

PROPOSAL PENELITIAN
PEMBUATAN GAME PUZZLE UNTUK SARANA PEMBELAJARAN
MENGGUNAKAN BAHASA PEMROGRAMAN JAVA

I. PENDAHULUAN

1.1. Latar Belakang

Seiring perkembangan teknologi informasi diberbagai bidang, baik dari sisi perangkat lunak maupun dari sisi perangkat keras khususnya komputer bidang multimedia. Seiring bertambahnya kebutuhan manusia maka teknologi juga berkembang untuk tujuan pembelajaran serta hiburan. Salah satu bentuknya adalah dalam dunia *game* (permainan) berbasis komputer. Baik itu *game* untuk orang dewasa maupun anak-anak. Khususnya bagian kalangan anak-anak, *game* merupakan salah satu sarana hiburan yang menarik karena berbasis visual.

Perkembangan teknologi khususnya dalam dunia *game*, seringkali dimanfaatkan sebagai sarana hiburan bagi kalangan anak-anak bahkan orang dewasa. Belakangan ini banyak pula *game-game* dengan menggunakan teknologi multimedia dan dengan *game puzzle* di dunia maya. *Game-game* dengan menggunakan teknologi multimedia relatif berukuran kecil dan tidak memerlukan spesifikasi *hardware* yang tinggi, sehingga bisa dimainkan di komputer dengan spesifikasi rendah.

Topik ini dipilih untuk memberikan suatu solusi dengan membuat suatu *game* untuk sarana pembelajaran serta sebagai sarana hiburan menggunakan media teknologi dan informasi.

Berdasarkan latar belakang diatas maka topik pada penelitian ini adalah **“Pembuatan *Game Puzzle* Untuk Saran Pembelajaran Menggunakan Bahasa Pemrograman *Java*”**. Diharapkan pembuatan *game* ini, memberikan motivasi serta inspirasi anak-anak untuk belajar.

1.1.1 Rumusan Masalah

Dari keterangan latar belakang diatas agar tidak menyimpang dari permasalahan pembuatan *game* ini, maka peneliti hanya merumuskan masalah “Bagaimana membuat suatu bentuk *game puzzle* untuk menunjang cara berpikir anak-anak serta sebagai sarana pembelajaran menggunakan bahasa pemrograman *java*?”

1.1.2 Batasan masalah

Pada penelitian ini batasan masalah yang diangkat sebagai berikut :

1. *Gameplay* yang dibuat sejenis *game puzzle*.
2. *Game* ini hanya bisa dimainkan pada *mode 1 player* saja

1.2 Tujuan dan Manfaat Penelitian

1.2.1 Tujuan Penelitian

Berdasarkan permasalahan topik diatas, tujuan dari penelitian ini adalah :

1. Mempelajari dan mengenal lebih dalam tentang pembuatan *game puzzle*.
2. Mengetahui lebih dalam tentang bahasa pemrograman yang dipakai dalam pembuatan *game puzzle*.

1.2.2 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Bagi penulis untuk menambah wawasan dan pengetahuan dalam merancang suatu *games* dengan menggunakan bahasa pemrograman *java*.
2. Bagi pengguna dari hasil penelitian ini diharapkan dapat memberikan motivasi dan pemahaman anak-anak dalam belajar.

II. TINJAUAN PUSTAKA

2.1. *Game*

Tentunya kita semua sudah tahu apa itu *game* dan pastinya kita sudah pernah memainkannya, baik di *Playstation*, komputer, bahkan di ponsel.

Menurut Jasson (2009:2), “*Game* adalah suatu system atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek di dalam game untuk suatu tujuan tertentu”.

2.1.1. Sejarah *Game*

Menurut Samuel Henry (2005:41) “Pada tahun 1960 beberapa staff dari *The Hingham Institute in Cambridge* membuat *game* pertama yang dinamakan dengan *Spacewar*. *Game* kuno ini berupa pertempuran dua pesawat antariksa yang berbentuk segitiga saling menembaki musuh”.

2.1.2. Kategori Dalam *Game*

Dalam game, terdapat tiga kategori yaitu : *Skill*, *Action* dan Strategi.

