SISTEM MODERATOR GRUP PEMAIN DALAM *GAME ONLINE* ULAR TANGGA EDUKATIF

SKRIPSI

Oleh :
PERMATA RAHMATUL HIJJAH
NIM. 13650083

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018

LEMBAR PENGAJUAN

SISTEM MODERATOR GRUP PEMAIN DALAM *GAME ONLINE* ULAR TANGGA EDUKATIF

SKRIPSI

Diajukan kepada :

Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer

Oleh :
PERMATA RAHMATUL HIJJAH
NIM. 13650083

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2018

LEMBAR PERSETUJUAN

SISTEM MODERATOR GRUP PEMAIN DALAM *GAME ONLINE* ULAR TANGGA EDUKATIF

SKRIPSI

Oleh :

PERMATA RAHMATUL HIJJAH
NIM. 13650083

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 18 Mei 2018

Dosen Pembimbing I,	Dosen Pembimbing II,
A'la Syauqi, M.Kom	Fachrul Kurniawan, M.MT
NIP. 19771201 200801 1 007	NIP. 19771020 200912 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

SISTEM MODERATOR GRUP PEMAIN DALAM *GAME ONLINE* ULAR TANGGA EDUKATIF

SKRIPSI

Oleh :

PERMATA RAHMATUL HIJJAH
NIM. 13650083

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : 4 Juni 2018

Susunan Dewan Penguji			**Tanda Tangan**
1. Penguji Utama	:	Supriyono, M.Kom NIDT. 19841010 20160801 1 078	( )
2. Ketua	:	Fatchurrochman, M.Kom NIP. 19700731 200501 1 002	( )
3. Sekretaris	:	A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )
4. Anggota	:	Fachrul Kurniawan, M.MT NIP. 19771020 200901 1 001	( )

Mengesahkan,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Permata Rahmatul Hijjah

NIM : 13650083

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 18 Mei 2018
Yang membuat pernyataan

PERMATA RAHMATUL HIJJAH
NIM. 13650083

MOTTO

Sebaik-baik manusia adalah yang bermanfaat bagi orang lain. (HR. Ahmad)

Kesuksesan bukanlah pencapaian diri semata. Kesuksesan adalah ketika kita dapat membuat orang lain bahagia.

HALAMAN PERSEMBAHAN

Puji Syukur kepada Allah ﷻ atas segala karunia dan ilmu pengetahuan yang telah diberikan. Sholawat dan salam pada nabi pembawa syafaat, Nabi Muhammad ﷺ.

Persembahan indah yang penuh perjuangan untuk:

Ayah Khoirul Bari yang telah memberikan didikan dan kasih sayangnya yang luar biasa. Semoga Ayah selalu bahagia dan diberikan tempat terbaik disisi-Nya

Ibunda Siti Nurhayati, sosok idaman, wanita kuat dan luar biasa yang tak henti mengalirkan dukungan dan doaanya untukku

Adik-adikku tersayang Ahmad Sahid Anwar, Ahmad Fajar Ibrahim, Ahmad Najibus Syihab yang selalu menggugah semangatku

Semua guru dan dosen yang telah memberikan ilmunya kepadaku

Pembimbing skripsi, Bapak A'la Syauqi, M.Kom dan Bapak Fachrul Kurniawan M.MT yang dengan sabar membimbing dan mengarahkan

Sahabat seperjuangan yang selalu ada dan menemani; Dwi Rahayu Utami, Elfiyatul Fitriyah, Dian Fitriani, Nur Khofifah, Auliya Setianti W, Risti Nurarifah, Lin Farihah, Linda Mutiara Dewi, Novita Pratiwi, Siti Muslihaeny, Dwi Umi W, Dwi Shinta D, dan masih banyak lainnya yang belum disebut.

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakaatuh.

Segala puji bagi Allah ﷻ Tuhan semesta alam, yang telah melimpahkan rahmat serta karunia-Nya kepada penulis sehingga dapat menyelesaikan skripsi dengan judul "Sistem Moderator Grup Pemain dalam Game Ular Tangga Edukatif" dengan baik dan lancar.

Shalawat serta salam semoga tercurahkan kepada Nabi Agung Muhammad ﷺ, insan mulia yang telah menghabiskan waktu untuk menuntun umatnya dari gelapnya kekufuran kearah keselamatan hidup, menuju cahaya Islam yang terang benderang, Islam yang *rahmatan lil 'aalamin*.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, mungkin akan sulit bagi penulis untuk menyelesaikan skripsi ini.

Maka dari itu dengan segenap kerendahan hati, penulis ucapkan terimakasih kepada:

1. Bapak A'la Syauqi, M.Kom. selaku Dosen Pembimbing I yang telah memberikan bimbingan serta saran-saran dalam proses penyelesaian penelitian dan penyusunan laporan skripsi ini.
2. Bapak Fachrul Kurniawan, M.MT. selaku Dosen Pembimbing II yang telah memberikan masukan dan bimbingannya dalam proses penyusunan laporan skripsi ini.
3. Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah mendukung semua proses penelitian penulis.

4. Seluruh staf Dosen dan Admin Jurusan Teknik Informatika UIN Maliki Malang yang senantiasa memberikan ilmu dan bantuan dalam proses pembelajaran selama penulis kuliah di UIN Maliki Malang.
5. Rekan-rekan Teknik Informatika terutama angkatan 2013 yang senantiasa berbagi ilmu dalam proses perkuliahan dan berjuang bersama selama menjadi mahasiswa.
6. Para peniliti sebelumnya yang telah menulis karyanya yang terkait dengan skripsi ini.
7. Terakhir kepada seluruh pihak yang telah membantu dalam penyelesaian skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan masukan yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peniliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. *Aamiin ya rabbal 'aalamin*.

Wassalamualaikum warahmatullahi wabarakaatuh.

Malang, 18 Mei 2018

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR MODUL PROGRAM

ABSTRAK

Hijjah, Permata Rahmatul. 2018. *Sistem Moderator Grup Pemain Dalam Game Online Ular Tangga Edukatif*. Skripsi. Jurusan teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (1) A'la Syauqi, M.Kom (2) Fachrul Kurniawan, M.MT

Kata Kunci: *game online*, *multiplayer*, grup pemain, *client*-server, *HashMap*, *Vector*.

Game telah menjadi kebutuhan hiburan tersendiri pada saat ini. Salah satu jenisnya adalah *game online* yang akhir-akhir ini banyak diminati. Survei dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) menunjukkan bahwa terdapat 10% dari keseluruhan pengguna internet adalah pemain *game online* aktif. *Game online* yang dapat dimainkan oleh beberapa pemain secara langsung (*realtime*) memerlukan sebuah sistem yang dapat mengatur pengiriman paket data grup pemain. Oleh karena itu dibuatlah sistem moderator grup pemain dengan menggunakan struktur data *HashMap* dan *Vector* untuk pengaturan grup pemain pada server *game*. Sistem moderator pada server tersebut dapat berjalan dengan baik. Hasil pengujian dengan menggunakan spesifikasi perangkat server yang berbeda menunjukkan hasil yang bervariasi. Hasil pengujian server 1 dan server 2 mencapai 77,7% keberhasilan pengiriman data. Pengujian dengan menggunakan server 3 menunjukkan keberhasilan sebesar 83.3%. Dan pengujian dengan server 4, keberhasilan pengiriman data mencapai 88.8%. Dari keseluruhan pengujian yang telah dilakukan didapatkan rata-rata tingkat keberhasilan pengiriman data sebesar 83,2%. Hal ini menunjukkan bahwa semakin tinggi spesifikasi perangkat server maka semakin besar persentase keberhasilan pengiriman paket data.

ABSTRACT

Hijjah, Permata Rahmatul. 2018. *Player Group Moderator System in Educative Snakes and Ladders Online Game.* Thesis. Department of Informatics, Faculty of Science and Technology of State Islamic University of Maulana Malik Ibrahim Malang.

Supervisor: (1) A'la Syauqi, M.Kom (2) Fachrul Kurniawan, M.MT

Keywords: online game, multiplayer, player group, client-server, HashMap, Vector.

Game has become entertaiment needs at this time. One of its kind is online game that lately very popular. Survey of Indonesian Internet Service Provider Association (APJII) shows that 10% of all internet users are active online game players. Online game that can be played by multiple players in realtime need a system which can be manage delivery of player groups data packets. Therefore a player group moderator system is created using HashMap and Vector for set up player group on game server. That moderator system on the server can run well. Test result using different server device spesification show different results. Server 1 and Server 2 test results reached 77 % success of data transmission. Test using server 3 showed 83,3% success of data transmission and server 4 showed 88,8% success of data transmission. Overall, average percentage test result reach 83,2% success of data transmission. This indicates that the higher server device spesification affect the percentage of successfull data transmissions.

ملخص البحث

الحجة، فرماتا رحمة. 2018. نظام مشرف المجموعة اللاعب في لعبة على الإنترنت للثعابين والسلالم التعليمية. البحث الجامعي. قسم المعلوماتية كلية العلوم والتكنولوجيا، جامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج.

الاشراف: أعلى شوقى، الماجستير، وفخر الكورنيوان، الماجستير

الكلمات الرئيسية: لعبة على الإنترنت، متعددة اللاعبين ، لاعبين جماعيين ، *client*-server، *HashMap*، *Vector*.

الان، قد أصبحت اللعبة احتياجات الترفيه الخاصة. واحدة منها هي لعبة على الإنترنت الذى (APJII) يتطلب في الآونة الأخيرة. أظهر استطلاع من جمعية مزود خدمة الإنترنت الأندونيسية أن هناك 10٪ من جميع مستخدمي الإنترنت هم لاعبون نشطون عبر الإنترنت. تتطلب الألعاب نظامًا الذى يمكن أن يدير تسليم حزم عبر الإنترنت التي تمكن تلعب بواسطة عدة لاعبين مباشرة بيانات المجموعة من اللاعبين. ولذلك ، يتم إنشاء نظام مشرف مجموعة اللاعب باستخدام هيكل لإعدادات المجموعة اللاعبين على خادم اللعبة. يمكن أن يعمل *Vector* و *HashMap* البنيات نظام المشرف على الخادم بشكل جيد. ظهرت نتائج الاختبار باستخدام مواصفات أجهزة الخادم المختلفة نتائج متفاوتة. بلغ نتائج الاختبار الخادم 1 والخادم 2 إلى 77.7٪ بنجاح نقل البيانات. وأظهر الاختبار باستخدام الخادم 3 نجاحًا بنسبة 83.3٪. والاختبار مع الخادم 4 ، بلغ نجاح تسليم البيانات إلى 88.8 ٪. من إجمالي الاختبار حصل متوسط معدل تسليم البيانات إلى 83،2٪. هذا دل على أنه كلما كانت زادت مواصفات جهاز الخادم فكلما زادت نسبة نجاح التسليم لحزم البيانات.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game telah menjadi kebutuhan hiburan tersendiri di era modern ini. Tidak hanya digemari oleh anak-anak saja, *game* juga semakin digemari oleh semua kalangan. *Game* dengan berbagai *genre* terus bermunculan dengan menampilkan berbagai inovasi yang menarik pengguna. Di Indonesia sendiri, pemain *game* semakin meningkat dari tahun ke tahun. Apalagi dengan adanya internet, perkembangan dan penggunaan *game* semakin pesat. Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) baru-baru ini mengadakan survei pengguna internet di Indonesia. Hasilnya adalah jumlah pengguna aktif internet Indonesia sudah mencapai 63 juta orang, atau sekitar 24% dari total populasi Indonesia. Jumlah ini meningkat sekitar 8% jika dibandingkan dengan tahun lalu, dimana saat itu hanya sekitar 55 juta pengguna saja. Dari data pengguna internet aktif, diperkirakan pemain *game online* aktif Indonesia berkisar 6 jutaan, atau sekitar 10% dari jumlah pengguna internet. Pengertian aktif di sini adalah mereka yang hampir tiap hari bermain *game online* atau mengakses internet. Untuk pemain *game online* pasif, diperkirakan mencapai sekitar 15 jutaan. Perkiraan ini didapat dari data pengguna Facebook di Indonesia yang telah tembus di atas 30 juta orang, dimana 50% penggunanya pernah memainkan *game online* yang terdapat di situs jejaring sosial tersebut.

Definisi *game* jaringan atau sering disebut *game online* adalah sebuah permainan yang dimainkan di dalam suatu jaringan baik *LAN* (*Local Area*

Netwok) maupun internet. *Game online* ini biasanya dapat dimainkan oleh lebih dari satu *player* (*multiplayer*) secara bersamaan. *Game online* merupakan *game* yang banyak diminati oleh para *gamers*. Hal ini dikarenakan *game online* dirasa lebih dinamis dan *update*. Selain itu dengan adanya *game online multiplayer*, para pengguna *game* dapat bermain dan berinteraksi bersama banyak pengguna lainnya, sehingga lebih menarik.

Game online ini merupakan permainan modern yang sudah menjadi ***trend*** untuk masa sekarang dan peminatnyapun mulai dari anak-anak sampai usia dewasa. Perkembangan *game online* saat ini semakin banyak dan beragam. Konten-kontennya pun sudah tidak terfilter. Tentunya hal ini sangat mengkhawatirkan untuk anak-anak. Anak-anak merupakan kelompok yang mudah terpengaruh oleh *game online* terutama pelajar. Pelajar yang sering memainkan suatu *game online*, akan menyebabkan ia menjadi ketagihan. Ketagihannya memainkan *game online* akan berdampak baginya, terutama dari segi akademik karna ia masih dalam usia sekolah. Hal ini tentunya berdampak buruk bagi aktifitas anak, sehingga anak menjadi malas melakukan aktifitas lain seperti belajar dan bersosialisasi dengan orang lain.

Fenomena ini menjadi kehawatiran setiap orang tua terhadap anaknya ketika bermain *game*. Oleh karena itu, penulis berinisiatif untuk mengaplikasikan pembelajaran (edukatif) dalam sebuah *game* jaringan *multiplayer*. *Game* yang akan dibangun yaitu *game* Ular Tangga *multiplayer* yang disertai pembelajaran aritmatika didalamnya. *Game* ini dibangun dengan *socket programming* dan memanfaatkan jaringan *LAN* (Local Area Network) sebagai media komunikasi

antar *player*, sehingga *game* ini dapat dimainkan oleh beberapa *player* secara bersamaan.

Menurut E. Mulyasa (2011) dalam edukasi (pembelajaran), penggunaan metode yang tepat akan turut menentukan efektivitas dan efisiensi pembelajaran. Pembelajaran perlu dilakukan dengan sedikit ceramah dan metode-metode yang berpusat pada guru, serta lebih menekankan pada interaksi peserta didik. Oleh karena itu, permainan cocok untuk diterapkan dalam pembelajaran. Permainan edukatif yang seru yang dimainkan oleh beberapa orang tentu saja lebih diminati dan diterima. Dengan demikian unsur edukatif secara tidak langsung akan mudah diserap oleh pemain *game*.

Menurut Elizabeth B. Hurlock (2011) anak-anak lebih memahami dan masih suka bermain, bergerak dan menyukai permainan yang mempunyai peraturan dan bernuansa persaingan sehingga membuat pemainnya akan bermain terus-menerus tanpa memperdulikan berapa lama waktu yang dipergunakan dan usia sekolah biasanya menyukai permainan kelompok atau tim yang mana permainan ini sangat terorganisasi dam mempunyai peraturan dan bernuansa persaingan yang kuat. Pada mulanya hanya sedikit anak yang bermain, lambat laun jumlah pemain bertambah dengan meningkatnya kecakapan dan persaingan menjadi lebih kuat. Permainan yang umum dari jenis ini adalah modifikasi dari sepakbola, bola basket, kasti dan lari namun dengan berkembangnya teknologi permainan permainan tersebut mulai ditinggalkan dan kini anak-anak lebih cenderung menyukai bermain *game online*.

Dalam game *online* terdapat server yang bertugas untuk mengatur jalannya *player*. Pengaturan server disesuaikan dengan *request player* yang sedang bermain. Dalam pandangan keislaman, sitem pengaturan terdapat pada beberapa ayat Al-Quran. Salah satunya terdapat dalam Surah Al-Anbiya ayat 33.

وَهُوَ ٱلَّذِى خَلَقَ ٱلَّيۡلَ وَٱلنَّهَارَ وَٱلشَّمۡسَ وَٱلۡقَمَرَۖ كُلٌّ فِى فَلَكٍ يَسۡبَحُونَ ﴿٣٣﴾

"Dan Dialah yang telah menciptakan malam dan siang, matahari dan bulan. Masing-masing dari keduanya itu beredar di dalam garis edarnya"

Dalam Tafsir Ibnu Katsir dijelaskan, وَٱلشَّمۡسَ وَٱلۡقَمَرَ ("Matahari dan bulan,") matahari memiliki cahaya yang khusus, ruang edar sendiri, masa yang terbatas serta gerakan dan perjalanan khusus. Sedangkan bulan dengan cahaya lain, ruang edar lain, perjalanan lain dan ukuran lain. كُلٌّ فِي فَلَكٍ يَسۡبَحُونَ ("Masing-masing dari keduanya itu beredar di dalam garis edarnya,") yaitu mereka beredar. Ibnu `Abbas berkata: "Mereka beredar sebagaimana tenunan beredar di alat putarannya." Mujahid berkata: "Tenunan tidak beredar kecuali di alat putarannya dan tidak ada alat putaran kecuali dengan tenunannya. Demikian pula dengan bintang-bintang, matahari dan bulan tidak beredar kecuali dengan alat edarnya dan alat edarnya tidak berputar kecuali dengan semua itu."

Kandungan ayat tersebut membuktikan bahwa Allah ﷻ telah mengatur segala sesuatu yang ada di dunia ini dengan sangat detail. Pengaturan ini akan senantiasa tetap teraratur hingga datangnya hari kiamat. Dengan adanya pengaturan, dunia ini dapat berjalan dengan baik. Jika kita bayangkan alam ini berjalan sendiri, tanpa ada pengaturan dari Allah ﷻ, tidak ada garis edar, pastilah dunia ini sudah hancur.

Oleh karena itu, adanya pengaturan merupakan hal yang sangat penting. Supaya segala sesuatu dapat berjalan dengan baik. Sehubungan dengan ayat yang telah dijelaskan, penelitian ini berfokus pada sistem pengaturan (moderator) grup pemain pada *game online* Ular Tangga edukatif. Dengan adanya pengaturan pada server, data-data yang dikirim *client* dan server dapat diproses dengan baik oleh sistem. Penelitian ini juga menerapkan *multiplayer game* yang dapat dimainkan secara bersama dalam satu waktu (*realtime*). Sistem moderator grup pemain dalam *game* ini diatur menggunakan *HashMap* dan *Vector*, sehingga pengiriman data grup dapat terkirim pada *player* dengan tepat.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah deijelaskan, maka didapat identifikasi masalah yaitu seberapa besar tingkat keberhasilan pengiriman data oleh *active server* ke *client-client* pada suatu grup?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu untuk mengetahui tingkat keberhasilan pengiriman data oleh server ke *client-client* pada suatu grup.

1.4 Batasan Masalah

Agar penyusunan penelitian sesuai dengan tujuan yang diharapkan, maka diperlukan batasan masalah. Batasan masalah dari penelitian ini adalah:

1. *Game* Ular Tangga *multiplayer* berbasis jaringan ini dapat dijalankan pada *Personal Computer*.
2. *Game* dapat dimainkan oleh beberapa *player* yang terdapat dalam satu jaringan (*LAN*) yang sama.

3. *Game* ini hanya dapat dimainkan apabila terdapat minimal 2 *player* dalam suatu grup.

1.5 Manfaat Penelitian

Game Ular Tangga m*ultiplayer* ini diharapkan dapat dimanfaatkan oleh pengguna *game* sebagai hiburan dan sarana belajar aritmatika yang menyenangkan dan efektif. Selain itu, penelitian ini diharapkan dapat berkontribusi dalam mengatasi masalah aplikasi jaringan seperti penganan grup *multiplayer* dan optimasi paket data.

1.6 Sistematika Penyusunan

Penulisan skripsi ini tersusun dari lima bab dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penyusunan skripsi.

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan mengenai berbagai teori dan penelitian terkait yang mendasari penyusunan skripsi ini.

BAB III ANALISA DAN PERANCANGAN

Bab ini menjelaskan analisa kebutuhan sistem dan perancangan sistem yang akan dibuat.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini memaparkan pembahasan dari hasil penelitian yang sudah dilaksanakan dan pengujian terhadap sistemyang sudah dibuat.

