerdas cermat singleplayer?

1
 2
 3
 4
 5

Kritik, saran maupun pendapat anda mengenai game ini?

Jawaban Anda

KIRIM100%: Anda berhasil.

Jangan pernah mengirimkan sandi melalui Google Formulir.