1. *Skil* dan *Action* terbagi beberapa subkategori yaitu :

a. *Combat games* (permainan perang)

Permainan berada disuatu konfrontasi secara langsung dengan musuh, pertempuran. Pemain ditantang untuk terus menghindari serangan musuh dan menghancurkan lawan sebanyak-banyaknya.

b. *Maze/Puzzle* (permainan teka-teki)

Game ini disebut juga game teka-teki. Maze game lebih cenderung mencari jalan keluar melalui jalan-jalan yang berliku-liku dan membingungkan.

c. *Sport games* (permainan olahraga)

Game yang ada dikategori ini memiliki keuntungan bahwa pemain sudah memiliki gambaran dari olahraga yang akan dimainkan. Yang diharapkan pemain dari game kategori ini adalah kualitas grafik yang sangat tinggi dan gerakan yang mendekati kenyataan.

d. *Paddle games* (permainan pantulan)

Cara memainkan *game* ini adalah dengan menggerakkan satu benda untuk mengenai benda yang lain. Cara kerjanya mirip permainan hoki.

2. *Strategy* terbagi dua subkategori yaitu :

a. *Adventures* (permainan pertualangan)

Game pertualangan adalah *game* yang tidak kalah populer dengan *game-game* kategori *skill* dan *action*. *Game* ini merupakan salah satu game yang komplek dari segi cerita dan cara bermainnya. Pemain akan melalui banyak rintangan, misteri, teka-teki. Untuk itu *game* ini membutuhkan banyak waktu.

b. *Real Time Strategy* (Permainan RTS)

Game ini adalah *game* yang paling kompleks dari pada *game* kategori manapun juga. Peraturan-peraturan didalam permainannya setebal undang-undangan pemerintahan dan menyita waktu paling sedikit tiga jam untuk memainkannya dari nol. Untuk *game* ini sulit untuk dimainkan. (Wicaksono Hidayat-detikInet, <http://www.detikInet.com>)

Game Puzzle dengan menggunakan bahasa pemrograman *java* termasuk dalam kategori *Skill*, *Action* dan *Strategy*, dalam subkategori *Maze/Puzzle* (permainan teka-teki).

2.2. **Multimedia**

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi (Hofstetter 2001). Multimedia sering digunakan dalam dunia hiburan. Selain dari dunia hiburan, Multimedia juga diadopsi oleh dunia Game.

Multimedia juga dapat diartikan sebagai penggunaan beberapa media yang berbeda dalam menyampaikan informasi berbentuk text, audio, grafik, animasi, dan video.

2.3. Bahasa Pemrograman

Bahasa pemrograman dapat dianalogikan dengan bahasa yang digunakan manusia (bahasa manusia). Sebagaimana diketahui, ada bermacam-macam bahasa manusia, seperti bahasa inggris, bahasa Indonesia, dan bahasa batak. Kumpulan instruksi dalam bahasa manusia yang berupa sejumlah kalimat dapat anda analogikan dengan suatu program. (Abdul Kadir : 2012 : 2).

2.4. Java

Java adalah bahasa pemrograman yang terkenal. *Java* banyak digunakan untuk membangun program, dirilis pertama kali pada tahun 1995 oleh *Sun Microsystem*. Penciptanya adalah James Gosling. Bahasa java merupakan bahasa pemrograman yang berorientasi pada objek, dimana pemrograman tersebut menggunakan suatu kode untuk menyusun program lebih handal dan lebih mudah dipahami (Abdul Kadir : 2012 : 56).

Program *java* ditulis menggunakan editor *teks* apa saja. Dilingkungan *Windows*, Anda bisa menggunakan editor seperti *Notepad*. Namun, mengingat dalam kegiatan pemrograman sering kali terjadi kesalahan, sebaliknya anda menggunakan editor *teks* yang dilengkapi nomor baris. Beberapa editor *teks* yang bisa diunduh secara gratis di internet adalah :

1. Notepad++
2. jEdit

2.5 Netbeans

Netbeans merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang telah banyak dipakai untuk mengembangkan program dalam bahasa java.(Rachmad Hakim S. Dan Ir. Sutarto, M.Si.:14)

2.6. My SQL

My SQL adalah *RDBMS (Relational Database Management System)*, seperti halnya *ORACLE*, *Postgresql*, *MS SQL*, dan sebagainya (MYSQL.com, 2009). *My SQL* merupakan *Relational Database Management System (RDBMS)* yaitu hubungan antar tabel yang berisi data-data pada suatu *database* (Kadir:2001).