BAB V PENUTUP

Bab ini berisi kesimpulan dari laporan penelitian dan saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 *Game*

Game merupakan kata dari bahasa inggris yang berarti permainan. *Game* merupakan aktifitas yang menyenangkan yang memiliki aturan sehingga ada yang menang dan ada yang kalah (Kamus Macmillan, 2009-2011).

2.1.1 Pengertian *Game*

Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu ditemukan oleh John von Neuman dan Oskar Morgenstern yang berisi: "Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri ataupun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi."

Pengertian *game* menurut beberapa ahli yaitu sebagai berikut:

1. Menurut Chris (1982); *Game* memiliki empat sifat. Yaitu : sistem formal tertutup (ini artinya *game* memiliki peraturan; "formal" berarti bisa didefinisikan); mengandung interaksi; melibatkan konflik; dan aman untuk dilakukan. Paling tidak dibandingkan hal yang diwakilinya (contohnya *game* peperangan *Clash of Clans* bukanlah hal yang benar-benar aman,

cedera sering terjadi tapi sebagai *game* ini adalah perwakilan abstrak dari perang, dan jelas lebih aman daripada di medan tempur).

2. Menurut Dawang Muchtar (2005); *Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius dengan tujuan *refreshing*.
3. Menurut Jim (2007); *Game* merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan
4. Menurut Tracy (2008); *Game* adalah sesuatu yang memiliki "akhir dan cara mencapainya" artinta ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya.

2.1.2 Jenis-Jenis *Game*

Game dalam perkembangannya, terdapat berbagai macam jenis *game*. *Game* berdasar Jenis *"Platform*" atau alat yang gunakan adalah sebagai berikut:

1. *Arcade games,* yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki *box* atau mesin yang memang khusus di *design* untuk jenis video *game*s tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa "masuk" dan "menikmati", seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).
2. *PC Games,* yaitu video *game* yang dimainkan menggunakan *Personal Computer*.

3. *Console games*, yaitu video *game*s yang dimainkan menggunakan *console* tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. *Handheld games*, yaitu yang dimainkan di console khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games*, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau PDA.

Selanjutnya, menurut Alana Ahmad (2016) *game* juga dipilah-pilah lagi berdasarkan jenis permainannya atau sering disebut *genre*, diantaranya adalah sebagai berikut:

1. *Role Playing Game* (*RPG*); *game* yg mengandung unsur *experience* (*EXP*) atau *leveling* dalam *gameplay* nya. *Game* RPG terbagi jadi beberapa *genre* lagi, yaitu *Action RPG*, *Turn Based RPG* (*TBRPG*)
2. *First Person Shooting* (*FPS*); *game* dimana para pemainnya saling tembak menembak.
3. *Role Playing Shooting* (*RPS*); *game* ini gabungan antara *RPG* dan *FPS* dengan berbagai macam aksi dan tidak terpatok pada tembak menembak.
4. *Third Person Shooter* (*TPS*) *game* yang mirip dengan *FPS* yaitu memiliki *gameplay* tembak menembak, hanya saja sudut pandang yg digunakan dalam *game* ini adalah sudut pandang orang ketiga.
5. *Strategy; game* yang memiliki *gameplay* untuk mengatur suatu unit atau pasukan untuk menyerang markas musuh dalam rangka memenangkan permainan.
6. *Simulation; genre* yang mementingkan realisme. Segala faktor pada *game* ini sangat diperhatikan agar semirip didunia nyata.

7. Tycoon; *game* dimana kita bertindak sebagai *businessman* yang menjalankan suatu usaha atau perusahaan dari nol.
8. *Racing; game* balapan yg memungkinkan kita untuk mengendalikan sebuah kendaraan untuk memenangkan sebuah pertandingan.
9. *Action Adventure; game* petualangan dengan salah seorang karakter yg penuh dengan penuh aksi yang akan terus ada hingga *game* tersebut tamat.
10. *Arcade; genre game* yang tidak terfokus pada cerita dan bahkan hanya dimainkan "*just for fun*".
11. *Fighting Game; genre game* bertarung.
12. *Sports; genre* bertema permainan olahraga.

2.1.3 *Game Online*

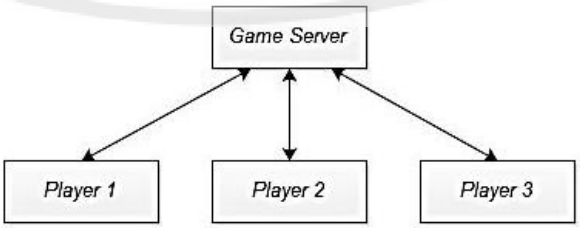
Game jaringan di Indonesia lebih dikenal dengan istilah *game online*. *Game online* berasal dari bahasa inggris. *Game* yang berarti permainan dan *online* yang terdiri dari kata *on* dan *line* dimana *on* artinya hidup dan *line* artinya saluran, sehingga online diartikan saluran yang mempunyai sambungan. Jadi dapat diartikan bahwa *game online* merupakan suatu permainan yang dapat dilakukan melalui saluran atau sambungan yang menggunakan sinyal atau jaringan. Sinyal atau jaringan yang dimaksud disini adalah berupa saluran yang menghubungkan antara *client* yang satu dengan *client* yang lain.

Menurut Andrew Rollings dan Ernest Adams (2006) *Game online* adalah jenis permainan komputer yang memanfaatkan jaringan komputer (*LAN* atau internet) sebagai medianya, *game online* lebih tepatnya disebut sebagai sebuah teknologi dibandingkan sebagai sebuah *genre* atau jenis permainan, sebuah mekanisme untuk

menghubungkan pemain bersama dibandingkan pola tertentu dalam sebuah permainan.

Berdasarkan penjelasan sebelumnya, dapat disimpulkan bahwa *game online* merupakan permainan komputer yang dimainkan dengan menggunakan jaringan komputer. *Game online* ini terdiri dari dua unsur utama, yaitu *server* dan *client*. *Client* adalah pengguna *game*, sedangkan *server* adalah penyedia layanan *game* yang bertugas mengatur jalannya komunikasi antar *client*.

Game online biasanya dimainkan oleh beberapa *player* secara bersamaan atau sering disebut *multiplayer*. *Game multiplayer* adalah *game* yang dapat dimainkan secara bersama-sama oleh dua orang pemain atau lebih sehingga para pemain bersaing satu sama lain. *Game multiplayer* dapat dibagi menjadi dua menurut jenisnya yaitu *game multiplayer* melalui server, *game* seperti *Farmville*, *Mafia Wars* dan berbagai produk lainnya berkaitan dengan situs jejaring sosial, melibatkan pemain yang saling berhubungan melalui server. Kondisi ini berarti perangkat atau pemain yang berhubungan melalui server. Diagram *game multiplayer* melalui server ditunjukkan pada Gambar 2.1. Diagram menunjukkan beberapa *player* sebagai *client* yang terhubung ke sebuah *server*. Server sebagai penghubung antar *player* satu dengan *player* lainnya.



Gambar 2.1 Diagram *Game Multiplayer* Melalui Server

Jenis *game multiplayer* yang ke dua yaitu *game multiplayer* dengan *Peer to Peer* (*P2P*). *Game P2P* dimainkan oleh *player* dengan *player* lain dalam jarak beberapa meter, biasanya menggunakan *Bluetooth*. Diagram *game multiplayer* dengan *P2P* ditunjukkan Gambar 2.2.

Gambar 2.2 Diagram *Game Multiplayer Peer to Peer*

Dari diagram pada Gambar 2.1 dan Gambar 2.2 dapat diketahui perbedaan antara *game multiplayer* melalui *server* dan *game multiplayer* dengan *P2P*. Pada *game multiplayer* melalui *server*, *server* berperan sebagai penghubung antar *player*. Sedangkan *game multiplayer P2P*, *player* langsung terhubung dengan *player* lainnya (Jeremy Kerf, 2011).

2.2 *Game* sebagai Edukasi

Game Edukasi didefinisikan sebagai aplikasi yang menggunakan karakteristik permainan video dan komputer untuk membuat menarik dan memperdalam pembelajaran (Sara De F, 2006). Jenis *game* edukasi ini lebih mengacu kepada isi dan tujuan *game* yaitu untuk memancing minat belajar anak sambil bermain. *Game* edukasi dibuat dengan tujuan khusus yaitu sebagai alat untuk membantu proses pendidikan.

Menurut E. Mulyasa (2011) dalam edukasi (pembelajaran), penggunaan metode yang tepat akan turut menentukan efektivitas dan efisiensi pembelajaran. Pembelajaran perlu dilakukan dengan sedikit ceramah dan metode-metode yang berpusat pada guru, serta lebih menekankan pada interaksi peserta didik.

Penggunaan metode yang bervariasi akan sangat membantu peserta didik dalam mencapai tujuan pembelajaran.

Permainan adalah situasi atau kondisi tertentu saat seseorang mencari kesenangan atau kepuasan melalui suatu aktivitas atau kegiatan bermain. Permainan merupakan suatu aktivitas yang bertujuan memperoleh ketrampilan tertentu dengan cara menggembirakan seseorang. Kegiatan bermain berhubungan dengan kegiatan interaksi seseorang dengan orang lainnya, barang (mainan), atau hewan yang dapat terjadi dalam konteks tertentu, baik pembelajaran (*learning*) maupun rekreasi yang bersifat menyenangkan (Fathul M & Nailul R, 2011).

Dari uraian di atas, dapat disimpulkan bahwa permainan merupakan metode yang tepat diterapkan dalam pembelajaran anak-anak. Permainan edukatif dapat dengan mudah diterima oleh anak-anak. Secara tidak langsung, mereka akan menyerap pembelajaran yang ada di dalamnya. Cara ini sangat efektif karena anak-anak telah membuka diri pada aktifitas (permainan) yang dilakukakan. Sehingga pembelajaran dapat diterima dengan cara yang menyenakan.

2.3 Permainan Ular Tangga

Ular tangga merupakan permainan papan yang sudah dikenal lama. Dalam permainan ini terdapat ular dan tangga pada papan bermainnya. Permainan ini dapat dimainkan oleh beberapa pemain. Terdapat aturan-aturan tertentu dalam memainkannya.

2.3.1 Sejarah Permainan Ular Tangga

Pada awalnya, permainan ular tangga bernama Paramapada Sopanan (*Ladder to Salvation*) yang diciptakan pada awal abad ke 2. Permainan ini

dikembangkan oleh pemuka agama Hindu untuk mengajarkan anak-anak mengenai penghargaan. Ular merepresentasikan keputusan yang buruk dan jahat, sedangkan tangga melambangkan keputusan yang bermoral dan baik. Pada tahun 1892 permainan ini masuk ke Inggris, dan pada tahun 1943 Milton Bradley mengubah nama permainan tersebut menjadi Chutes and Ladders untuk direkomendasikan di Amerika.

Permainan ular tangga di Indonesia telah menjadi bagian dari permainan tradisional, meskipun tidak ada data yang lengkap mengenai kapan munculnya permainan tersebut. Banyaknya anak-anak Indonesia yang memainkan permainan ini membuat permainan ular tangga menjadi sangat populer di masyarakat. Permainan ini ringan, sederhana, mendidik, menghibur, dan sangat interaktif ketika dimainkan bersama-sama.

2.3.2 Pengertian Permainan Ular Tangga

Ular tangga merupakan permainan yang berbentuk papan yang dimainkan oleh dua orang atau lebih. Papan permainan dibagi dalam kotak-kotak kecil untuk menjalankan bidak dan terdapat sejumlah tangga atau ular digambar di beberapa kotak yang menghubungkannya dengan kotak yang lain. Permainan ular tangga adalah permainan dimana pemain yang menempati kotak ular diharuskan turun dan pemain yang menempati kotak akan naik (Albertin C, 2012).

Permainan ular tangga merupakan permainan (*games*) adanya kontes antara pemain yang berinteraksi satu sama lain dengan mengikuti aturan-aturan tertentu untuk mencapai tujuan tertentu pula. Para pemain meletakkan bidak pada papan permainan ular tangga yang bertuliskan kata "*Start*" selanjutnya tiap

pemain mengocok dadu untuk menentukan berapa langkah yang harus dijalankan. Pemain harus melangkah sesuai dengan jumlah mata dadu yang keluar. Setelah berhenti di salah satu kotak, pemain dapat langsung menebak nama bilangan, langkah permainan diatas dilakukan oleh pemain secara bergantian hingga berakhir di kotak yang bertuliskan kata "*Finish*".

Sebagai alat bermain yang bersifat edukatif, permainan ular tangga membuat anak-anak senang bermain sekaligus mengembangkan kemampuan, mengasah logika dan meningkatkan ketrampilan mereka juga melatih anak untuk berkonsentrasi, teliti dan sabar menunggu giliran. Permainan ini cocok untuk anak-anak. Jadi melalui permainan ular tangga dapat membuat anak-anak meyakini bahwa belajar itu hal yang menyenangkan tidak membosankan dan kemampuan perkembangan anak dapat berkembang dengan baik (Sugiawati, 2013).

2.3.3 Manfaat Permainan Ular Tangga

Menurut Yasin Yusuf dan Umi Auliya permainan belajar, termasuk permainan ular tangga jika digunakan dengan bijaksana dapat menghasilkan manfaat sebagai berikut:

1. Menyingkirkan keseriusan yang menghambat;

   Harus ada keseimbangan antara suasana menyenangkan dan keseriusan. Keseriusan yang menghambat misalnya keseriusan yang disebabkan oleh ketakutan yang berlebihan.

2. Menghilangkan stres dalam lingkungan belajar;

   Banyak hal-hal kecil yang dengan mudah menyebabkan terjadinya stres dalam belajar yang harus dihindari.

3. Mengajak orang terlibat penuh;

   Dengan permainan, anak-anak benar-benar terlibat penuh dalam proses belajar.

4. Meningkatkan peroses belajar;

   Jika bersemangat dalam bermain, anak akan termotivasi untuk mengikuti proses belajar.

5. Membangun kreativitas diri;

   Dengan bermain, anak tidak terasa telah terbangun kemampuan kreativitasnya.

6. Mencapai tujuan dengan ketidaksadaran;

   Pada keadaan asyik bermain, unsur materi belajar yang telah dimasukkan dalam permainan akan masuk tanpa disadari.

7. Meraih makna belajar melalui pengalaman;

   Suatu saat ketika ditanya materi yang dipelajari, anak akan mampu mengingatnya kembali karena kebermaknaan proses belajar yang dilakukannya.

8. Memfokuskan siswa sebagai subjek belajar;

   Anak akan semakin fokus dengan materi yang dimasukkan ke dalam permainan. Hal ini disebabkan oleh rasa tertantang, motivasi, dan semangat yang tinggi.

2.4 Jaringan Komputer

Seiring dengan perkembangan teknologi komputer dan komunikasi, suatu model komputer tunggal yang melayani seluruh tugas-tugas komputasi suatu organisasi kini telah berganti dengan sekumpulan komputer yang terpisah-pisah

akan tetapi saling berhubungan dalam melaksanakan tugasnya, sistem seperti ini disebut jaringan komputer (*computer network*). Sistem jaringan komputer adalah gabungan atau kumpulan dari beberapa komputer yang dapat diakses secara bersama-sama, dan dapat berhubungan dengan komputer induk sistem lainnya.

Jaringan komputer diklasifikasikan berdasarkan jarak jangkaunnya. Berikut tabel yang menampilkan klasifikasi sistem multiprosesor berdasarkan ukuran-ukuran fisiknya.

Tabel 2.1 Klasifikasi Prosesor Interkoneksi Berdasarkan Jarak

Jarak antar prosesor	Prosesor di tempat yang sama	Contoh
0,1 m	Papan rangkaian	*Data Flow machine*
1 m	Sistem	*Multicomputer*
10 m	Ruangan	*Local Area Network*
100 m	Gedung	
1 km	Kampus	
10 km	Kota	*Metropolitan Area Network*
100 km	Negara	*Wide Area Network*
1.000 km	Benua	
10.000 km	Planet	*Internet*

Dari *Tabel 2.1* terlihat pada bagian paling atas adalah *dataflow machine*, komputer-komputer yang sangat paralel yang memiliki beberapa unit fungsi yang semuanya bekerja untuk program yang sama. Kemudian *multicomputer*, sistem yang berkomunikasi dengan cara mengirim pesan-pesannya melalui bus pendek dan sangat cepat. Setelah kelas *multicomputer* adalah jaringan sejati, komputer-komputer yang bekomunikasi dengan cara bertukar data/pesan melalui kabel yang lebih panjang. Jaringan seperti ini dapat dibagi menjadi *Local Area Network* (*LAN*), *Metropolitan Area Network* (*MAN*), dan *Wide Area Network* (*WAN*).

Akhirnya, koneksi antara dua jaringan atau lebih disebut *internetwork*. Internet merupakan salah satu contoh yang terkenal dari suatu *internetwork*.

2.4.1 *LAN (Local Area Netwok)*

LAN juga merupakan network atau jaringan sejumlah sistem komputer yang lokasinya terbatas dalam satu gedung, satu komplek gedung atau suatu kampus dan tidak menggunakan media fasilitas komunukasi umum seperti telepon, melainkan pemilik dan pengelola media komunikasinya adalah pemilik *LAN* itu sendiri.

Menurut Jogiyanto (2004) *LAN* adalah suatu network yang terbatas dalam jarak atau area setempat. Network ini banyak digunkan satu perusahan yang menghubungkan antara departemen-departemen dalam satu gedung. Sedangkan menurut Husni (2003) terdapat beberapa komponen dasar yang biasanya membentuk suatu *LAN* yaitu sebagai berikut.

1. *Host* atau *node*, yaitu sistem komputer yang berfungsi sebagai sumber atau penerima dari data yang dikirimkan. *Node* ini dapat berupa:
   a. *Server* : komputer tempat penyimpanan data dan program-program aplikasi yang digunakan dalam jaringan,
   b. *Client* : komputer yang dapat mengakses sumber daya (berupa data dan program aplikasi) yang ada pada *server*,
   c. *Shared pheriperal* : peralatan-peralatan yang terhubung dan digunakan dalam jaringan (misalnya, printer, scanner, harddisk, modem, dan lain-lain).

2. *Link*, adalah media komunikasi yang menghubungkan antara node yang satu dengan node lainnya. Media ini dapat berupa saluran transmisi kabel dan tanpa kabel.
3. *Software* (perangkat lunak), yaitu program yang mengatur dan mengelola jaringan secara keseluruhan. Termasuk di dalamnya sistem operasi jaringan yang berfungsi sebagai pengatur komunikasi data dan periferal dalam jaringan.

LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama *resource* (misalnya *printer*, *scanner*) dan saling bertukar informasi. *LAN* dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik: ukuran, teknologi transmisi dan topologinya.

LAN mempunyai ukuran yang terbatas, yang berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Dengan mengetahui keterbatasnnya, menyebabkan adanya kemungkinan untuk menggunakan jenis desain tertentu. Hal ini juga memudahkan manajemen jaringan.

2.4.2 *IP Address*

IP Address adalah suatu deretan bilangan unik yang mengidentifikasi suatu *host* atau komputer di internet. Bilangan-bilangan tersebut biasanya ditampilkan dalam kelompok-kelompok yang dipisahkan oleh titik, seperti 202.145.34.57. Semua sumber daya di internet harus mempunyai IP *Address* atau sama sekali tidak dapat hadir di internet. IP (*Internet Protocol)* mendefinikan

bagaimana informasi dilewatkan antara satu sistem dengan sistem yang lain di internet (Husni, 2003).

2.4.3 *Client*-Server

Definisi *client* server menurut Budhi Irawan (2005), server adalah komputer *database* yang berada di pusat, dimana informasinya dapat digunakan bersama-sama oleh beberapa user yang menjalankan aplikasi di dalam komputer lokalnya yang disebut dengan c*lient*. Server merupakan sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. Secara fisik, sebuah server dapat berupa komputer *mainframe*, mini-komputer, *workstation*, PC atau peranti lain seperti *printer*. Server tidak harus berupa sistem fisik, tetapi juga suatu proses. *Client* mempunyai kemampuan untuk melakukan pemrosesan sendiri. Ketika sebuah *client* meminta suatu data ke server, server akan segera menanggapinya dengan memberikan data yang diminta ke client bersangkutan. Setelah diterima, *client* segera melakukan pemprosesan.

Sebuah file server menjadi jantung dari keseluruhan sistem, memungkinkan unutk mengakses sumber daya, dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber sumber daya yang ada pada file server. Model hubungan komponen yang ada di jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada file server.

Model komputasi yang berbasis *client* server mulai banyak diterapkan pada sistem informasi. Dengan menggunakan arsitektur ini , sistem informasi

dapat digunakan dan dibangun dengan perangkat lunak client server yang bermacam-macam dan berbeda-beda. Contoh implementasi *Client* Server yaitu:

1. Aplikasi pesan, misalnya surat elektronis (*email*)
2. Penyebaran basis data pada beberapa jaringan komputer
3. Memungkinkan berbagi berkas atau periferal atau pengaksesan komputer melalui jarak jauh
4. Pemrosesan aplikasi yang intensif dengan suatu pekerjaan (*job*) dibagi menjadi tugas-tugas (*task*) yang masing-masing dilaksanakan pada komputer yang berbeda.

Model komputasi yang berbasis client server memiliki beberapa fitur dan keuntungan seperti pada Tabel 2.2.

Tabel 2.2 Fitur dan Keuntungan Model Komputasi *Client* Server

Fitur	Keuntungan
Jaringan mesin-mesin yang kecil tetapi berdaya guna	Jika sebuah mesin macet, bisnis tetap berjalan
Kumpulan komputer dengan ribuan MIPS (million instruction per second)	Sistem memberikan kekuatan dalam melaksanakan suatu tugas tanpa memonopoli sumber-sumber daya. Pemakai akhir diberi hak untuk bekerja secara local
Beberapa workstation sangat handal seperti mainframe, tetapi dengan biaya 90% lebih rendah	Menawarkan keluwesan untuk melakukan pembelian pada hal-hal lain atau untuk meningkatkan keuntungan
Sistem terbuka	Bebas memilih perangkat keras, perangkat lunak, dan layanan dari berbagai vendor
Sistem tumbuh dengan mudah dan dapat diperluas secara tak terbatas	Mudah untuk memperbaharui system
Lingkungan operasi client yang bersifat individual	Dapat mencampur dan mencocokkan platform komputer yang gsesuai dengan kebutuhan masing-masing departemen dan pemakai

2.4.4 *Socket*

Biasanya, sebuah aplikasi sebagai server dijalankan dalam sebuah komputer dengan *socket* menggunakan port tertentu. Ketika dijalankan, server hanya menunggu jika ada permintaan dari *client* yg ditujukan ke *socket* milik server. Untuk membuat koneksi ke server, *client* harus sudah mengetahui socket dari server. Dengan kata lain, client harus sudah mengetahui alamat *IP* (atau bisa juga dalam bentuk *hostname* seperti example.com) dan *port* yg dituju dalam pengiriman *request*. Dalam *request* tersebut, *client* menyertakan alamat *IP* dan port milik *client* (yaitu sumber *request* yg biasanya sudah ditentukan oleh sistem operasi milik *client*) dan alamat *IP* dan *port* yg dituju (milik server). Bersiap-siap menerima *request* dari *client* dengan *IP* mana saja dan *port* apa saja, selagi *request* tersebut diarahkan ke alamat *IP* dan *port* yg digunakan oleh aplikasi *server*.

Socket adalah *interface* pada jaringan yang menjadi titik komunikasi antarmesin pada *Internet Protocol*, dan tentunya tanpa komunikasi ini tidak akan ada pertukaran data dan informasi jaringan. *Socket Address* adalah kombinasi dari alamat *IP* dan sebuah *port* (yang berhubungan dengan proses suatu aplikasi) yang menjadi satu identitas. Sebuah internet *socket* memiliki ciri-ciri khusus yang dibentuk oleh kombinasi unik dari:

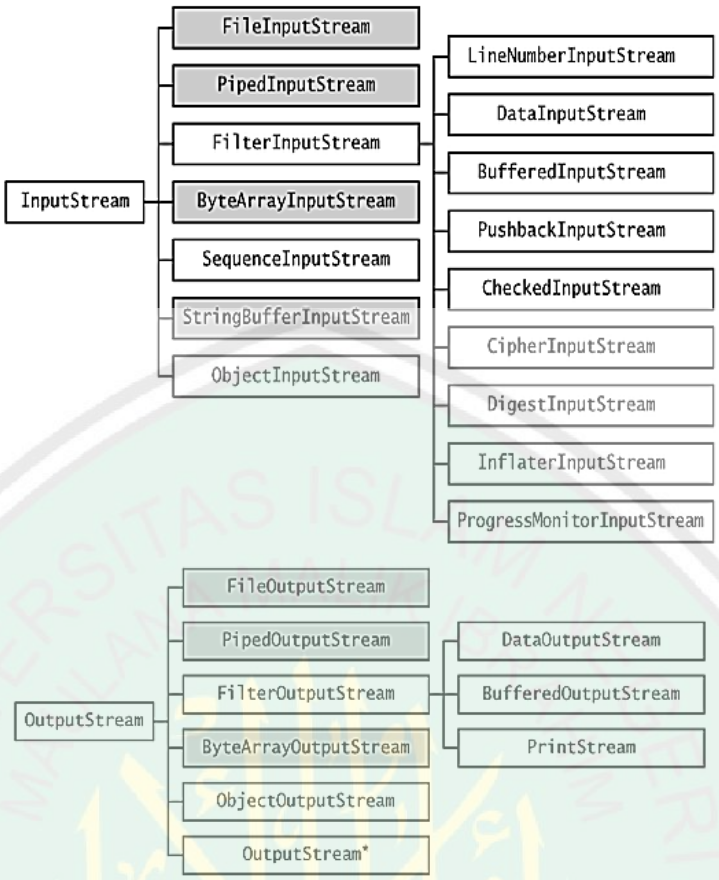
1. *Protocol* (*TCP, UDP*)
2. Alamat *socket* lokal (*Local Socket Address*) yang terdiri dari *Local IP address* dan nomor *port*
3. *Remote socket address* (Hanya untuk *TCP socket*)

2.4.5 Data *Streams*

Pada pemrograman jaringan, komunikasi melalui jaringan, file, ataupun aplikasi, direpresentasikan dalam Java dengan menggunakan *Stream*. Konsep *Stream* sangat penting ketika membahas aplikasi jaringan. Hampir semua *network communication* (kecuali komunikasi UDP) dibangun menggunakan *Steams*.

Stream adalah aliran data yang berupa aliran data *byte* atau karakter dari *device input* ke *device output* pada saat program di eksekusi. *Stream* juga dapat diartikan sebagai representasi abstrak dari *input* dan *output device* dimana aliran data *bytes* akan ditransfer seperti transfer file ke dalam harddisk. Operasi *input* dan *output* pada java menggunakan konsep aliran data, yaitu aliran data dari *device output* dengan memanfaakan *method* println() pada objek System.out, dan *device input* pada objek System.in.

Proses *stream* untuk membaca diimplementasikan dengan memanggil kelas turunan dari superclass java.io.InputStream. Sedangkan proses *stream* untuk menulis diimplementasikan dengan memanggil kelas turunan dari *superclass* java.io.OutputStream. *InputStream* dan *OutputStream* merupakan kelas abstrak yang tidak bisa digunakan secara langsung, namun bisa menggunakan *subclass* dari masing-masing sesuai kebutuhan. Kelas turunan dari *InputStream* dan *OutputStream* ditunjukkan pada bagan Gambar 2.3.

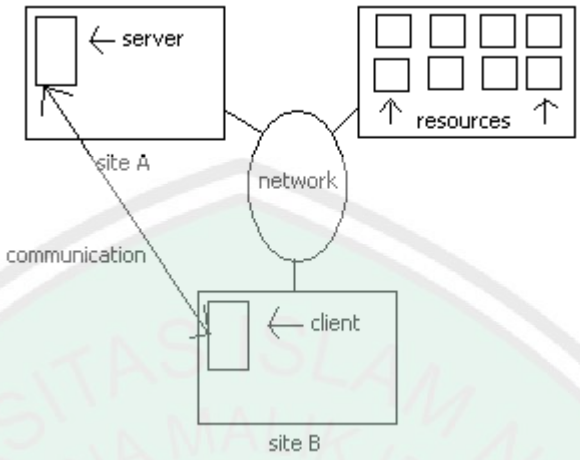


Gambar 2.3 Turunan *Input Stream* dan *Output Stream*

2.5 Sistem Moderator dan Sistem Terdistribusi

Sistem terdistribusi didefinisikan sebagai suatu kesatuan dari elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk mendistribusikan data (Marseleno, 2003). Sistem terdistribusi adalah sekumpulan prosesor yang tidak saling berbagi memori atau *clock* dan terhubung melalui jaringan komunikasi yang bervariasi, yaitu melalui *Local Area Network* ataupun melalui *Wide Area Network*. Prosesor dalam sistem terdistribusi bervariasi, dapat berupa small *microprocessor*, *workstation*, *minicomputer*, dan lain sebagainya. Contoh dari sistem terdistribusi yaitu internet, *mobile computing*, sistem operasi

bank, *GPS* (*Global Positioning System*), dan masih banyak yang lainnya. Gambar 2.4 menunjukkan ilustrasi struktur sistem terdistribusi.



Gambar 2.4 Ilustrasi Struktur Sistem Terdistribusi

Karakteristik sistem terdistribusi adalah sebagai berikut:

1. *Resource Access and Sharing*; kemampuan menggunakan *hardware*, *software* atau data dimanapun dan kapanpun.
2. *Openness*; kemampuan sebuah sistem dalam mengembangkan fleksibilitas terhadap peningkatan kinerja sebuah sistem.
3. *Concurrency*; semua proses dalam sistem terdistribusi dilakukan secara bersama-sama
4. *Scalability*; sebuah sistem terdistribusi harus dapat ditingkatkan kinerjanya tanpa mengubah komponen-komponen di dalamnya
5. *Fault Tolerance*; kemampuan sistem untuk menangani kesalahan.
6. *Transparency*; keterbukaan sistem untuk mempermudah bagi pengembang dan pemelihara sistem

Dalam pengaplikasiannya, sistem terdistribusi memiliki beberapa arsitektur atau pemodelan. Arsitektur yang paling populer dan sering digunakan adalah model *Client*-Server. Server merupakan komputer yang memberikan

service ke *client*, sedangkan *client* adalah komputer yang mengakses beberapa service yang ada di server.

Dalam penelitian ini, menerapkan arsitektur *Client*-Server. Server dalam sistem ini berperan aktif dalam memproses paket data grup pemain atau *request* dari *client*. Oleh karena itu di dalam server terdapat sistem moderator. Menurut Munif Chatib (2011), moderator didefinisikan sebagai pemimpin jalannya kegiatan agar tepat dan terarah. Sedangkan sistem moderator adalah sistem yang bertugas untuk memandu, mengatur, dan mengawasi jalannya suatu proses atau kegiatan. Sistem moderator memiliki tugas-tugas tertentu agar proses tetap berjalan dengan baik dan benar sesuai dengan aturan. Sistem ini merupakan inti dari proses itu sendiri. Hal ini dikarenakan sistem moderator memiliki pengaruh yang sangat besar terhadap keberhasilan jalannya proses/kegiatan. Sistem moderator berjalan dengan aturan-aturan tertentu sesuai dengan tujuan yang ingin dicapai.

Sistem moderator pada server bertugas mengatur distribusi paket data grup pemain. Sistem ini mengatur paket data yang dikirim oleh *client* (*player*) ke server. Paket data tersebut diproses oleh server dan akhirnya server mem-*broadcast* paket data tersebut ke *client* (*player*) lain dalam grup yang sama. Dalam pemrosesan paket data, sistem moderator pada server menggunakan struktur data *HashMap* dan *Vector*. *HashMap* digunakan sebagai penyimpanan seluruh *username* dan *socket address player* sedangkan *Vector* untuk pengaturan pengiriman data paket grup pemain.

2.6 *Vector* dan *HashMap*

Vector adalah sebuah *class* yang diturunkan dari *interface collection*, yaitu sebuah *interface* yang digunakan untuk pengolahan data yang bersifat seperti *array* dinamis, yakni *array* yang ukurannya secara dinamis dapat membesar ketika data yang dimasukkan melebihi daya tampung. Setiap metode dalam *Vector* diberi *keyword* "*synchronized*", sehingga ketika dieksekusi dalam sebuah *thread*, maka tak akan terjadi kemacetan *thread*. *Array* memiliki daya tampung terbatas dari yang dideklarasikan (daya tampung tidak dinamis). *ArrayList* memiliki fungsi yang sama dengan *Vector*, namun metode yang terdapat pada *ArrayList* tidak diberi *keyword* "*sycrhonized*", sehingga ketika dieksekusi dalam sebuah *thread* dapat mengakibatkan *unsafe thread* dan terjadinya tubrukan *thread* atau kemacetan saat eksekusi.

HashMap adalah *class* implementasi dar *Map*, *Map* itu sendiri adalah *interface* yang memiliki fungsi untuk memetakan nilai dengan *key* unik. *HashMap* berfungsi sebagai *memory record management*, dimana setiap *record* dapat disimpan dalam sebuah *Map*. kemudian setiap *Map* diletakkan pada *vektor*, *list* atau *set* yang masih turunan dari *collection*. *HashMap* sangat baik untuk menghandle *resultset* dari *query*. Hal ini memungkinkan waktu eksekusi operasi dasar, seperti get() dan put(), tetap konstan bahkan untuk set yang besar.

2.6 Studi Literatur *Game Multiplayer*

Peneliatian mengenai *game multiplayer* jaringan sebelumnya telah dilakukan oleh M. Fikri (2013) pada penelitiannya yang berjudul Rancang Bangun *Game Batleship Multiplayer* pada jaringan *LAN*. Dalam penelitiannya, *LAN* digunakan sebagai media penghubung antara pemain *game* yang bertindak

sebagai *server* dan *client*. *Game Battleship Multiplayer* ini dirancang dengan menggunakan *OOA* (*Object Oriented Analisys*) dan *OOD* (*Object Oriented Design*). Namun *game* ini memiliki keterbatasan hanya dapat menampung dua *player* saja. Satu pemain bertindak sebagai *server* dan pemain lainnya sebagai *client*. Dalam *game Batleship* ini belum terdapat *grup* yang dapat menampung beberapa *player*.

Penelitian *game* Ular Tangga berbasis jaringan sebelumnya dilakukan oleh Marta Ika Wijayanti (2013). *Game* ini juga menggunakan *LAN* sebagai media penghubung antar *player*. Pada penelitian ini, jumlah *player* yang bertindak sebagai *client* lebih meningkat dibandingkan pada penelitian *Game Batleship Multiplayer* yang telah dibahas sebelumnya. *Game* ini dapat dimainkan oleh 2 sampai 4 *player* secara langsung. Konsep *server* tunggal yang terkoneksi dengan beberapa *client* telah diterapkan pada penelitian ini. Namun *game* ini hanya sebatas *game* Ular Tangga seperti pada umumnya (tidak ada penambahan aturan/ variasi permainan). Selain itu, pada *game* ini juga belum ada *grup*.

M. Fadli Syahputra, dkk (2013) juga telah melakukan penelitiannya mengenai *multiplayer game*. *Game* yang dibangun adalah game kartu "Truf Gembira". Dalam perancangannya, *game* ini dapat dimainkan oleh 4 *player* secara *real time*. *LAN* digunakan sebagai media koneksi antar player. Untuk menjalankan *game* ini terdapat 4 *player* dimana salah satu *player* bertindak sebagai *server* sekaligus *client* dan 3 *player* lainnya bertindak sebagai *client*. *Game* ini telah diterapkan pada *platform mobile*. Hanya saja dalam *game* ini belum terdapat kontrol jumlah *player* yang dapat memainkan *game*. Harus ada 4 *player* untuk

memainkan *game*, sehingga jika *player* kurang dari 4, *game* ini tidak dapat dimainkan.

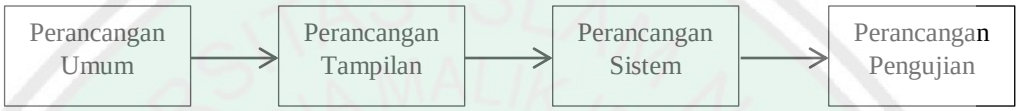
Penelitian yang mengusung *game* jaringan lokal lainnya yaitu game *multiplayer* Gobak Sodor oleh Belia, dkk (2016). Game ini diusung dari *game* tradisional dari Jawa Tengah yang dapat dimainkan di *Personal Computer* oleh lebih dari satu *player*. *Game* ini pada dasarnya sudah cukup baik. *Game* ini dapat dimainkan sendiri dan bersama maksimal oleh 5 *player* secara *realtime*. Namun masih terdapat banyak bug pada menu main bersama (*multiplayer*) dan ketika jumlah *player* ada lebih dari 2, maka pergerakan musuh akan menjadi lambat atau berhenti.

Penelitian *game multiplayer* baru-baru ini yaitu *Game FPS (First Person Shooter)* dengan menggunakan *Multiplayer Game* (Widharma, 2016). Peneliti membuat simulasi *game multiplayer* dengan 4 *player* yang bekerja sama untuk menghadapi *NPC* (*Non-Player Character*) atau musuh dalam *game* tersebut. Game ini sudah cukup baik, dilengkapi dengan *Artifical Intelligent* pada *NPC*. Namun *game* ini masih dalam tahap simulasi saja, belum dikembangkan dengan grafis yang lebih menarik.

BAB III

ANALISA DAN PERANCANGAN

Pada bab ini dibahas mengenai beberapa tahapan rancangan penelitian yang dilakukan. Perancangan penelitian meliputi perancangan umum, perancangan tampilan, perancangan sistem, dan perancangan pengujian. Tahapan perancangan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan Perancangan

3.1 Perancangan Umum

Pada subbab ini dijelaskan mengenai game yang dibangun serta alur yang berjalan pada saat memainkan game (*gameplay*).

3.1.1 Deskripsi Umum *Game*

Game edukasi ular tangga yang akan dibangun merupakan *game* ular tangga yang dilengkapi dengan soal-soal aritmatika didalamnya. Soal-soal aritmatika disini meliputi penjumlahan, pengurangan, perkalian, dan pembagian serta soal campuran dari beberapa operator. Sasaran pengguna *game* edukasi yaitu anak-anak usia SD/MI. *Game* ini dapat dimainkan oleh dua sampai empat *player* (*multiplayer*). *Game* ini dijalankan pada desktop dengan menggunakan jaringan lokal. Jaringan lokal ini digunakan untuk koneksi antar *player*. Dalam *game* ini terdapat *client* dan *server*. *Client* disini adalah *player*. *Server* bertugas untuk koneksi *player* satu dengan yang lainnya dan mengatur aliran pengiriman data.

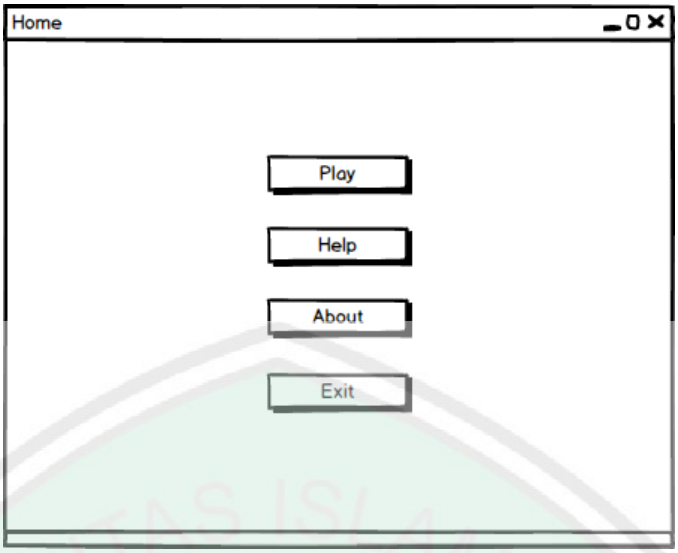
3.1.2 *Gameplay*

Game ular tangga ini terdiri dari 100 blok atau kotak dan didalamnya terdapat beberapa ular dan tangga. *Game* ini dapat dimainkan oleh dua sampai empat *player*. Alur permainan *game* ular tangga ini sama seperti *game* ular tangga pada umumnya, yaitu:

1. Pertama *player* mengacak dadu, kemudian *player* tersebut berjalan sejumlah angka dadu yang didapatkan.
2. Apabila *player* berhenti di ujung ular atau tangga, maka secara otomatis akan muncul soal aritmatika yang harus dijawab.
3. Apabila ia dapat menjawab soal tersebut dengan benar, maka *player* dapat naik tangga atau tidak menuruni ular. Tetapi jika jawaban *player* salah, ia tidak bisa naik tangga atu harus menurun jika ada ular.
4. *Player* terus bermain secara bergantian sampai salah satu *player* memenangkan *game* di kotak 100.