2.7. Adobe Photoshop

Photoshop adalah merupakan format gambar standar pada aplikasinya. Format ini menyimpan memori cukup besar dalam file. Format *photoshop*, objek gambar nya tidak bisa di transfer atau di export ke program aplikasi lain. (Eppy Yunardi:11).

2.8. UML

Unified Modeling language (UML) adalah salah satu alat bantu yang sangat handal didunia pengembangan system yang berorientasi obyek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan *visual* yang memungkinkan bagi

pengembang system untuk membuat cetak biru atau visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

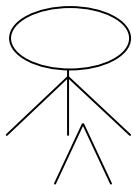
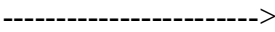
UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode *Booch* dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan interaktif, yaitu : identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih member penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian. (Munawar, 2005:17)

2.8.1. Use Case Diagram

Menurut Munawar (2005:63) *Use Case* adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan system disebut *scenario*. Setiap *scenario*

mendeskripsikan urutan kejadian. *Use case* diagram menampilkan *actor*, *use case*, dan hubungan antara mereka:

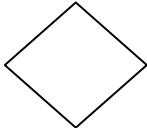
1. Aktor mana yang menggunakan *use case* mana.
2. *Use case* mana yang memasukkan *use case* lain.

No	Simbol	Keterangan
1	 <i>Aktor</i>	<i>Actor</i> merupakan pemain atau pengguna sistem yang memperhatikan himpunan atau sesuatu yang berinteraksi dengan sistem yang akan di kembangkan.
2	 <i>Use Case</i>	Menspesifikasikan fungsi dari suatu sistem.
3	 <i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
4	 <i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5	 <i>Association</i>	Apa yang menghubungkan antara objek satu dengan yang lainnya.

Tabel 1. Simbol *use case Diagram*

2.8.2. Activity Diagram

Menurut Munawar (2005 : 109) *Activity diagram* adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity Diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bias mendukung perilaku paralel sedang *flowchart* tidak bisa.

No.	Nama dan Simbol	Keterangan
1	 Start	Mendeskripsikan suatu tindakan sebelum aktivitas dimasukan.
2	 Arus Kegiatan	Mendeskripsikan ke mana aliran kegiatan.
3	 Proses / Kegiatan	Mendeskripsikan tentang suatu tindakan aktivitas, proses kegiatan.
4	 <i>Decisions</i>	Mendeskripsikan tentang suatu tindakan untuk menghasilkan keputusan.
5		Menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan

	<i>Fork</i>	dua kegiatan paralel menjadi satu.
6	 <i>Final Activity</i>	Mendeskripsikan suatu tindakan sesudah aktivitas.

Tabel 2. Simbol *Activity Diagram*

2.8.3. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel 3. *Class Diagram*

Nama Kelas
Atribut
Operasi

2.8.4. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

2.8.5. Sequence Diagram

Sequence diagram menggunakan interaksi antar objek di dalam di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

2.8.6. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki *sequence number*, di mana *message* dari level tertinggi memiliki 1, *messages* dari level yang sama memiliki prefiks yang sama.

2.8.7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*) di antaranya.

2.8.8. Deployment/Physical diagram

Deployment/Physical diagram menggambarkan detail bagaimana komponen *di-deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin,

server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal yang bersifat fisikal.

III. METODE PENELITIAN

3.1. Waktu Penelitian

Waktu penelitian dilakukan pada bulan April sampai dengan bulan Juli 2013.

3.2. Alat dan Bahan

Dalam pembuatan pembuatan *Game Puzzle* ini, alat dan bahan yang digunakan meliputi *hardware*, *software* serta bahan-bahan penunjang lainnya.

3.2.1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah laptop dengan spesifikasi berikut :

- a. Laptop acer aspire 4733Z
- b. Intel pentium processor T4500
- c. *RAM 1GB*
- d. *Hardisk 320 GB*
- e. *Flashdisk 4 GB*
- f. *Modem*

3.2.2. Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) adalah program yang digunakan untuk menjalankan perangkat keras. Tanpa adanya perangkat lunak ini komponen perangkat keras tidak dapat berfungsi, adapun aplikasi dan *software* yang digunakan dalam

pembuatan *game* ini adalah Sistem Operasi *Windows seven*, *Microsoft Office Word* 2007, *Netbeans*, dan program-program lain yang mendukung pembuatan *game*.