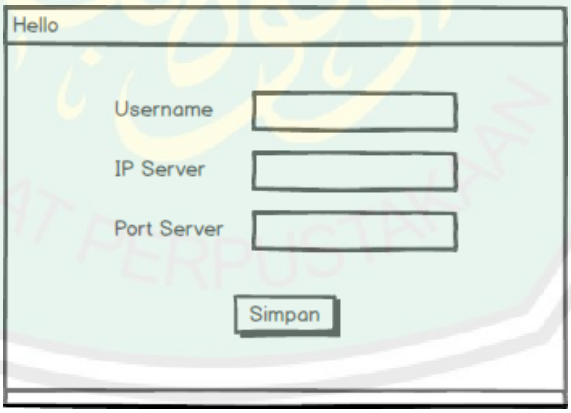
3.2 Perancangan Tampilan Game

Game ini terdiri dari beberapa tampilan yang didalamnya terdapat beberapa fitur. Ketika pertama membuka game biasanya terdapat tampilan *splash screen* atau halaman yang muncul sesaat sebagai pengantar ke halaman utama. Gambar 3.2 menunjukkan tampilan halaman utama. Pada halaman utama, terdapat beberapa menu seperti pada Gambar 3.2, yaitu menu *Play* untuk memulai *game*, Help untuk bantuan mengenai cara memainkan game, *About* berisi tentang informasi *game*, dan menu *Exit* untuk keluar dari game.



Gambar 3.2 Rancangan Tampilan Menu Utama

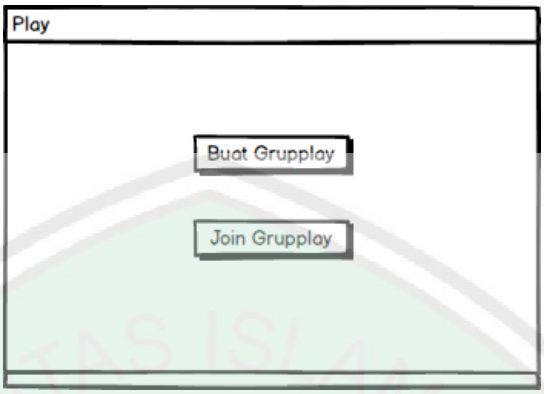
Selanjutnya adalah tampilan ketika memulai *game*. Ketika *player* memilih menu *play* akan muncul form login player seperti pada Gambar 3.3. *Player* mengisikan *username* yang digunakan sebagai identitas *player* ketika bermain *game*, dan *IP* serta *port server* yang digunakan untuk menghubungkan koneksi *player* pada *server game*.



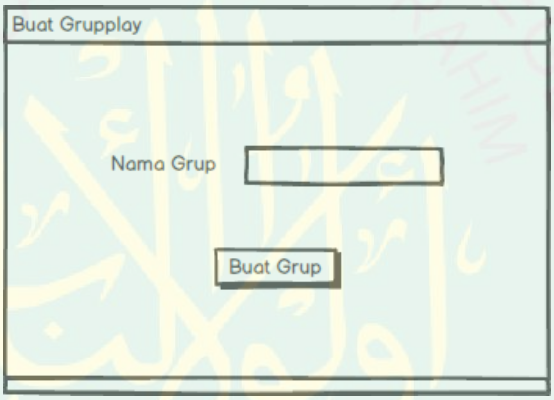
Gambar 3.3 Rancangan Form *Login Player*

Setelah *player* mengisi *form login*, kemudian *player* membuat grup atau bergabung dengan grup yang sudah ada sebelumnya. Rancangan tampilannya

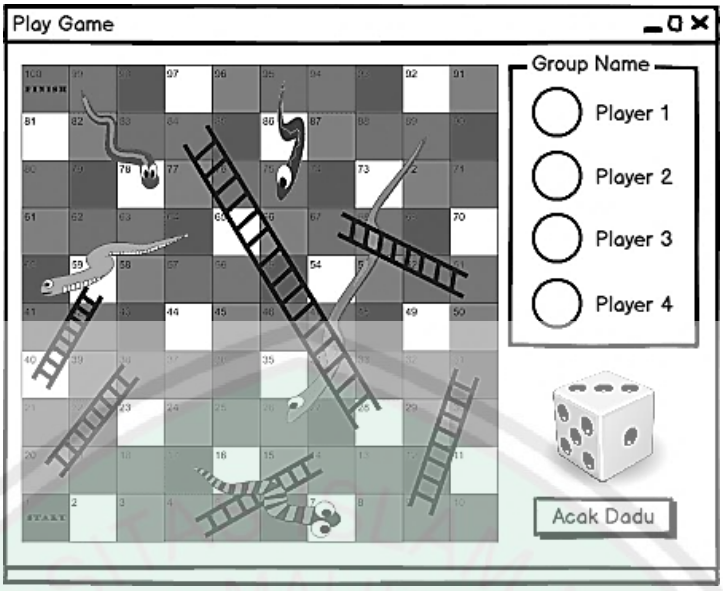
ditunjukkan pada Gambar 3.4. Apabila player ingin membuat grup baru, maka akan muncul form buat grup seperti pada Gambar 3.5.



Gambar 3.4 Rancangan Tampilan Grup

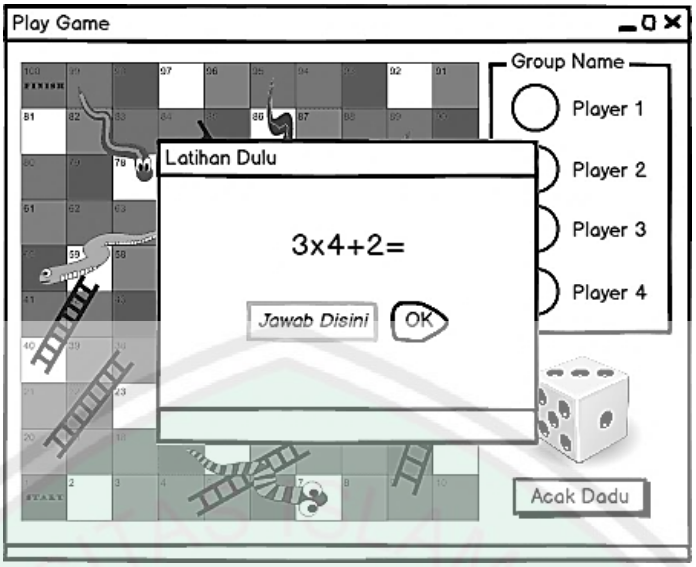


Gambar 3.5 Rancangan Tampilan Form Buat Grup



Gambar 3.6 Rancangan Tampilan *Game Board*

Setelah *player* mendapat grup, maka *player* dapat memainkan *game*. Tampilan halaman *game* ditunnjukkan Gambar 3.6. Pada halaman *game* terdapat papan *game* yang terdiri dari 100 blok/kotak, di dalamnya terdapat beberapa ular dan tangga. Papan ini yang digunakan oleh *player* dalam bermain. Disisi kanan halaman, terdapat identitas (*username*) *player-player* yang tergabung dalam grup yang sama dan tentunya dadu untuk menentukan langkah dalam bermain game ular tangga. Pada game ini ditambahkan soal-soal aritmatika sebagai unsur edukatif. Soal-soal ini muncul ketika player menemui ular atau tangga seperti yang telah dijelaskan pada subbab 3.1.2 sebelumnya. Rancangan tampilan halaman soal dapat dilihat pada Gambar 3.7



Gambar 3.7 Rancangan Tampilan Soal

Soal aritmatika ini ditampilkan dalam bentuk *pop up* (Gambar 3.7) yang otomatis muncul ketika player berada di ujung ular atau tangga. *Player* harus menjawab dengan mengisikan jawaban pada *text field* yang disediakan kemudian sistem akan memproses jawaban apakah jawaban tersebut benar atau salah.

3.3 Perancangan Sistem

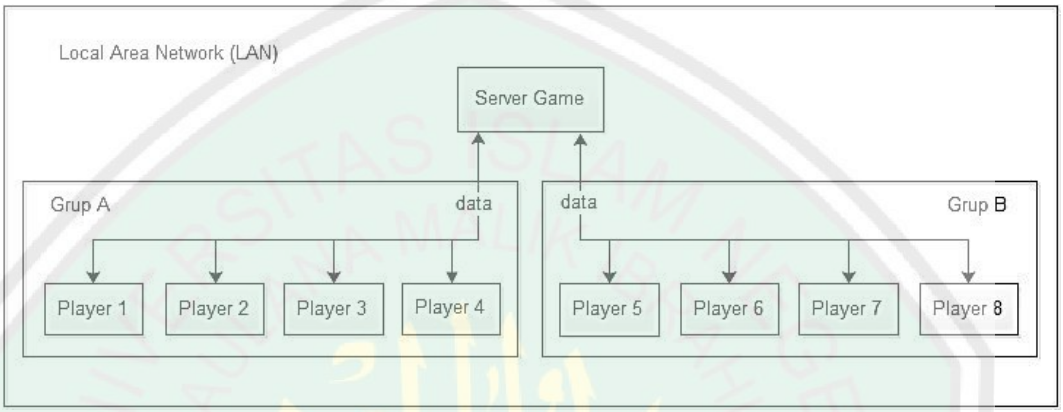
Secara umum, alur pengiriman data pada *client* dan *server game* edukasi ular tangga *multiplayer* berbasis jaringan ini adalah sebagai berikut (Gambar 3.8):

Gambar 3.8 Alur Pengiriman Data pada *Client* dan Server

Untuk lebih jelasnya, penjelasan diagram pada Gambar 3.8 adalah sebagai berikut.

3.2.1 Koneksi *Client* dan *Server*

Game ini dapat dimainkan oleh dua sampai empat *player* yang berperan sebagai *client* sebagaimana yang telah dijelaskan sebelumnya. Diagram (Gambar 3.9) berikut menunjukkan hubungan antara *server game* dan beberapa *client* (*player*).



Gambar 3.9 Diagram *Client* Server *Game*

Gambar 3.9 menunjukkan hubungan antara *server game* sebagai pusat jaringan dan pengelola paket data yang dikirim oleh bebrapa *client* (*player*). Semua data *client* dikirimkan ke *server*. Kemudian *server* memproses dan mem*broadcast* data tersebuat ke *client-client* yang berada dalam grup yang sama. *LAN* disini merupakan media penghubung antara *server* dan *client* untuk pengiriman data.

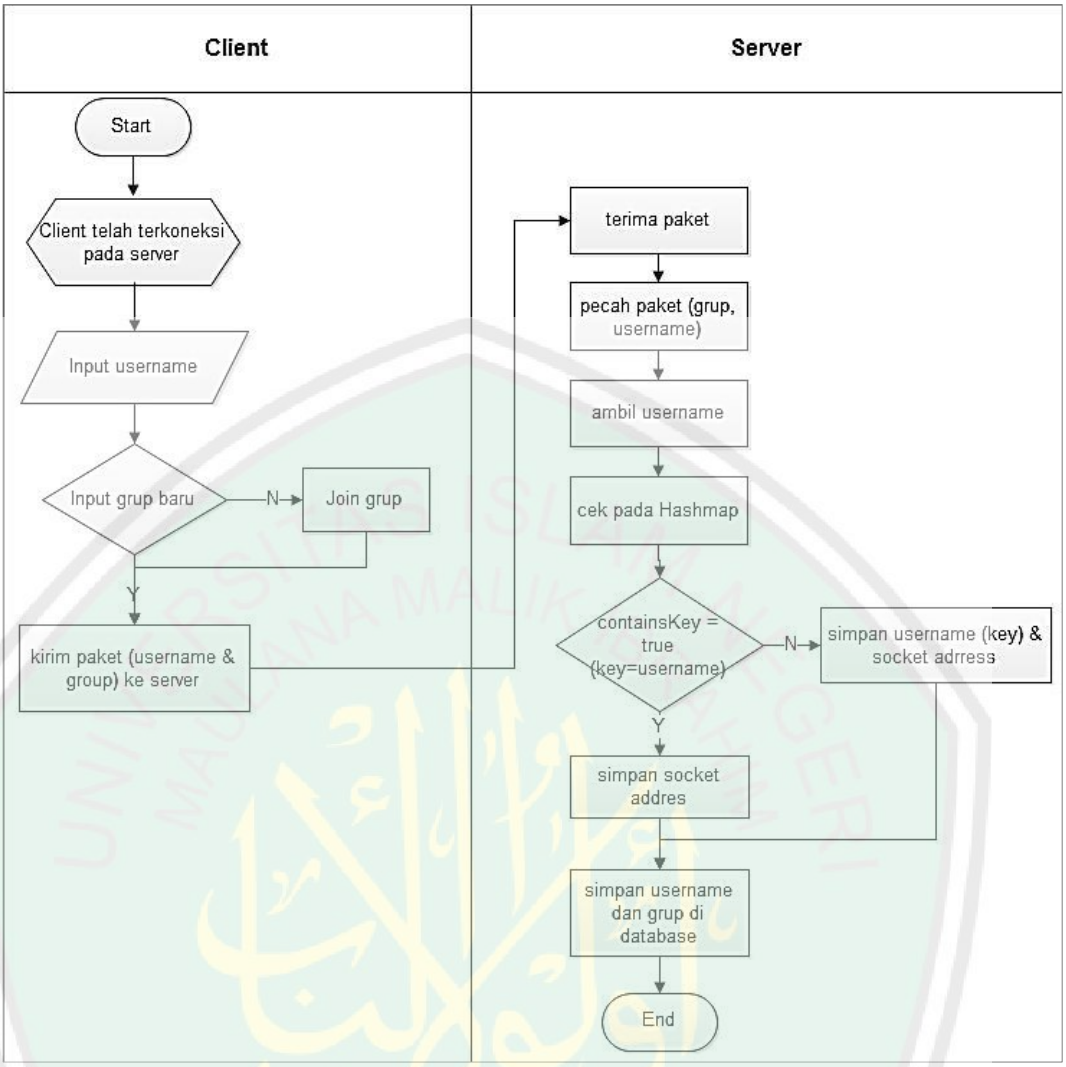
Proses pertama dalam membangun jaringan, diawali dengan *server* membuka koneksi dan menunggu *client* yang akan bergabung. *Server* membuka sebuah *port* yang telah ditentukan. *Port* merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung. Pada sisi aplikasi server, suatu *socket* server dibentuk dan melakukan operasi *listen*/menunggu. Operasi ini pada intinya menunggu permintaan koneksi dari sisi *client*.

Selanjutnya, *client* melakukan *request* ke *IP* server dan port yang telah dibuka oleh server sebelumnya. Pada saat permintaan koneksi *client* sampai pada server, maka server akan membuat suatu *socket*. *Socket* ini yang nantinya akan berkomunikasi dengan *socket* pada sisi *client*. Setelah itu *socket* server dapat kembali melakukan *listen* untuk menunggu permintaan koneksi dari *client* lainnya.

Dalam *game* ini, server dapat menerima lebih dari satu *client* (*player*). Oleh karena itu, *server* mengimplementasikan *multithreading* supaya dapat menampung proses koneksi dari beberapa *client*. *Multithreading* sendiri merupakan kemampuan yang memungkinkan beberapa intruksi atau proses dapat dijalankan secara bersamaan dalam sebuah program. Sedangkan *thread* adalah alur kontrol dari suatu proses atau sekumpulan instruksi yang dapat dieksekusi secara teratur dengan proses lainnya.

3.2.2 Input pada *Client*

Apabila *client* telah terkoneksi dengan *server*, proses selanjutnya yaitu *client* menginput *username* dan memilih grup. Alur detailnya dijelaskan pada gambar 3.10.



Gambar 3.10 *Flowchart* Proses *Client Input Username* dan Grup

Client yang telah terkoneksi pada *server*, selanjutnya menginput *username* yang digunakan selama permainan. *Username* ini menjadi inisiasi tersendiri bagi tiap-tiap *player*. Setelah input *username*, *player* dapat membuat grup atau bergabung dengan grup yang sudah dibuat oleh *player* lain sebelumnya. Setelah *player* meng-*input* grup, *client* mengirimnya ke *server*. *Server* menerima dan menyimpannya ke dalam *Hashmap*. Struktur dari *HashMap* ini adalah

```
HashMap (String, Socket)
```

String disini berperan sebagai *key* yang berisi *username*. Sedangkan *Socket* merupakan *value*-nya untuk menyimpan *socket address player* tersebut.

Pada dasarnya *HashMap* hampir sama dengan *ArrayList*, yaitu suatu *class* untuk menyimpan data (objek) namun *index* atau ordinat (biasanya disebut *key*) nya bertipe objek juga (*string*, *class*, *numeric*, dsb) yang di *hash*. Sehingga secara fungsi dan penggunaan *HashMap* lebih komplek dari pada *ArrayList* yang notabene lebih mirip dengan data tipe *Array* biasa. Penggunaan *HashMap* merupakan struktur yang tepat untuk pengelolaan data yang kompleks. Sistem server pada game ini *HashMap* digunakan untuk menyimpan *username* dan *socket address* tiap-tiap *player*. Hal ini dapat diibaratkan seperti penyimpanan kontak atau nomor telepon yang biasanya terdiri dari nama dan nomor telepon. *Username* diibaratkan namanya dan *socket address* adalah nomor telepon.

Selanjutnya server menyimpan *username* dan nama grup ke *database*. Penyimpanan ini nantinya digunakan dalam pengiriman data untuk mencari *player-player* dalam grup. Tabel 3.1 menunjukkan desain *database* yang digunakan dalam game ini.

Tabel 3.1 Desain Database

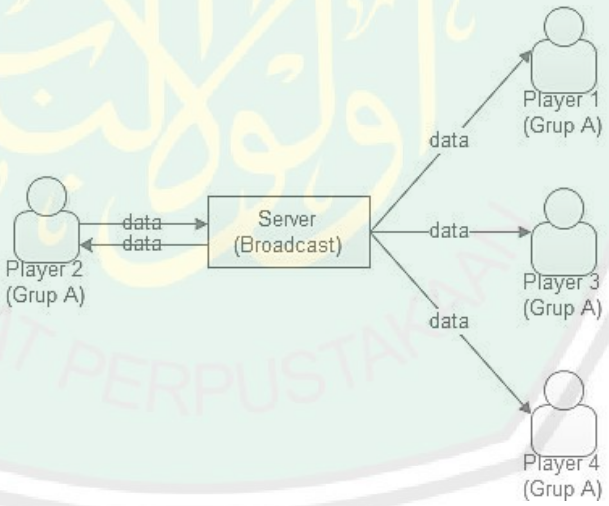
No	Nama *Field*	Jenis	Keterangan
1	username	varchar(10)	Primary Key
2	grup	varchar(10)	
3	ket	varchar(10)	

Database dalam *game* ini hanya terdapat satu tabel yang berisi 3 *field* yaitu *username*, grup, dan keterangan. *Field username* untuk menyimpan *username player*, *field* grup untuk menyimpan nama grup yang diikuti *player*, dan *field* ket menyimpan keterangan *player* atau grup tersebut. *Username* merupakan *primary*

key dari tabel ini. Hal ini bertujuan untuk memudahkan pencarian dan pengelolaan *socket* pada *HashMap*.

3.2.3 Pemrosessan Data

Game ular tangga ini dapat dimainkan oleh 2 sampai 4 *player*. *Player* menunggu *player* yang lainnya bergabung dalam grup yang sama. Setelah semua pemain dalam suatu grup siap, maka *game* dapat mulai dimainkan. Setiap *player* bermain dan mengirimkan paket data yang berisi posisinya ke *server* untuk di-*broadcast* kepada *player* yang lain dalam grup yang sama. Gambar 3.11 menunjukkan rancangan pengiriman paket data. Misalnya dalam Grup A terdiri dari empat *player*. Player2 mendapat giliran bermain. Maka Player2 mengirim paket ke *server*. Kemudian server mem-*broadcast* ke semua *player* yang tergabung dalam grup A.



Gambar 3.11 Rancangan Pengiriman Paket

Sebuah paket data terdiri dari dua bagian yang paling penting, yaitu *header* dan data.

1. *Header*

   Header adalah *single byte* yang menyatakan bagaimana cara menangani data. *Header* biasanya dikenal dengan Data ID. Ketika header dikirim oleh *client* kepada *server*, maka *server* akan mengetahui bahwa seluruh data pada paket merupakan informasi sebagaimana *header* yang dideklarasikan.

2. Data

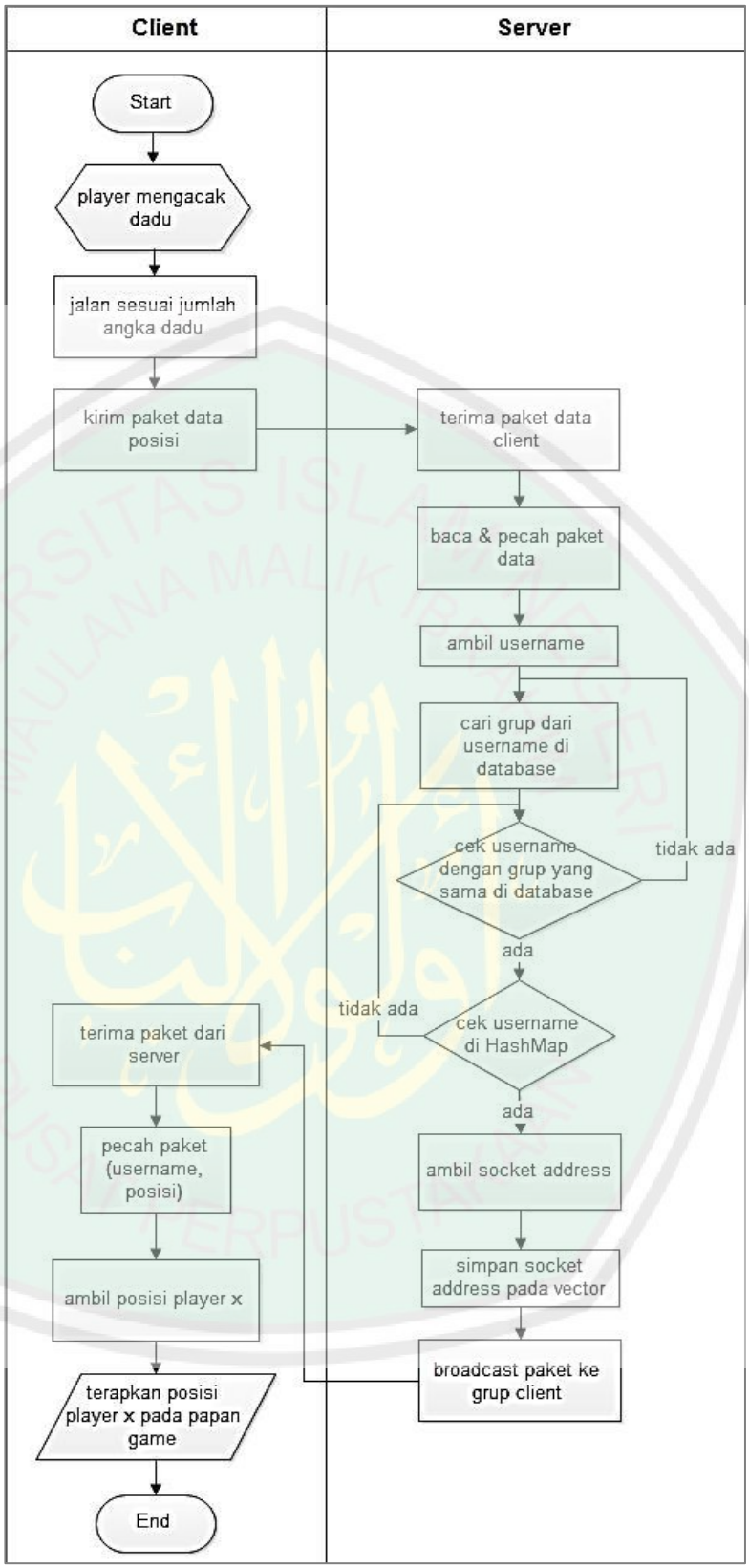
   Data merupakan isi dari paket yang akan diproses oleh *server*/*client*. Kegunaan data di dalam paket bergantung dari informasi *header* yang dimilikinya.

Pada umumnya struktur paket adalah sebagai berikut:

```
<DataID>(<Data><Separator><Data><Separator>....)<EndID>
```

Struktur diatas adalah struktur yang paling umum paket. Untuk penerapannya sendiri disesuaikan dengan kebutuhan sistem.

Proses pengiriman data pada *game* ini dapat dilihat pada diagram alur berikut (Gambar 3.12).



Gambar 3.12 *Flowchart* Pemrosesan Data *Game*

Untuk mendapatkan data posisi, *player* mengacak dadu. Kemudian *player* mendapatkan sejumlah angka dadu untuk berjalan pada papan permainan. Selanjutnya *player* (*client* x) ini mengirim data posisinya ke *server* dengan format paket:

```
<nama_grup>:<username>:<posisi_saat_ini>:<posisi_sebelumnya>
```

Nama Grup merupakan nama dari grup bermain yang diikuti *player*, *username* merupakan nama *player* yang sedang bermain. Posisi saat ini merupakan posisi terakhir *player* pada bidak. Sedangkan posisi sebelumnya merupakan posisi sebelum mengacak dadu. Tanda ":" merupakan separator yang digunakan untuk memisahkan data agar dapat diproses secara terpisah.

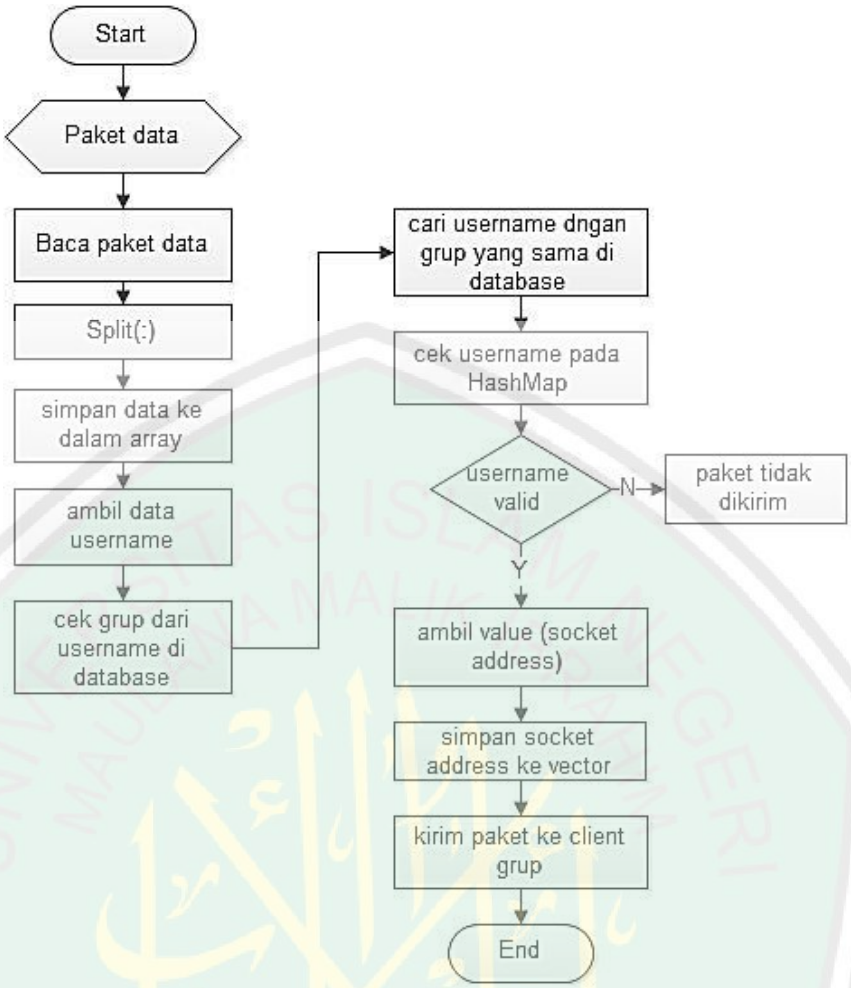
Misalnya,

```
GrupX:Budi:17:15
```

GrupX merupakan nama grup, Budi adalah *username player*, 17 merupakan posisi terakhir dari *player* Budi, dan 15 yaitu posisi sebelumnya.

3.2.4 Pengolahan Paket oleh Server

Selanjutnya, *server* menerima paket yang dikirim oleh *player*. Tidak hanya sebagai *broadcaster*, server dalam game ini juga berperan aktif dalam memproses paket data. Server memproses dan memecah paket tersebut dengan memisahkan data-data dalam paket tersebut. *Server* memecah paket untuk mengetahui grup dari *client* yang sedang bermain. Diagram alur berikut (Gambar 3.13) menunjukkan proses server mengolah paket data.



Gambar 3.13 *Flowchart* Pengolahan Paket oleh Server

Pemecahan paket dapat dilakukan dengan fungsi *split*. Fungsi *split* ini biasanya digunakan dalam manajemen suatu *String* kalimat dalam *Java* untuk mengambil beberapa kata yang dibutuhkan dengan delimiter tertentu. Delimiter merupakan satu akrakter atau lebih yang dipakai untuk memberi batasan atau sebagai pemisah data yang disajikan dalam bentuk *plain text*. Contoh delimiter adalah tanda koma (,) titik (.) titik dua (:) dsb. Paket dipecah menjadi tiga bagian (nama grup, *username*, dan posisi), data yang telah dipecah ini ditampung di dalam *array*. Apabila digambarkan, isi *array* tersebut menjadi:

```
arraydata[]={[nama_grup],[username],[posisi_saat_ini],[posisi_sebe
lumnya]};
```

Untuk mendapatkan *username*, dari *array* data paket diatas tinggal diambil pada index ke 1.

```
namaGrup = arraydata[1];
```

Setelah mengetahui *username* dari *client* tersebut, server melakukan pengecekan pada *HashMap* apakah *username* tersebut ada. Selanjutnya server mengambil nama grup dari *username* tersebut pada *database*. Kemudian dicari *username player* yang sama grupnya dalam database. Setelah mengetahui *username* dari *client-client* dalam satu grup, *username* tersebut dicek di *HashMap* untuk mengambil *socket adrres client*. *Socket address* yang didapatkan dikumpulkan di dalam *vector*. Selanjutnya paket dikirim ke *socket address* yang telah disimpan dalam *vector*. Server mem-*broadcast* paket tersebut kepada semua *client* anggota grup yang sama. Sedangkan *client* lain dengan grup yang berbeda tidak akan mendapatkan kiriman paket tersebut.

Vector digunakan sebagai penyimpanan *socket address* dalam *game* ini. *Vector* adalah sebuah *class* yang diturunkan dari *interface collection*, yaitu sebuah *interface* yang digunakan untuk pengolahan data yang bersifat seperti *array* dinamis, yakni *array* yang ukurannya secara dinamis dapat membesar ketika data yang dimasukkan melebihi daya tampung. Setiap metode dalam *Vector* diberi *keyword* "*synchronized*", sehingga ketika dieksekusi dalam sebuah *thread*, maka tak akan terjadi kemacetan *thread*. *Array* memiliki daya tampung terbatas dari yang dideklarasikan (daya tampung tidak dinamis). *ArrayList* memiliki fungsi yang sama dengan *Vector*, namun metode yang terdapat pada *ArrayList* tidak

diberi *keyword* "*sycrhonized*", sehingga ketika dieksekusi dalam sebuah *thread* dapat mengakibatkan unsafe *thread* dan terjadinya tubrukan *thread* atau kemacetan saat eksekusi.

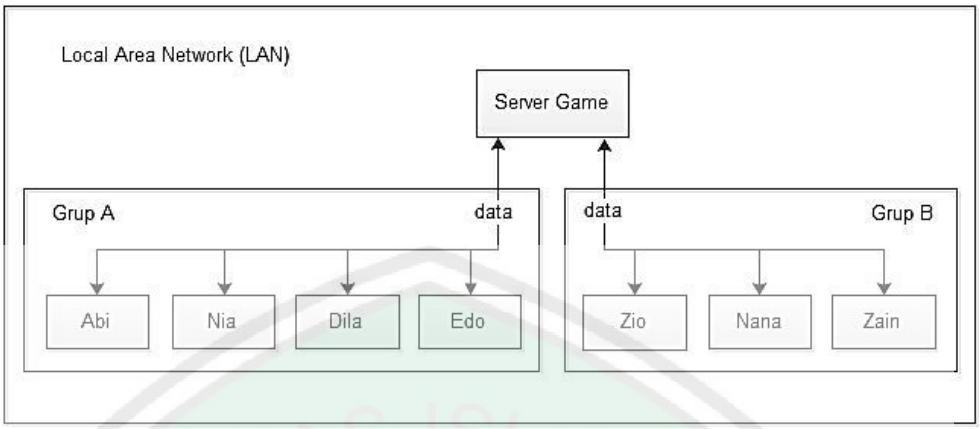
Berikut *pseudo code* pengolahan paket menggunakan *HashMap* dan *Vector*

```
hm = new HashMap<>
ClientHandler ch
in = new BufferedReader
String dt;
if (dt = in.readLine) != null then
  String dl[] = login.split(":")
  if dl[0]="LOGIN" then
    String grup = dl[1]
    String username = dl[2]
    hm.put(username, ch)
  end if
end if
```

```
in = new BufferedReader
String tm = in.readLine()
String dt[] = tm.split(":")
listgrup, listuser = new Vector()
String sqlcode = "SELECT grup FROM gruppemain
GROUP BY grup"
ResultSet rs = stat.executeQuery(sqlcode)
while rs.next() do
  listgrup.add(rs.getString("grup"))
end while
if listgrup=dt[0] then
  grupname = dt[0]
  String sql ="SELECT * FROM gruppemain
  WHERE grup='"+grupname+"'"
  ResultSet rss = stat.executeQuery(sql)
  while rss.next() do
    listuser.add(rss.getString("username"))
  end while
  for int i=0 to i<listuser.size() do
    ClientHandler sc = ssa.ds.get(listuser.get(t))
    sc.send(textMasuk);
  end for
end if
```

2.3.5 *Server Broadcast* dan *Output*

Paket yang telah dikirim oleh server diterima oleh *client*. *Client* memproses data tersebut dengan melakukan pemecahan data untuk mendapatkan data posisi *player* x. Proses ini sama halnya dengan proses pemecahan paket pada *server*. Setelah mendapatkan posisi *player* x, *client* menerapkan posisi tersebut pada *interface* bidak *game* Ular Tangga. Dengan demikian *game* ini dapat dimainkan secara *real time*. Contoh komunikasi data *client* server dalam game Ular Tangga *multiplayer* terdapat pada Gambar 3.14.



Gambar 3.14 Contoh Komunikasi Data *Client* Server

Contoh komunikasi data pada Gambar 3.14 menunjukkan beberapa *client* (*player*) yang terhubung ke *server*. *Client-client* tersebut terbagi dalam 2 grup yaitu Grup A dan Grup B. Untuk lebih jelasnya, Tabel 3.2 menunjukkan contoh pengiriman dan penerimaan paket data. Dari Tabel 3.2 dapat diketahui bahwa server secara aktif memproses data dan mengambil informasi grup kemudian mem-*broadcast* pada *client* yang berada pada grup yang sama.

Tabel 3.2 Pengiriman Paket Data

Pengirim (Posisi)	Paket yang diterima server	Client yang menerima						
		Abi	Nia	Dila	Edo	Zio	Nana	Zain
Nia(3)	GrupA:Nia:3:1	v	v	v	v			
Edo(5)	GrupA:Edo:5:1	v	v	v	v			
Zain(6)	GrupB:Zain:6:1					v	v	v
Abi(5)	GrupA:Abi:5:1	v	v	v	v			
Zio(4)	GrupB:Zio:4:1					v	v	v
Nia(9)	GrupA:Nia:9:3	v	v	v	v			
Dila(6)	GrupA:Dila:6:1	v	v	v	v			
Nana(5)	GrupB:Nana:5:1					v	v	v
Edo(11)	GrupA:Edo:9:5	v	v	v	v			
Zain(12)	GrupB:Zain:12:6					v	v	v

3.4 Kebutuhan Sistem

Perangkat yang digunakan untuk pembuatan dan pengoprasian *game* ini meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Spesifikasi perangkat yang digunakan adalah sebagai berikut.

3.4.1 Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada proses pembuatan dan pengoperasian game ini yaitu: Intel(R) Core(TM) i3-4030U CPU 1.90 GHz., *RAM* 4 GB, *keyboard*, *mouse*, dan *router* jaringan sebagai media komunikasi data.

3.4.2 Kebutuhan Perangkat Lunak (*Software*)

Adapun kebutuhan perangkat lunak yang digunakan pada proses pembuatan *game* ini yaitu, Java SE Development Kit 7 Update 9, NetBeansIDE 7.3.1, MySQL sebagai media penyimpanan, serta Adobe Photoshop CS3 untuk desain tampilan *game*.

3.5 Perancangan Pengujian

Pengujian yang akan dilakukan terhadap keseluruhan sistem menggunakan teknik pengujian *white-box* dan *black-box*. Pengujian *black-box* yang dilakukan bertujuan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Sedangkan pengujian *white-box* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

Pengujian selanjutnya yaitu pengujian kinerja server sebagai moderator grup pemain (*client*). Pengujian ini dilakukan dengan mengoneksikan beberapa

client dengan grup yang berbeda-beda dan melakukan pengamatan pengiriman paket data. Dari hasil pengamatan terhadap *client* dan server, akan dapat diketahui tingkat keberhasilan server dalam pengiriman paket. Sehingga presentase keberhasilan *active server* dalam pengiriman data dapat didapatkan.

$$\text{Keberhasilan pengiriman data} = \frac{\sum \text{keberhasilan pengiriman} \times 100\%}{\sum \text{seluruh pengiriman data}}$$

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas mengenai hasil implementasi dari analisis dan perancangan yang telah disusun. Implementasi disini meliputi implementasi *interface* dan implementasi metode. Selain itu, pada bab ini juga menjelaskan pengujian sistem yang dilakukan untuk mendapatkan hasil dari penelitian.

4.1 Implementasi *Interface*

Interface yang terdapat pada aplikasi yang telah dibuat disesuakan dengan perancangan tampilan yang telah ada. Berikut merupakan *interface* yang ada dalam game ini.

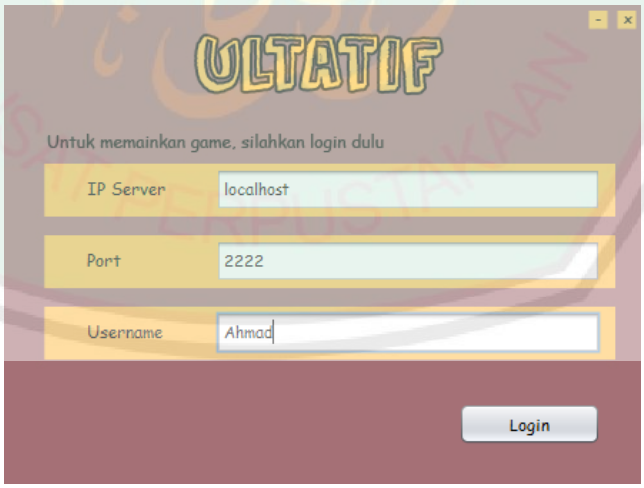
4.1.1 *Interface* Server

Server merupakan komponen utama yang mengatur dan mengontrol seluruh aktifitas *player*. Server digunakan untuk memonitor aktifitas pengiriman data paket. Dalam game ini, *interface* server dibuat sederhana (Gambar 4.1). Terdapat *toogle button* (*ON/OFF*) untuk menghidupkan atau mematikan kinerja server dan textarea untuk memonitor proses yang sedang berlangsung.

Gambar 4.1 *Interface* Server

4.1.2 *Interface* Login

Pada client terdapat beberapa *interface*. *Interface* login pada Gambar 4.2 menampilkan form *Login player* yang akan bermain. *Player* mengisikan *host* atau *IP* server dan *port* yang sama dengan server serta *username* yang digunakan sebagai identitas pemain dalam *game*. Setelah semua data terisi, pemain dapat memilih grup pemain.



Gambar 4.2 *Interface* Login

4.1.3 *Interface* Daftar Grup Pemain

Interface daftar grup pemain merupakan *interface* yang menampilkan daftar grup bermain yang telah dibuat oleh *player*. Apabila belum ada grup yang tersedia, maka *user* dapat membuat grup baru. Tampilan *interface* daftar grup pemain ditunjukkan pada Gambar 4.3.

Gambar 4.3 *Interface* Daftar Grup Pemain

Disini *player* dapat memilih untuk bergabung dalam grup yang telah dibuat oleh *player* lain atau membuat grup bermain yang baru. Tombol "*Refresh*" digunakan untuk memuat atau memperbarui grup bermain yang tersedia. Tombol "Buat Grup" digunakan *player* saat membuat grup baru. Tombol ini menyambungkan pada interface Buat Grup Baru. Selain itu, pada *interface* ini juga terdapat tombol menu lain seperti "Petunjuk" dan "Tentang".

Apabila *user* akan bergabung dengan grup yang sudah ada, *user* dapat memilih (*click*) pada *list* atau daftar grup. Maka akan muncul jendela konfirmasi seperti pada Gambar 4.4.