3.3. Metode Penelitian

Metode penelitian merupakan suatu cara yang dapat digunakan untuk mencapai tujuan yang diharapkan melalui suatu penelitian dengan teknik-teknik dan alat-alat tertentu. Adapun metode yang digunakan dalam penelitian ini yaitu metode *development research* suatu kegiatan penelitian yang bertujuan dan berusaha mengembangkan atau melengkapi pengetahuan yang sudah ada atau diketahui. Permasalahan manusia dan lingkungan alamnya selalu berkembang yang kesemuanya ini harus memperoleh jawaban yang simbang. (Supardi, 2005:25)

3.4. Metode Pengumpulan Data dan Analisis

1. Wawancara

Melakukan survey kepada kalangan anak-anak dengan cara bertanya langsung.

2. Studi Pustaka

Mencari teori dan informasi yang berhubungan dengan topik yang akan dibuat. Pencarian teori dan informasi akan dicari melalui buku-buku, internet, dan hasil penelitian maupun karya ilmiah.

3. Analisis *game* sejenis

Mencari informasi tentang aplikasi atau *game* dengan topik dan membandingkannya.

3.5. Metode Pengembangan Perangkat Lunak

Untuk metode pengembangan perangkat lunak, penulis menggunakan metode *prototyping*. “*Prototype*” adalah implementasi bagian dari produk *software* yang secara *typical* fungsinya dibatasi, realibilitas rendah, tampilannya miskin, dan kurang ketegasan. *Prototype* sering dikembangkan secara cepat dalam bahasa tingkat tinggi atau bahasa *prototype* tertentu, tanpa memperhatikan kebenaran dan ketegapan dan sebagainya. (Al Bahra Bin Ladjamudin, 2006:22)

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut:

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format\ *output*).