Gambar 4.4 *Interface* Gabung Grup

4.1.4 *Interface* Buat Grup Baru

Apabila *player* ingin membuat grup sendiri, user dapat mengisikan nama grup yang akan dibuat pada *textfield* seperti pada Gambar 4.5. Grup yang telah dibuat akan ditampilkan pada daftar grup pemain.

Gambar 4.5 *Interface* Buat Grup Baru

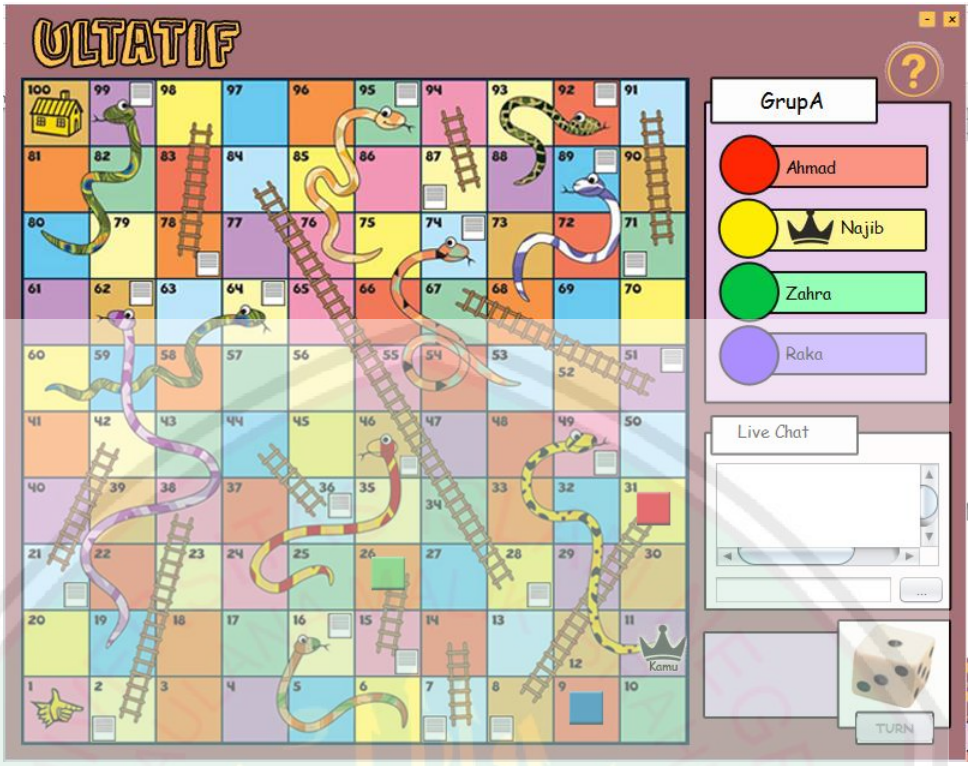
4.1.5 *Interface* Daftar Pemain dalam Grup

Pemain yang tergabung dalam grup yang sama ditampilkan dalam *interface* daftar pemain (Gambar 4.6). Pada *interface* ini terdapat *textarea* yang menampilkan siapa saja yang tergabung dalam grup. Apabila ada pemain yang bergabung dengan grup yang sama, maka akan muncul pada daftar pemain. Selain itu terdapat tombol "*Play*" yang digunakan untuk memulai *game*, apabila seluruh *player* dalam grup telah bergabung.

Gambar 4.6 *Interface* Daftar Pemain dalam Suatu Grup

4.1.6 *Interface* Game

Interface game adalah tampilan yang digunakan untuk memainkan *game* ular tangga. Pada *interface* ini terdapat papan permainan ular tangga, pion pemain, dadu, identitas pemain dan lawan-lawannya, serta beberapa menu untuk memudahkan pemain dalam memainkan *game* ini. *Interface game* ditunjukkan pada Gambar 4.7.



Gambar 4.7 *Interface Game*

Pion *player* diberikan warna yang berbeda untuk mengetahui siapa yang sedang bermain. Pion *player* yang memainkan dibedakan dengan tanda dan bentuk yang berbeda. Hal ini ditujukan untuk memudahkan untuk mengetahui pion *player* dan pion lawan.

Unttuk memainkan *game* ini cukup mudah. *Player* memutar dadu dengan klik tombol "*Turn*", kemudian dadu diputar sera *random*. *Player* akan berjalan sebanyak angka perolehan dadu tersebut. Dalam papan permainan terdapat 100 kotak untuk lintasan *player* berjalan. Pada *game* ini ada beberapa kotak yang berisi soal kuis. Terdapat tanda (simbol soal) pada kotak yang memunculkan soal kuis.

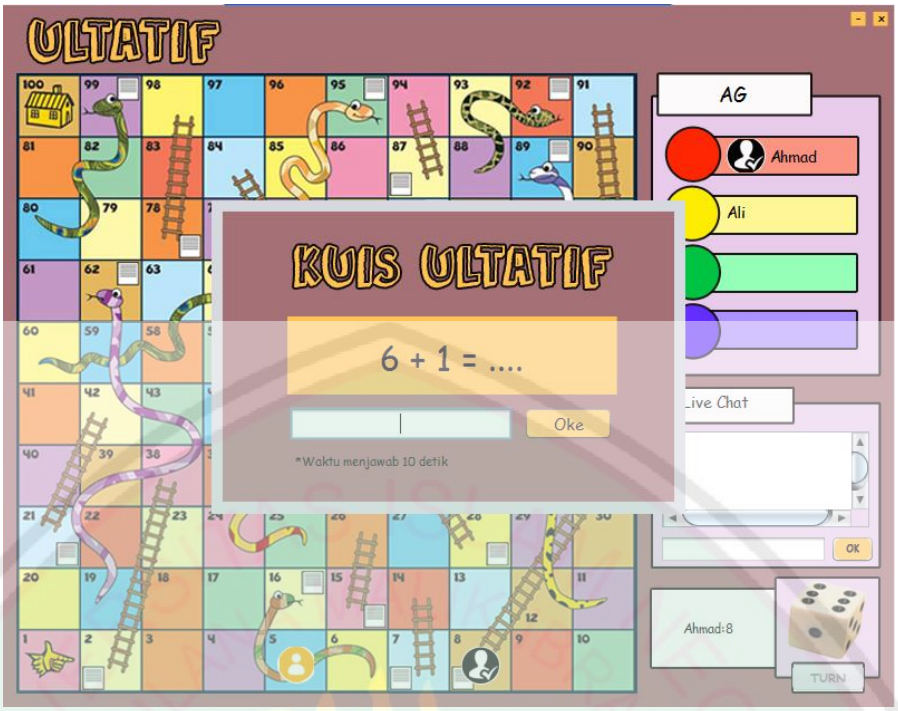
Dalam *game* ini juga disertakan fitur *Live Chat* (Gambar 4.8). Fitur ini dapan digunakan *player* dalam grup tersebut untuk saling bertukar pesan singkat.

Untuk mengirim pesan, player dapan menulis pesan pada *textfield* yang tersedia kemudian klik tombok "OK" untuk mengirimkan pesan. Maka *player* dalam grup yang sama akan menerima dan penampilkan pesannya pada *textarea*.

Gambar 4.8 Fitur *Live Chat*

4.1.7 *Interface* Soal Kuis

Interface soal kuis akan muncul pada saat pemain menemui ular atau tangga. Tampilan *interface* ini seperti pada Gambar 4.9. Pemain diharuskan untuk menjawab soal terlebih dahulu sebelum naik tangga atau menuruni ular. Soal kuis yang disajikan merupakan soal aritmatika (mencongak). Kemudian pemain dapat mengisikan jawabannya pada *textfield* yang tersedia dan mengecek jawabannya. Apabila jawaban yang diisikan benar akan muncul notifikasi dan dapat melanjutkan aksi dalam permainan. Begitu pula apabila jawaban yang diisikan salah, maka akan muncul notifikasi dan pemain akan menerima konsekuensi dalam permainan tersebut. *Popup* soal ini ditampilkan hanya 10 detik. Apabila *player* tidak menjawab lebih dari 10 detik, maka dianggap menjawab salah.



Gambar 4.9 *Interface* Soal Kuis

4.2 Implementasi Metode

Game ular tangga yang dibangun dapat dimainkan secara bersama-sama dalam satu waktu oleh beberapa *player* yang berada pada satu jaringan (*Real Time Multiplayer Game*). Dalam satu permainan dapat diikuti oleh 2 sampai 4 *player*. Untuk menampung lebih banyak *player* yang dapat memainkan *game*, maka dibuat grup-grup bermain. Dalam hal ini, *Hashmap* dan *Vector* diterapkan pada server *game* yang menangani grup *client* (*player*). *Hashmap* dan *Vector* pada server digunakan dalam manajemen penyimpanan dan pengiriman data grup.

HashMap adalah *class* implementasi dari *Map*, *Map* itu sendiri adalah *interface* yang memiliki fungsi untuk memetakan nilai dengan *key* unik. *HashMap* berfungsi sebagai *memory record management*, dimana setiap *record* dapat disimpan dalam sebuah *Map*. Sedangkan *Vector* adalah sebuah *class* yang diturunkan dari *interface collection*, yaitu sebuah *interface* yang digunakan untuk

pengolahan data yang bersifat seperti *array* dinamis, yakni *array* yang ukurannya secara dinamis dapat membesar ketika data yang dimasukkan melebihi daya tampung. Setiap metode dalam *Vector* diberi *keyword* "*synchronized*", sehingga ketika dieksekusi dalam sebuah *thread*, maka tak akan terjadi kemacetan *thread*.

4.2.1 Implementasi *HashMap* pada Server

HashMap pada pada server digunakan untuk penyimpanan *socket address player*. Penggunaan *HashMap* dalam sistem ini dapat diibaratkan seperti penyimpanan kontak atau nomor telepon yang biasanya terdiri dari nama dan nomor telepon. *Username* diibaratkan nama dan *socket address* adalah nomor telepon. Salah satu kelebihan *HashMap* yaitu dapat menampung variabel yang bervasiasi, salah satunya yaitu *Socket*. *Socket address* merupakan alamat yang digunakan untuk pengiriman data. Pada Java, *HashMap* dideklarasikan dengan format berikut:

```
HashMap<String, Socket> nama_HashMap = new HashMap();
```

HashMap digunakan untuk menyimpan untuk menyimpan *username player* (*String*) dan *socket address player* (*Socket*).

Socket adrress player diambil dari *player* yang yang login dan melalui *class* ClientHandler dengan *constructor* yang ditunjukkan pada Modul Program 4.1.

Modul Program 4.1 *Source Code Client Handler*

```
private Socket client;
private BufferedReader in;
private PrintWriter out;
ClientHandler(Socket clientSocket) {
    try {
        client = clientSocket;
        in = new BufferedReader(new
        InputStreamReader(client.getInputStream()));
        out = new PrintWriter(client.getOutputStream(), true);
    } catch (Exception e) {
```

```
            System.out.println(e);
        }
}
```

Selanjutnya pada *class* Server, dibuat object dari *class* ClientHandler untuk *input socket address*.

Modul Program 4.2 *Source Code* Penanganan Login

```
HashMap<String, ClientHandler> daftarSocket = new HashMap<>();
ServerSocket ss = new ServerSocket(2222);
Socket s = ss.accept();
ClientHandler ch = new ClientHandler(s);
In = new BufferedReader(new InputStreamReader(s.getInputStream()));
Connection connect = Koneksi.getConnection();
String datalogin = "";
if ((datalogin = in.readLine()) != null) {
    String dataLog[] = login.split(":");
    if (dataLog[0].equals("LOGIN")) {
        String groupname = dataLog[1];
        String username = dataLog[2];
        daftarSocket.put(username, ch);
        String sqlcode = "INSERT INTO gruppemain (username, grup, ket)
            VALUES('"+username+"','"+groupname+"','"+"READY')";
        try {
            Statement stat = connect.createStatement();
            stat.executeUpdate(sqlcode);
            stat.close();
        } catch (Exception ex) {
            System.out.println("Kesalahan, silahkan periksa :" + ex);
        }
    }
}
```

Ketika *client* (*player*) login, client mengirimkan paket dengan struktur

```
LOGIN:grup:username
```

Pada saat server menanggapi *request* login (Modul Program 4.2), paket tersebut kemudian dipecah (`login.split(":")`) menjadi 3 bagian dan disimpan dalam *array*. "LOGIN" merupakan data Id grup adalah grup *player* dan *username* adalah nama/identitas *player*. Data *username* dan *socket* yang telah diambil sebelumnya disimpan dalam *HashMap*.

```
nama_hashmap.put(username, socket_address);
```

Username dan grup kemudian disimpan di *database* server. *Socket adrres* tidak disimpan dalam *database* dikarenakan tidak terdapat data dengan tipe *socket* di dalam *database*. Sehingga *HashMap* digunakan sebagai penyimpanan *socket address client*.

4.2.2 Implementasi *Vector* pada Server

Vector dalam sistem ini digunakan sebagai manajemen *username* dan *socket address* untuk mengirim paket data ke *client-client* dengan grup yang sama. *Vector* memiliki beberapa kelebihan salah satunya yaitu ukurannya secara dinamis dapat membesar ketika data yang dimasukkan melebihi daya tampung dan tersinkronisasi. Hal ini sangat mendukung kinerja server yang mengimplementasikan *multithreading* dalam sistemnya. *Source code* untuk pengiriman paket data grup ditunjukkan pada Modul Program 4.3.

Modul Program 4.3 *Source Code Broadcast* Data Grup

```
BufferedReader in = new BufferedReader(new
    InputStreamReader(client.getInputStream()));
PrintWriter out = new PrintWriter(client.getOutputStream(), true);
textMasuk = in.readLine();
String data[] = textMasuk.split(":");

try {
    String sqlcode = "SELECT grup FROM gruppemain GROUP BY grup";
    Connection connect = Koneksi.getConnection();
    stat = connect.createStatement();
    ResultSet rs = stat.executeQuery(sqlcode);
    Vector listgrup = new Vector();
    while (rs.next()) {
        listgrup.add(rs.getString("grup"));
    }
    rs.close();

    if (listgrup.contains(data[0])) {
        Vector listuser = new Vector();
        String groupname = data[0];
        String sql ="SELECT * FROM gruppemain WHERE
            grup='"+groupname+"'";
        Connection conn = Koneksi.getConnection();
        Statement stat = conn.createStatement();
        ResultSet rss = stat.executeQuery(sql);
        while (rss.next()) {
```

```
                listuser.add(rss.getString("username"));
            }
            for (int t = 0; t < listuser.size(); t++) {
                ClientHandler sc = ssa.daftarSocket.get(listuser.get(t));
                sc.send(textMasuk);
            }
        }
        rss.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
```

Pertama, server menerima paket yang dikirim oleh client (`textMasuk = in.readLine()`). Paket yang dikirim client berstruktur:

`namagrup:username:posisi_awal:posisi_tujuan`

Kemudian server memecah paket dan menyimpan data-datanya dalam *array*. dan mengambil nama grup *client* tersebut.

```
data = textMasuk.split(":");
```

Dari pemecahan paket, didapatkan nama grup adalah `data[0]`. Nama grup kemudian digunakan untuk mencari *username* dengan nama grup yang sama di database. Berikut *query* yang digunakan untuk mencari *username* dalam grup yang sama.

```
"SELECT * FROM gruppemain WHERE grup='"+groupname+"'"
```

Username-username tersebut dikumpulkan dan disimpan dalam *Vector*.

```
Vector listuser = new Vector();
listuser.add(rss.getString("username"));
```

Selkanjutnya sistem mengambil *socket address username* tersebut dari *HashMap* yang telah disimpan pada saat *client* login.

```
for (int t = 0; t < listuser.size(); t++) {
    ClientHandler sc = ssa.daftarSocket.get(listuser.get(t));
    sc.send(textMasuk);
}
```

Dalam HashMap, *username* merupakan *key* dan *socket address* adalah *value*. Ketika akan mengambil *value* dari *key* tertentu dapat diakses dengan *method*

`get()` dari *HashMap*. Selanjutnya server mengirimkan kembali (mem-*broadcast*) paket tersebut kepada *client-client* yang berada pada grup tersebut. *Method* `send()` diambil dari *class* ClientHandler untuk mengirim paket (Modul Program 4.4).

Modul Program 4.4 *Souce Code* Mengirim Paket

```
PrintWriter out = new PrintWriter(client.getOutputStream(), true);
public void send(String message) {
   out.println(message);
}
```

Output Stream digunakan untuk mengirimkan data yang berupa aliran *byte* atau karakter dari *device input* ke *device output*.

4.2.3 *Multithreading Client*-Server

Multithreading adalah kemampuan sebuah program untuk melakukan lebih dari satu pekerjaan sekaligus. Keuntungan dari *multithreading* adalah sifat *respons* yang interaktif dan *real-time*. Hal ini sangat cocok digunakan dalam *realtime multiplayer game* karena server yang selalu aktif untuk menampung koneksi *player*. Selain itu, server dan *client* (*player*) harus selalu siap untuk menerima paket data yang dikirim oleh keduanya.

1. ***Multithreading* pada Server**

   Server mengimplementasikan *Multithreading* yang digunakan untuk menampung *client* yang request koneksi. *Thread* ditempatkan pada *class* ClientHandler.

Modul Program 4.5 *Source Code Multithreading Client* yang Masuk pada Server

```
class ClientHandler extends Thread {
private Socket client;
private BufferedReader in;
private PrintWriter out;
   public ClientHandler(Socket clientSocket) {
      try {
         client = clientSocket;
         in = new BufferedReader(new
             InputStreamReader(client.getInputStream()));
```

```
            out = new PrintWriter(client.getOutputStream(), true);
         } catch (Exception e) {
            System.out.println(e);
         }
      }

      @Override
      public void run() {
         do {
            String textMasuk = in.readLine();
            String data[] = textMasuk.split(":");
            .
            .
         }
      }
}
```

Class extends Thread otomatis memunculkan method run() di dalamnya. Method run() inilah yang akan dieksekusi oleh program ketika *thread* dimulai. Intruksi dan perintah yang dimasukkan dalam *method* ini adalah respon server ketika *client* mengirim pesan paket. Secara umum *client* mengirim paket dengan format.

```
<DataID><Separator><Data><Separator><Data>....
```

DataID menyatakan bagaimana cara sistem menangani data. Separator digunakan untuk memecah paket agar data dapat diproses secara terpisah. Sedangkan data merupakan inti/isi paket. *Client* akan mengirim data sesuai dengan *request* dari kebutuhannya dan server akan menangani *request* dari *client* tersebut. *Request client* terhadap server dalam sistem ini ada beberapa macam, diantaranya yaitu:

Tabel 4.1 *Request Client* terhadap Server

Data ID/Header	Fungsi
LOGIN	*Request* koneksi saat pertama *client* masuk
GRUPLIST	*Request* data grup yang tersedia
PLAYERLIST	*Request* data *player* dalam suatu grup
PLAY	*Request* kirim data pada *client* dalam suatu grup saat *game* dimulai
DATAGRUP	*Request* kirim data posisi atau data pesan singkat

Data ID/Header	Fungsi
	pada *client* dalam suatu grup saat bermain *game*
CLOSED	*Request player* keluar dari *game*

Dari DataID tersebut server akan melakukan penanganan *request* dari *client*. Pada saat *client* LOGIN, server melakukan penyimpanan *username*, *socket address*, dan grup *client* seperti yang telah dijelaskan sebelumnya. GRUPLIST merupakan *request* pada saat *client* akan memilih grup. Server mengirimkan data grup yang tersedia. Syarat grup yang dapat di *join* adalah jumlah *player* dalam grup tersebut kurang dari 4 dan grup tersebut belum memulai permainan.

PLAYERLIST yaitu *request client* untuk mendapat daftar *player* yang berada pada grup yang sama. Setelah mendapatkan daftar *player*, *client* menampilkannya pada *interface*. *Request* dengan dataID PLAY dilakukan jika ada *client* yang memulai *game*. Hal ini dimaksudkan agar semua *player* dapan memulai *game* secara bersamaan. Jadi apabila salah satu *player* memuluskan untuk memulai permainan maka *player* yang lain akan mendapatkan notifikasi bahwa permainan sudah dimulai.

Request client selanjutnya yaitu request pengiriman paket ke grup yang sama. *Request* ini merupakan inti dari *realtime multiplayer game* ini. *Client* mengirimkan data posisi player dan pesan singkat melalui *request* DATAGRUP. Pada *request* ini server meneruskan paket yang dikirim salah satu *client* ke *client* lainnya dalam suatu grup. *Request* yang terakhir yaitu CLOSED, digunakan pada saat *player* keluar atau selesai memainkan *game*.

Ketika *player* selesai bermain, server akan menghapus data *player* tersebut dari *database* dan *HashMap*.

2. *Multithreading* pada *Client* (*Player*)

Client dalam *game* ini juga mengimplementasikan *multithreading*. Penerapan *multithreading* pada *client* hampir sama seperti server. *Client* menggunakannya untuk menerima paket yang dikirim oleh server. Karena pengiriman paket ini dapat berlangsung setiap waktu, maka *multithreading* digunakan untuk menerima dan memproses paket tersebut sehingga *client* dapat menampilkannya pada *interface game*. Paket yang diproses *client* adalah paket dengan dataID berikut.

Tabel 4.2 Pemrosesan Data pada *Client*

Data ID/Header	Isi Data
PLAYERLIST	Daftar *player* dalam sebuah grup
PLAY	*Request* memulai *game*
DATAGRUP	Data posisi *player* yang sedang bermain
DATAGRUP:#AKSI#	Data posisi *player* yang melakukan aksi pada ular atau tangga
DATAGRUP:#CHAT#	Pesan singkat yang dikirim *player* dalam grup

Paket dengan data Id PLAYERLIST berisi data *player* dalam grup. Struktur paket ini adalah `PLAYERLIST:player1#player2#player3#....` Paket ini dipecah sebanyak 2 kali. Pertama memecah paket untuk mengetahui data Id-nya, kedua memecah subpaket yang berisi daftar *player*. *Source Code* pemrosesan datanya ditunjukkan pada Modul Program 4.6.

Modul Program 4.6 *Source Code* Pemrosesan Data Id PLAYERLIST

```
DataInputStream is = new DataInputStream(socket.getInputStream());
ArrayList plist = new ArrayList();
while ((dariSocket = is.readLine()) != null) {
String data[] = dariSocket.split(":");
   if (data [0].equals("#PLAYERLIST#")) {
```

```
      playerlist = data [1];
      String[] arr = playerlist.split("#");
      for (int i = 0; i < arr.length; i++) {
         if (plist.contains(arr[i])) {
      } else {
         plist.add(arr[i]);
      }
   }
}
```

Daftar *player* dipecah menggunakan `split("#")` dan selanjutnya disimpan dalam *ArrayList*. Paket dengan data Id PLAY merupakan paket yang berisi instruksi untuk memunculkan notifikasi bahwa *game* telah dimulai. Paket ini dikirim oleh player yang memulai game kemudian server mem-*broadcast* paket ini kepada *player* yang berada dalam grup yang sama. Sehingga *player* dapat memainkan *game* secara bersamaan. Gambar 4.10 menunjukkan hasil dari *request* PLAYERLIST dan PLAY.

Gambar 4.10 Notifikasi Mulai Bermain

Paket DATAGRUP berisi data posisi *player*. Struktur paketnya adalah `nama_grup:username:posisi_awal:posisi_tujuan`. Kemudian *client* memecah paket dan mengambil data *user* dan posisinya untuk diterapkan

pada *interface game*. S*ouce code* pengambilan data posisi *player* ditunjukkan pada Modul Program 4.7.

Modul Program 4.7 *Source Code* Pengambilan Data Posisi *Player*

```
PrintStream os = new PrintStream(socket.getOutputStream());
DataInputStream is = new DataInputStream(socket.getInputStream());
while ((dariSocket = is.readLine()) != null) {
   String data[] = dariSocket.split(":");
      int posawal = Integer.parseInt(data[3]);
      int postujuan = Integer.parseInt(data[2]);
   if (data[1].equals(plist.get(0))) {
      playerpos(data[1], PlayerIcon1, posawal, postujuan);
   } else if (data[1].equals(plist.get(1))) {
      playerpos(data[1], PlayerIcon2, posawal, postujuan);
   } else if (plist.size()>=3 && data[1].equals(plist.get(2))){
      playerpos(data[1], PlayerIcon3, posawal, postujuan);
   } else if (plist.size()==4 && data[1].equals(plist.get(3))){
      playerpos(data[1], PlayerIcon4, posawal, postujuan);
   {
   ....
}
```

Data paket di pecah kemudian disimpan dalam array `datamasuk[]`. Untuk mengetahui siapa *player* yang bermain, diambil dari data *username* yaitu `data[1]`. Kemudian *username* di cocokkan dengan daftar *player* dalam grup (`plist`). Jika data cocok maka dapat diketahui `data[2]` dan `data[3]` adalah data posisi awal dan posisi tujuan *player* tersebut. Data ini diolah kembali dalam *method* `playerpos()` untuk menerapkan posisi *player* pada papan *game* (*interface*).

DATAGRUP:#AKSI# merupakan data posisi *player* ketika melakukan aksi (pergerakan) pada ular atau tangga. DATAGRUP tidak hanya berisi data posisi *player* saja. DATAGRUP:#CHAT# berisi pesan singkat yang dikirim *player* ke grup. Struktur paketnya adalah `nama_grup:#CHAT#:username>> pesan`. Data kemudian diproses, dan pesan

ditampilkan pada *textArea* khusus. Berikut source code yang digunakan untuk pemrosesan data pesan (Modul Program 4.8).

Modul Program 4.8 *Source Code* Pemrosesan Data Pesan Singkat (*Chat*)

```
if (datamasuk[1].equals("#CHAT#")) {
   taChatBox.append(datamasuk[2] + "\n");
}
```

Dalam implementasi *interface, multitherading* digunakan untuk animasi jalannya pion *player*. Pion *player* berjalan perkotak/blok. Untuk memberikan efek animasi, digunakan *thread* untuk mengatur *timer* jalannya pion *player* yang diimplementasikan pada *privat class* Animasi.

Modul Program 4.9 *Source Code Class* Animasi

```
private class Animasi extends javax.swing.JPanel implements
Runnable {
   JLabel iconplayer = new JLabel();
   int posawal = 0, postujuan = 0, x, y;
   String uname;
   public Animasi(String uname, int posawal, int postujuan, JLabel
   label) {
      this.posawal = posawal;
      this.postujuan = postujuan;
      this.iconplayer = label;
      this.uname = uname;
   }
   @Override
   @SuppressWarnings("SleepWhileInLoop")
   public void run() {
      while (posawal <= postujuan && postujuan <= 100) {
         if (posawal <= 10) {
            x = 27 + ((posawal - 1) * 55);
            y = 575;
            iconplayer.setLocation(x, y);
            x += 55;
         } else if (posawal >= 11 && posawal <= 20) {
            x = 525 - ((posawal - 11) * 55);
            y = 520;
            iconplayer.setLocation(x, y);
            x -= 55;
         }....
         ....
         try {
            Thread.sleep(300);
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
         posawal++;
      }}}
```

Class Animasi (Modul Program 4.9) memiliki *constructor* dengan *input username*, posisi awal, posisi tujuan, dan label pion *player*. Posisi *player* ditampilkan pion pada papan permainan. Selama posisi awal kurang dari posisi tujuan, pion akan terus berjalan. Posisi awal akan terus bertambah selama perulangan berlangsung (`posawal++`). Dan perulangan akan berhenti ketika posisi awal sama dengan posisi tujuan. Untuk menambahkan efek pergerakan pion *player* maka ditambahkan `Thread.sleep(300)`. *Method* `sleep()` pada *thread* berfungsi untuk menangguhkan eksekusi dari *thread* yang sedang berjalan untuk sementara waktu. 300 menunjukkan waktu jeda *thread* sebanyak 3 *milisecond.*

4.3 Pengujian

Uji coba pada sistem terdiri dari dua tahap, yaitu pengujian aplikasi dan pengujian server sebagai sistem moderator grup pemain.

4.3.1 Pengujian Aplikasi

Pengujian aplikasi server dan client *game* ini dilakukan secara *black-box*. Pengujian yang dilakukan bertujuan untuk mengamati hasil eksekusi dan memeriksa fungsional dari *server* dan *game*. Pengujian ini dilakukan dengan mengamati jalannya aplikasi server dan *client* game. Secara umum hasil yang didapatkan dari pengujian aplikasi ditunjukkan pada tabel 4.3.

Tabel 4.3 Hasil Pengujian Aplikasi

Skenario Pengujian	Hasil yang diharapkan	Kesimpulan
Server membuka koneksi (*Player* request koneksi)	Muncul data *player login* di monitor server	Valid
Server menutup koneksi	Semua data yang tersimpan	Valid

Skenario Pengujian	Hasil yang diharapkan	Kesimpulan
	dihapus	
Player login (Data yang diinput tidak lengkap)	Sistem tidak akan menyimpan data player dan muncul notifikasi "Data belum lengkap"	Valid
Player bergabung dengan grup yang tersedia (*Player* klik daftar grup)	Muncul pilihan (Ya/Tidak) "Apakah ingin bergabung dengan grup?"	Valid
Player membuat grup baru (nama grup tidak diisi)	Grup tidak disimpan dan muncul notifikasi (Nama grup harus diisi)	Valid
Muncul soal kuis aritmatika (*Player* menginput selain angka)	*Textfield* tidak mendeteksi	Valid
Muncul soal kuis (*Player* tidak menjawab)	Selama 10 detik *pop-up* soal otomatis menutup dan jawaban diangap salah	Valid
Player mengirim pesan singkat	Semua *player* dalam grup yang sama menampilkan pesan singkat tersebut	Valid
Player menyelesaikan *game* (menang)	Lawan mendapatkan informasi bahwa *player* x telah menang	Valid
Player keluar dari *game*	Muncul pilihan (Ya/Tidak) "Apakah ingin keluar dari *game*"?	Valid

Pengujian aplikasi dilakukan pada fungsi-fungsi utama yang terdapat pada aplikasi server dan aplikasi *client*. Dari skenario pengujian yang telah disediakan, Aplikasi diuji fungsionalitasnya. Ketika hasil pengujian sama dengan hasil yang diharapkan, maka pengujian tersebut telah valid. Pengujian aplikasi yang terdapat pada Tabel 4.3 menunjukkan hasil yang valid. Hal ini menunjukkan bahwa aplikasi server dan game *online* Ular Tangga edukatif ini dapat berjalan dengan baik.

4.3.2 Pengujian Server sebagai Moderator Grup Pemain

Untuk mendapatkan server yang optimal dalam pemrosesan dan pengiriman data pada *client-client* dalam grup yang tepat, dilakukan pengujian kinerja server. Pengujian ini menerapkan sistem pengujian *white-box* untuk menganalisa detil proses. Dalam pengujian yang dilakukan, terdapat beberapa prosedur dan persiapan pengujian yang harus dilakukan.

1. **Prosedur Pengujian**

   Pengujian server sebagai moderator grup pemain dilakukan dengan pengamatan dan analisa kinerja server dalam menerima *request client* dan mem-*broadcast* data ke beberapa *client* dengan grup yang tepat. Pengujian dilakukan dengan menjalankan sebuah server dan banyak *client* yang terdiri dari beberapa grup yang berbeda. Dalam satu sesi pengujian terdiri dari 1 server dengan client yang tiap grupnya terdiri dari 2 player, 3 player dan 4 player.

2. **Kebutuhan Perangkat**

   Perangkat yang diperlukan dalam pengujian ini adalah sebagai berikut.

   a. **Perangkat Server**

      Perangkat server merupakan komponen utama dalam sistem *game*. Dalam sistem ini dilakukan pengujian server dengan menggunakan beberapa perangkat dengan spesifikasi yang berbeda. Daftar perangkat dan spesifikasinya ditunjukkan pada Tabel 4.4.

Tabel 4.4 Daftar Spesifikasi Perangkat Server

No	*Processor*	*RAM*	Tipe Sistem
1	AMD E1-1200 APU with Radeon(tm) HD Graphics 1.40 GHz.	2,00 GB	32-bit *Operating System*
2	AMD E-300 APU with Radeon(tm) HD Graphics 1.30 GHz.	4.00 GB	64-bit Operating System
3	Intel(R) Core(TM) i3-4030U CPU 1.90 GHz.	4,00 GB	64-bit Operating System
4	Intel(R) Core(TM) i7-7700HQ CPUCPU 2.81 GHz.	8,00 GB	64-bit Operating System

Dengan menggunakan beberapa variasi perangkat server, dapat diketahui perangkat server yang optimal untuk menjalankan server pada *game*.

b. Perangkat Client

Client dalam sistem ini adalah *pemain* game (*player*). Perangkat *client* digunakan untuk menjalankan aplikasi *game*. Dalam pengujian ini, perangkat *client* tidak ditentukan secara spesifik. Hanya saja untuk standar minimum, penulis menggunakan perangkat dengan kapasitas memori (*RAM*) 2 GB.

3. Hasil Pengujian

Setelah dilakukan beberapa kali pengujian, didapatkan hasil sebagai berikut.

a. Hasil Pengujian Server 1

Spesifikasi Server:

Processor : AMD E1-1200 APU with Radeon(tm) HD Graphics 1.40 GHz.
RAM : 2,00 GB
Tipe Sistem : 32-bit *Operating System*

Tabel 4.5 Hasil Pengujian Server 1

Pengirim	Server	Penerima																	
		Grup A		Grup B		Grup C			Grup D			Grup E				Grup F			
		A1	A2	B1	B2	C1	C2	C3	D1	D2	D3	E1	E2	E3	E4	F1	F2	F3	F4
A1	v	v	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	v	v	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C1	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
C2	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
C3	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
D1	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
D2	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
D3	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
E1	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
E2	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
E3	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
E4	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
F1	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	v	v	v
F2	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	v	v	v
F3	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	v	v	v
F4	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	v	v	v

Keterangan:

v : data paket terkirim - : tidak menerima data paket

x : data paket tidak terkirim **xv** : data paket tidak lengkap

b. Hasil Pengujian Server 2

Spesifikasi Server:

Processor : AMD E-300 APU with Radeon(tm) HD Graphics 1.30 GHz.

RAM : 4,00 GB

Tipe Sistem : 64-bit Operating System

Tabel 4.6 Hasil Pengujian Server 2

Pengirim	Server	Penerima																	
		Grup G		Grup H		Grup I			Grup J			Grup K				Grup L			
		G1	G2	H1	H2	I1	I2	I3	J1	J2	J3	K1	K2	K3	K4	L1	L2	L3	L4
G1	v	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G2	v	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H1	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H2	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I1	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
I2	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
I3	v	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-	-	-	-
J1	v	-	-	-	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-
J2	v	-	-	-	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-
J3	v	-	-	-	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-
K1	v	-	-	-	-	-	-	-	-	-	-	x	v	v	v	-	-	-	-
K2	v	-	-	-	-	-	-	-	-	-	-	x	v	v	v	-	-	-	-
K3	v	-	-	-	-	-	-	-	-	-	-	x	v	v	v	-	-	-	-
K4	v	-	-	-	-	-	-	-	-	-	-	x	v	v	v	-	-	-	-
L1	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v

Pengirim	Server	Penerima																	
		Grup G		Grup H		Grup I			Grup J			Grup K				Grup L			
		G1	G2	H1	H2	I1	I2	I3	J1	J2	J3	K1	K2	K3	K4	L1	L2	L3	L4
L2	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
L3	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
L4	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v

Keterangan:
v : data paket terkirim - : tidak menerima data paket
x : data paket tidak terkirim xv : data paket tidak lengkap

c. Hasil Pengujian Server 3

Spesifikasi Server:
Processor : Intel(R) Core(TM) i3-4030U CPU 1.90 GHz.
RAM : 4,00 GB
Tipe Sistem : 64-bit *Operating System*

Tabel 4.7 Hasil Pengujian Server 3

Pengirim	Server	Penerima																	
		Grup M		Grup N		Grup O			Grup P			Grup Q				Grup R			
		M1	M2	N1	N2	O1	O2	O3	P1	P2	P3	Q1	Q2	Q3	Q4	R1	R2	R3	R4
M1	v	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M2	v	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N1	v	-	-	xv	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N2	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
O1	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-

Pengirim	Server	Penerima																	
		Grup M		Grup N		Grup O			Grup P			Grup Q				Grup R			
		M1	M2	N1	N2	O1	O2	O3	P1	P2	P3	Q1	Q2	Q3	Q4	R1	R2	R3	R4
O2	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-
O3	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-
P1	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
P2	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
P3	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
Q1	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
Q2	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
Q3	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
Q4	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
R1	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
R2	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
R3	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
R4	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v

Keterangan:
v : data paket terkirim
x : data paket tidak terkirim
- : tidak menerima data paket
xv : data paket tidak lengkap

d. Hasil Pengujian Server 4

Spesifikasi Server:

Processor : Intel(R) Core(TM) i7-7700HQ CPUCPU 2.81 GHz.
RAM : 8,00 GB
Tipe Sistem : 64-bit *Operating System*

Tabel 4.8 Hasil Pengujian Server 4

Pengirim	Server	Penerima																	
		Grup S		Grup T		Grup U			Grup V			Grup W				Grup X			
		S1	S2	T1	T2	U1	U2	U3	V1	V2	V3	W1	W2	W3	W4	X1	X2	X3	X4
S1	v	v	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
S2	v	v	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T1	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T2	v	-	-	x	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-
U1	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-
U2	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-
U3	v	-	-	-	-	v	v	v	-	-	-	-	-	-	-	-	-	-	-
V1	v	-	-	-	-	-	-	-	xv	v	v	-	-	-	-	-	-	-	-
V2	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
V3	v	-	-	-	-	-	-	-	x	v	v	-	-	-	-	-	-	-	-
W1	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
W2	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
W3	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
W4	v	-	-	-	-	-	-	-	-	-	-	v	v	v	v	-	-	-	-
X1	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
X2	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
X3	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v
X4	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	v	v	v	v

Keterangan:

v : data paket terkirim - : tidak menerima data paket

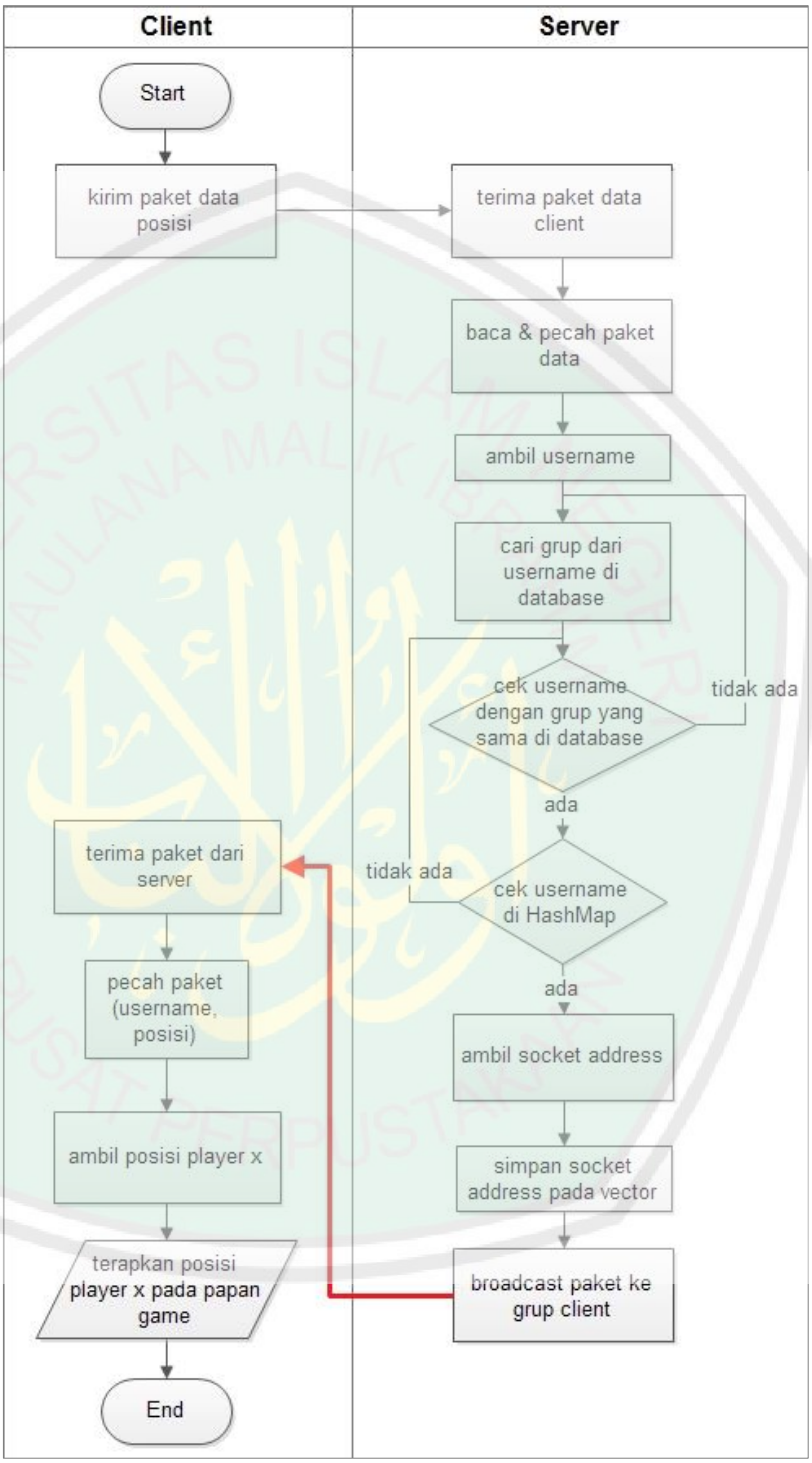
x : data paket tidak terkirim xv : data paket tidak lengkap

Pengujian yang telah dilakukan menunjukkan hasil yang bervariasi. Pada saat pengujian dengan menggunakan server 1 (Tabel 4.5), terdapat cukup banyak paket yang tidak terkirim ke *client* (*player*). Begitu pula ketika pengujian dikaukan dengan server 2, terdapat paket yang tidak diterima oleh beberapa *client* seperti yang ditunjukkan pada Tabel 4.6. Perangkat yang digunakan server 1 dan server 2 memiliki spesifikasi *processor* yang sama, tetapi *memory* (RAM) server 2 lebih besar dibandingkan server 1. Namun, perbedaan *memory* disini tidak memberikan dampak yang signifikan terhadap pengiriman data paket.