3. Evaluasi *protootyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah 1, 2, dan 3.

4. Mengkodekan *system*

Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji *system*

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

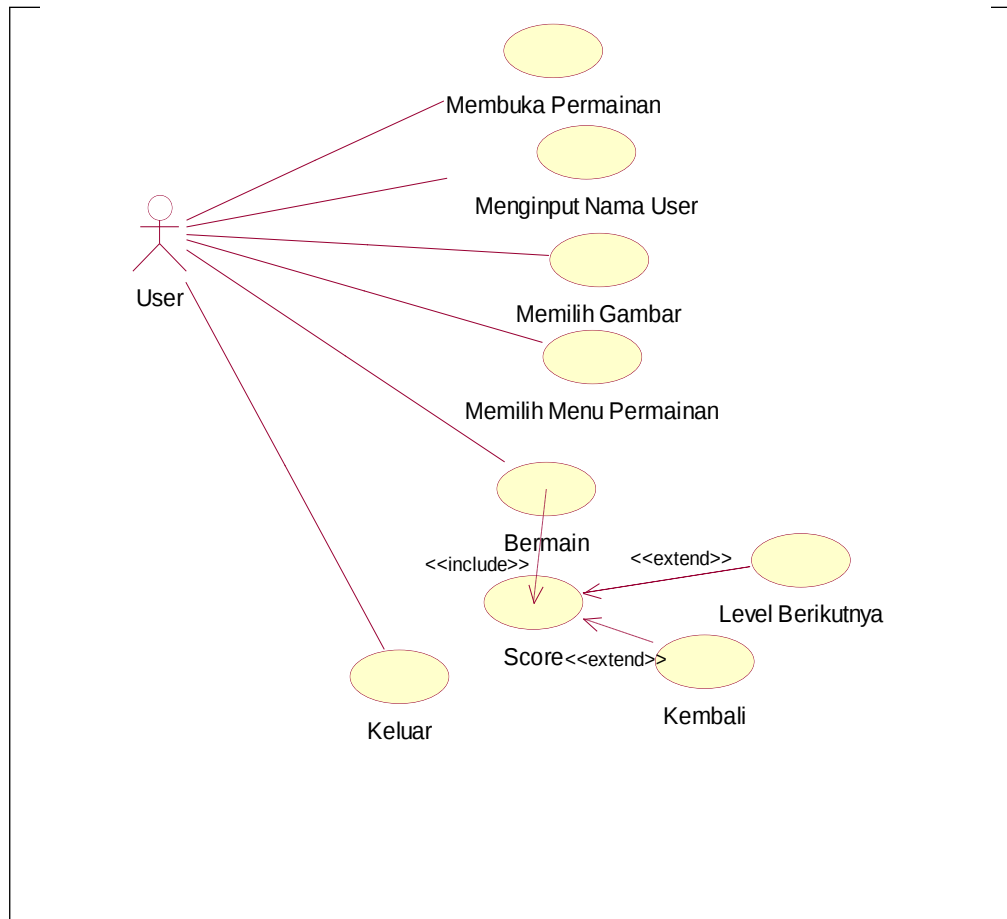
7. Menggunakan *system*

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan

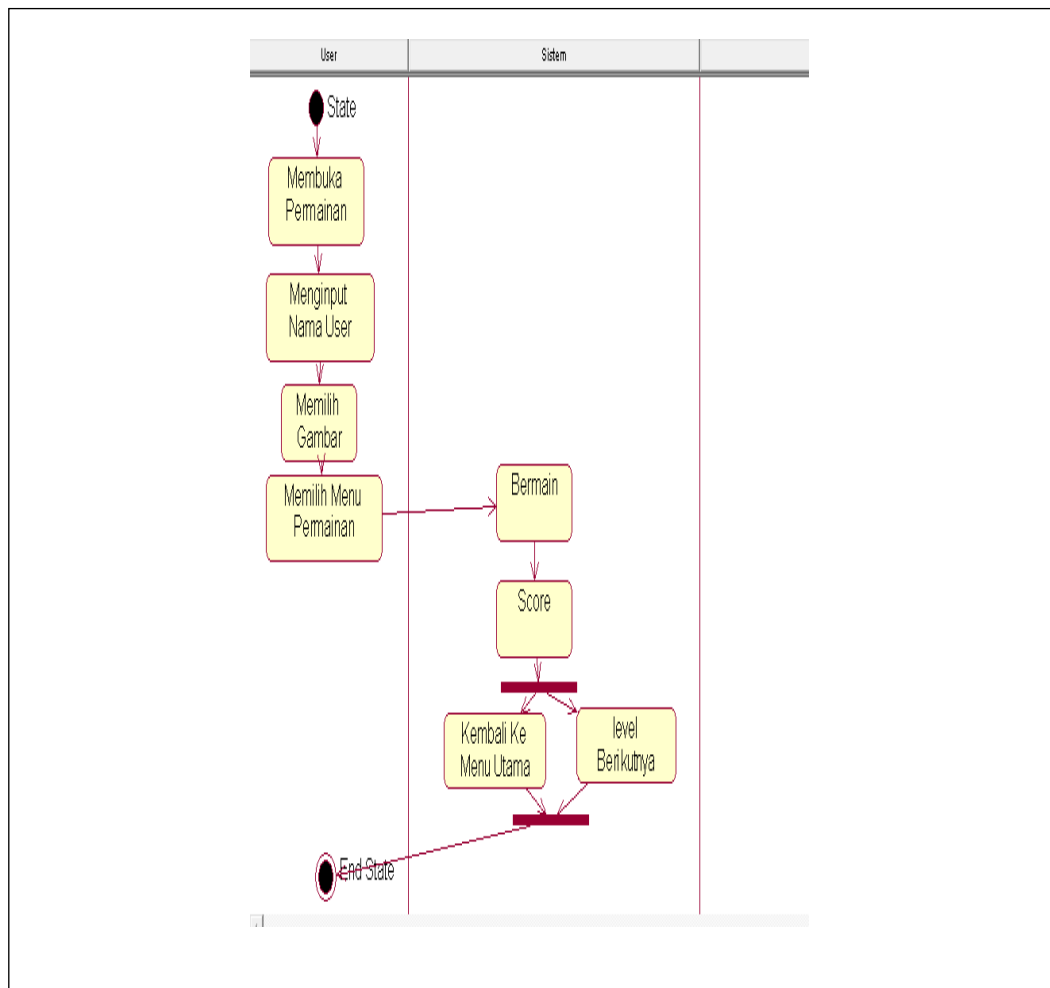
LAMPIRAN

1. Rancangan Sistem

1.1. Use Case



1.2. Diagram Activity



2. Rancangan Database

2.1. Tabel User

Filed	Type	Ukuran	Keterangan
Id_Pemain	Int	10	Primary Key
Nama_User	Varchar	45	
Skor	Int	10	
Tanggal	Date		