Ketika spesifikasi perangkat server dinaikkan lagi, yaitu server 3, hasil pengiriman data lebih baik dibandingkan pengujian sebelumnya (Tabel 4.7). Kemudian pengujian dilanjutkan dengan menggunakan perangkat server yang lebih tinggi lagi untuk mengurangi kegagalan pengiriman paket data. Dan hasilnyapun lebih baik lagi. Hasil pengujian dengan server 4 terdapat pada Tabel 4.8. Dari 18 client hanya terdapat 2 client yang paket datanya tidak terkirim. Hal ini menunjukkan bahwa semakin tinggi spesifikasi perangkat server, maka semakin baik pula kinerja sistem server.

Jika dilihat dari keseluruhan hasil pengujian yang telah dilakukan, rata-rata paket data yang tidak terkirim adalah paket yang diterima oleh *client* 1 atau pemain pertama. Apabila ditelusuri dengan pengujian white box dengan mengamati data masuk dan keluar pada server, hal ini disebabkan oleh paket data awal yang dikirim oleh *client* 1 gagal diterima kembali oleh *client* tersebut, sehingga menyebabkan paket data yang masuk selanjutnya

tidak dapat diterima dan diproses. Detail proses ditunnjukkan pada Gambar 4.11.



Gambar 4.11 *Flowchart* Kegagalan Pemrosesan Data

Anak panah berwarna merah pada Gambar 4.11 menunjukkan adanya kegagalan proses. Sebenarnya, *client* tersebut telah berhasil mengirimkan paket data ke server, dan server juga telah mem-*broadcast* paket tersebut kepada *client* dalam grup yang sama. Namun *client* tersebut tidak bisa atau gagal menerima paket. Proses ini akan terus berlanjut sampai akhir. Sehingga dapat mengganggu jalannya *game multiplayer*.

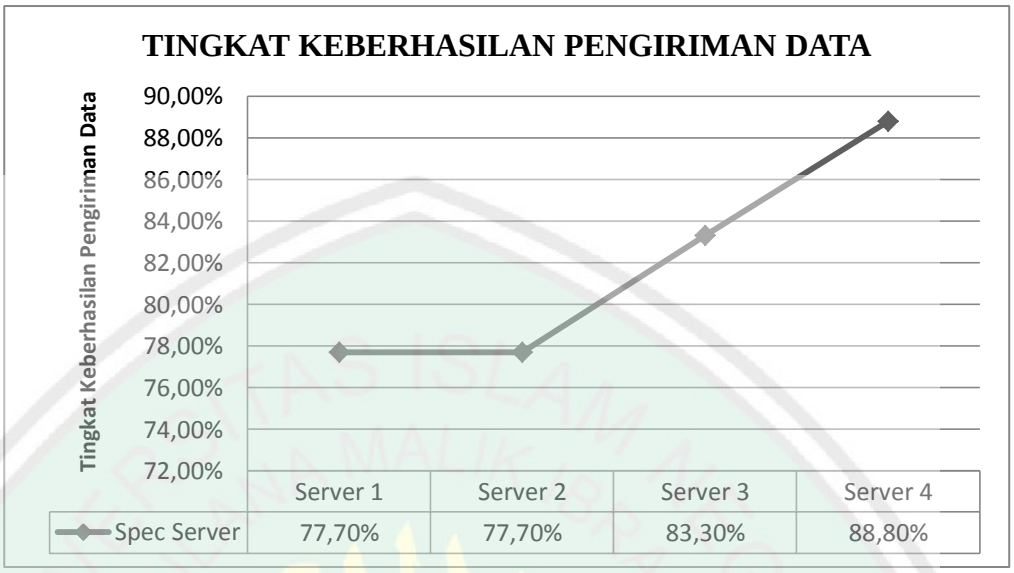
Pada hasil pengujian juga terdapat data paket yang diterima *client* tidak lengkap. Hal ini biasanya disebabkan oleh pengolahan paket pada client atau server yang menjadikan data *corrupt*, sehingga data paket yang dikirim tidak lengkap. Data *corrupt* dapat mempengaruhi pengiriman atau penerimaan paket selanjutnya. Namun kasus ini jarang terjadi pada sistem.

Walaupun terdapat beberapa kegagalan pengiriman paket, hasil pengujian secara umum menunjukkan keberhasilan server sebagai moderator grup pemain. Hasil pengujian menunjukkan sama sekali tidak ada data paket grup yang salah kirim ke grup yang lain. Paket hanya dikirim kepada *client-client* yang berada dalam grup yang sama dengan pengirim saja. Untuk mengukur persentase keberhasilan pengiriman data dihitung dengan rumus:

$$\text{Keberhasilan pengiriman data} = \frac{\sum \text{keberhasilan pengiriman} \times 100\%}{\sum \text{seluruh pengiriman data}}$$

Hasil pengujian menggunakan server 1 dan server 2 menunjukkan 77,7% keberhasilan pengiriman data. Untuk pengujian dengan menggunakan server 3 hasilnya adalah 83.3% keberhasilan. Dan pengujian dengan server 4, keberhasilan pengiriman data mencapai 88.8%. Dari keseluruhan pengujian yang telah dilakukan didapatkan rata-rata tingkat keberhasilan pengiriman

data sebesar 83,2%. Gambar 4.12 menunjukkan grafik keberhasilan pengiriman data menurut spesifikasi server.



Gambar 4.12 Grafik Tingkat Keberhasilan Pengiriman Data

Seperti yang telah dijelaskan pada Tabel 4.4 sebelumnya, tiap-tiap server mempunyai spesifikasi perangkat yang berbeda-beda. Server 1 memiliki spesifikasi perangkat paling rendah. Spesifikasi perangkat server dinaikkan pada pada Server 2, berlanjut ke Server 3, dan Server 4. Setelah dilakukan pengujian server menunjukkan hasil yang berbeda-bada. Grafik pada Gambar 4.12 menunjukkan bahwa server 4 dengan spesifikasi perangkat yang lebih baik memiliki tingkat keberhasilan pengiriman data tertinggi. Hal ini menunjukkan bahwa semakin tinggi spesifikasi perangkat server, semakin tinggi pula keberhasilan pengiriman datanya.

4.4 Integrasi Nilai Islam

Allah adalah sebaik-baik pengatur. Segala sesuatu yang ada di dunia ini pasti memiliki ketetapan dan aturannya masing-masing. Allah ﷻ berfirman dalam Surah Al-Anbiya ayat 33.

وَهُوَ ٱلَّذِى خَلَقَ ٱلَّيۡلَ وَٱلنَّهَارَ وَٱلشَّمۡسَ وَٱلۡقَمَرَۖ كُلٌّ فِى فَلَكٍ يَسۡبَحُونَ ﴿٣٣﴾

"Dan Dialah yang telah menciptakan malam dan siang, matahari dan bulan. Masing-masing dari keduanya itu beredar di dalam garis edarnya"

Dalam Tafsir Ibnu Katsir dijelaskan, وَٱلشَّمۡسَ وَٱلۡقَمَرَ ("Matahari dan bulan,") matahari memiliki cahaya yang khusus, ruang edar sendiri, masa yang terbatas serta gerakan dan perjalanan khusus. Sedangkan bulan dengan cahaya lain, ruang edar lain, perjalanan lain dan ukuran lain. كُلّٞ فِي فَلَكٖ يَسۡبَحُونَ ("Masing-masing dari keduanya itu beredar di dalam garis edarnya,") yaitu mereka beredar. Ibnu `Abbas berkata: "Mereka beredar sebagaimana tenunan beredar di alat putarannya." Mujahid berkata: "Tenunan tidak beredar kecuali di alat putarannya dan tidak ada alat putaran kecuali dengan tenunannya. Demikian pula dengan bintang-bintang, matahari dan bulan tidak beredar kecuali dengan alat edarnya dan alat edarnya tidak berputar kecuali dengan semua itu."

Kemudian dalam surah Yasin ayat 40, Allah ﷻ berfirman:

لَا ٱلشَّمۡسُ يَنۢبَغِى لَهَآ أَن تُدۡرِكَ ٱلۡقَمَرَ وَلَا ٱلَّيۡلُ سَابِقُ ٱلنَّهَارِۚ وَكُلٌّ فِى فَلَكٍ يَسۡبَحُونَ ﴿٤٠﴾

"Tidaklah mungkin bagi matahari mendapatkan bulan dan malampun tidak dapat mendahului siang. Dan masing-masing beredar pada garis edarnya"

Dr. Quraish Syihab dalam tafsirnya menjelaskan mengenai ayat ini bahwa matahari tidak akan melenceng dari tata aturannya sehingga mendahului bulan dan masuk dalam peredarannya. Demikian pula malam, tidak akan mendahului siang dan menghalangi kemunculannya. Akan tetapi siang dan malam itu selalu silih berganti. Baik matahari, bulan dan lainnya senantiasa beredar dalam garis edarnya dan tidak pernah melenceng.

Ayat-ayat tersebut membuktikan bahwa Allah ﷻ telah mengatur segala sesuatu yang ada di dunia ini dengan sangat detail. Pengaturan ini akan senantiasa tetap teraratur sampai datangnya hari kiamat. Dengan adanya pengaturan, dunia ini berjalan dengan baik. Jika kita bayangkan alam ini berjalan sendiri, tanpa ada pengaturan dari Allah, tidak ada garis edar, pastilah dunia ini sudah hancur.

Oleh karena itu, adanya pengaturan merupakan hal yang sangat penting. Supaya segala sesuatu dapat berjalan dengan baik. Sehubungan dengan ayat-ayat yang telah dijelaskan, penelitian ini berfokus pada sistem pengaturan (moderator) grup pemain pada *game online* Ular Tangga edukatif. Dengan adanya pengaturan pada server, data-data yang dikirim *client* dan server dapat diproses dengan baik oleh sistem.

Sedangkat *game* atau permainan di dalam Al-Qur'an terdapat dalam beberapa ayat. Salah satunya Allah ﷻ dalam Surah Muhammad ayat 36 berfirman.

إِنَّمَا ٱلْحَيَوٰةُ ٱلدُّنْيَا لَعِبٌ وَلَهْوٌ ۚ وَإِن تُؤْمِنُوا۟ وَتَتَّقُوا۟ يُؤْتِكُمْ أُجُورَكُمْ وَلَا يَسْـَٔلْكُمْ أَمْوَٰلَكُمْ ﴿٣٦﴾

"Sesungguhnya kehidupan dunia hanyalah permainan dan senda gurau. Dan jika kamu beriman dan bertakwa, Allah akan memberikan pahala kepadamu dan Dia tidak akan meminta harta-hartamu."

Dalam ayat tersebut, kehidupan dunia ini diibaratkan suatu permainan dan senda gurau. Hal-hal yang bersifat duniawi merupakan hal yang remeh, sia-sia, dan membuat hati lalai. Manusia seringkali dilalaikan oleh harta dan kekuasaannya, bermain-main dengan amal yang tidak ada faedahnya. Oleh karena itu, hal ini seharusnya membuat orang tidak melakukan amalan yang sia-sia dan lebih memperhatikan keimanan dan ketakwaannya kepada Allah. Inilah yang

seharusnya dikejar dan diberi perhatian secara serius. Allah tidak ingin membebani hambanya dengan sesuatu yang memberatkan seperti meminta dikeluarkan semua harta atau meminta banyak dari harta sehingga memadharratkan.

Dalam Tafsir Ibnu Katsir dijelaskan bahwa Allah berfirman sebagai bentuk penghinaan terhadap urusan dunia dan meremehkan terhadapnya إِنَّمَا ٱلْحَيَوٰةُ ٱلدُّنْيَا لَعِبٌ وَلَهْوٌ (“Sesungguhnya kehidupan dunia hanyalah permainan dan senda gurau”) maksudnya demikianlah hasilnya, kecuali jika dimaksudnya beribadah kepada Allah. Oleh karena itu Allah berfirman: وَإِن تُؤْمِنُواْ وَتَتَّقُواْ يُؤْتِكُمْ أُجُورَكُمْ وَلَا يَسْـَٔلْكُمْ أَمْوَٰلَكُمْ (“Dan jika kamu beriman dan bertaqwa, Allah akan memberikan pahala kepadamu dan Dia tidak akan meminta harta-hartamu”) maksudnya Dia tidak akan pernah butuh kepada kalian, Dia tidak meminta sesuatupun dari kalian. Dan Dia telah mewajibkan kepada kalian zakat dari harta kalian untuk membantu saudara-saudara kalian yang fakir dan miskin agar bermanfaat, dan pahalanya kembali kepada kalian.

Sedangkan pada Tafsir Jallalain disebutkan bahwa sesungguhnya kehidupan dunia maksudnya, menyibukkan diri dalam kehidupan dunia (hanyalah permainan dan senda gurau. Dan jika kalian beriman serta bertakwa) kepada Allah, yang demikian itu adalah termasuk perkara akhirat (Allah akan memberikan pahala kepada kalian dan Dia tidak akan meminta harta-harta kalian) semuanya, melainkan hanya zakat yang diwajibkan. Ayat-ayat lain yang serupa terdapat pada Q.S. Al-An’am ayat 32 dan Q.S. An-Ankabut ayat 64.

Dari ayat yang telah disebutkan dapat disimpulkan bahwa dunia ini hanyalah permainan yang sementara. Dengan demikian, seharusnya kita tidak bermain dengan amalan yang sia-sia. Tentu saja kehidupan dunia juga semestinya diperhatikan, namun kehidupan akhirat tetaplah menjadi prioritas. Apabila kita dapat menjalani keduanya, yaitu memperhatikan kehidupan dunia dan mempersiapkan kehidupan akhirat dengan sebaik-baiknya tentulah lebih baik. Oleh karena itu segala permainan yang kita lakukan hendaknya membawa manfaat, supaya apa yang kita perbuat tidak sia-sia. Dari penjelasan-penjelasan tersebut penulis berharap penelitian ini tidak hanya menghasilkan sebuah permainan yang sia-sia, namun juga menghasilkan inovasi yang baru dan bermanfaat bagi penggunanya.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan implementasi dan pengujian sistem yang telah dilakukan, maka dapat diambil kesimpulan bahwa pengujian aplikasi menunjukkan bahwa rancang bangun aplikasi server dan *game online* Ular Tangga Edukatif telah berhasil dan dapat dijalankan dengan baik. Kinerja server sebagai moderator grup pemain yang telah diuji menggunakan beberapa perangkat dengan spesifikasi yang bervariasi menunjukkan tingkat keberhasilan pengiriman data yang berbeda-beda. Hasil pengujian server 1 dan server 2 mencapai 77,7% keberhasilan pengiriman data. Pengujian dengan menggunakan server 3 menunjukkan keberhasilan sebesar 83.3%. Dan pengujian dengan server 4, keberhasilan pengiriman data mencapai 88.8%. Dari keseluruhan pengujian yang telah dilakukan didapatkan rata-rata tingkat keberhasilan pengiriman data sebesar 83,2%. Spesifikasi perangkat yang digunakan oleh server mempengaruhi tingkat keberhasilan pengiriman data. Semakin tinggi spesifikasi perangkat server, semakin tinggi pula keberhasilan pengiriman datanya.

5.2 Saran

Penelitian yang telah dilakukan ini masih sederhana dan banyak kekurangan. Perlu adanya perbaikan dan pengembangan supaya penelitian ini menjadi lebih baik, diantaranya yaitu:

1. Penggantian *database* dengan file .txt sebagai penyimpanan grup yang lebih ringan dan praktis.

2. Pengembangan *interface game* berbasis *mobile* (Android/iOS) agar lebih praktis penggunaannya.

DAFTAR PUSTAKA

Al-Mahally, Imam Jalaluddin, Imam Jalaluddin As-suyutti. 1990. *Tafsir Jalalain Berikut Asbab An-Nuzulnya Jilid I*. Bandung: Sinar Baru.

Atkins D, dkk. *Internet Security Professional Reference*, Macmillan Computer Publishing.

Alana Ahmad, 2016. *Jenis-Jenis Game Yang Wajib Kalian Ketahui Sebagai Gamer!*, https://www.kreasitekno.com/ini-dia-jenis-jenis-game-yang-wajib-kalian-ketahui-sebagai-gamer, diakses pada 31 Juli 2017

Chatib, Munif. 2011. *Gurunya Manusia: Menjadikan Semua Anak Istimewa dan Semua Anak Juara*. Bandung: Mizan Pustaka.

Christian Isman, Albertin. 2012. *Metode Belajar dengan Permainan Ular Tangga yang Dipadukan dengan Token Ekonomi untuk Meningkatkan Kemampuan Membaca Permulaan Siswa TK B.*

De Freitas, Sara. 2006. *Learning in Immersive Worlds: A Review of Game-based Learning. Prepared for the JISC e-Learning Programme*. London.

E. Mulyasa. 2009. *Menjadi Guru Profesional Menciptakan Pembelajaran Kreatif dan Menyenangkan*. Bandung: PT. Remaja Rodakarya.

Fikri, Muhammad. 2013. *Rancang Bagun Game Batleship Multiplayer pada Jaringan LAN*. UIN Sultan Syarif Kasim Riau.

Husni. 2003. *Implementasi Jaringan Komputer dengan Linux Redhat 9*. Andi : Yogyakarta

Hurlock, Elizabeth.B. 2011. *Psikologi Perkembangan*. Jakarta: Erlangga.

HM, Jogiyanto. Pengenalan computer. Yogyakarta : penerbit Andi Yogyakarta, 2004

Ibnu Katsir. 2003. *Tafsir Ibnu Katsir, Jilid 1-7*. Bogor : Pustaka Imam Syafi'I.

Irawan, Budhi. 2005. *Jaringan Komputer.* Yogyakarta: Graha Ilmu.

Kerfs, Jeremy. 2011. *Beginning Android Tablet Games Programming*. USA: Apress Inc.

Marseleno, Andino. 2003. *Kamus Istilah Komputer dan Informatika*. https://www.ilmukomputer.com.

Mujib, Fathul, Nailul Rahmawati. 2011. *Metode Permainan-Permainan Edukatif dalam Belajar Bahasa Arab.* Yogyakarta: Diva Press.

Putri, Belia, dkk. 2016. *Perancangan Aplikasi Permainan Gobak Sodor Berbasis Flash di Lingkup Jaringan Lokal*. ResearchGate

Shihab, M. Quraish. 2002. *Tafsir Al-Mishbah (Pesan, Kesan dan Keserasian al-Qur'ān) Vol. 11*. Jakarta: Lentera Hati.

Sugiawati. 2013. *Metode Bermain Ular Tangga untuk Meningkatkan Perkembangan Kognitif Kelompok A di TK Vol. 2.* Surabaya: Publikasi UNESA Fakultas Ilmu Pendidikan.

Syahputra, Mohammad F, dkk. 2013. *Perancangan Permainan Multiplayer Truf Gembira Berbasis Android*. ResearchGate.

Yusuf, Yasin, Umi Auliya. 2011. Sirkuit Pintar Melejitkan Kemampuan Matematika dan Bahasa Inggris dengan Metode Ular Tangga. Jakarta: Visimedia.

Wijayanti, Marta Ika. 2013. *Perangkat Lunak Permainan Ular Tangga Multiplayer Berbasis Jaringan*. UIN Sunan Kalijaga Yogyakarta

Widharma, Ida B. 2016. *Game FPS dengan Menggunakan Multiplayer Game*. JIPI Volume 1, No. 1