3. Rancangan Interface

3.1. Desain Menu Awal

Game Puzzle

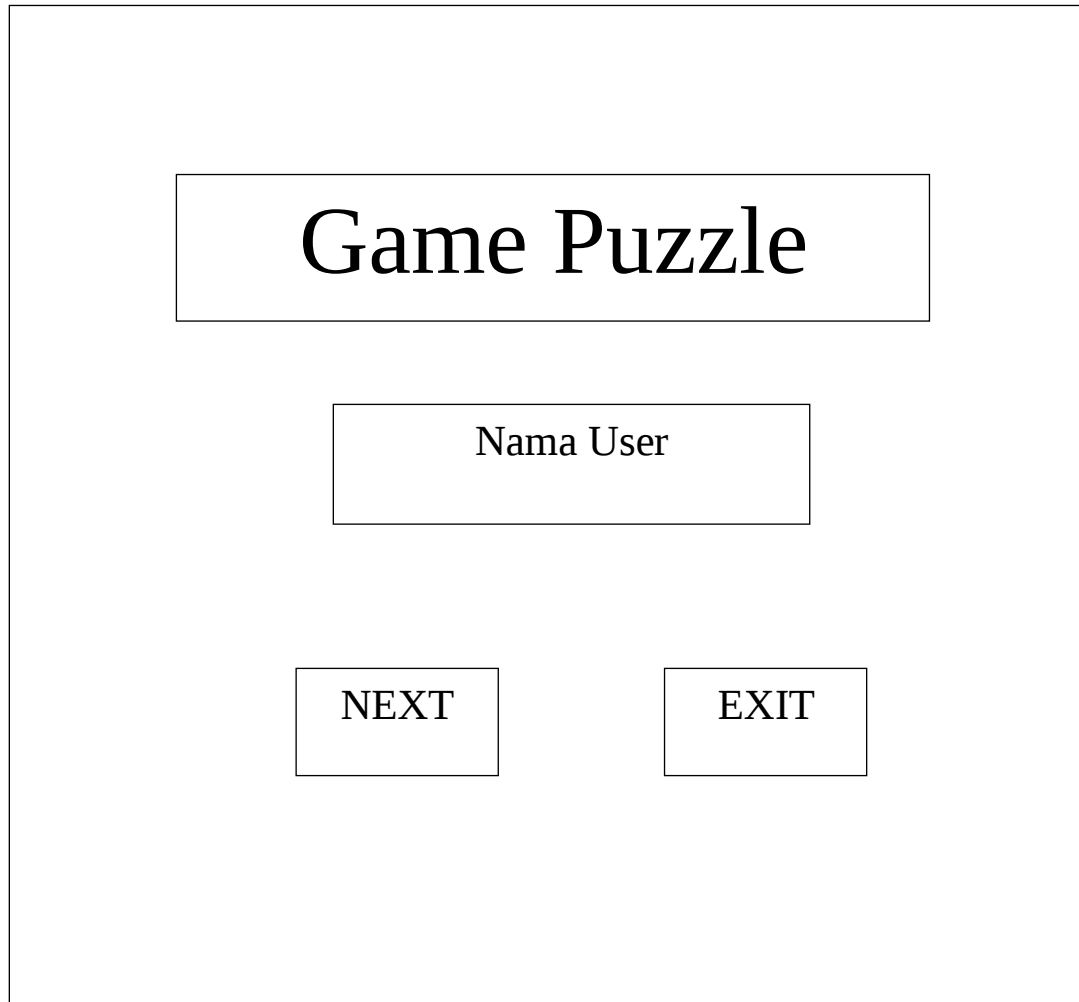
New Game

Load

Option

Exit

3.2. Desain Menu User



```
graph TD; Title[Game Puzzle] --> Input[Nama User]; Input --> Next[NEXT]; Input --> Exit[EXIT];
```

The diagram illustrates the user menu design for a game titled "Game Puzzle". It features a central rectangular frame containing three main elements: a title box at the top, a user input field in the middle, and two action buttons at the bottom. The title box, labeled "Game Puzzle", is a wide rectangle. Below it is a smaller rectangle for "Nama User", which serves as the input field. At the bottom, there are two separate rectangular buttons labeled "NEXT" and "EXIT", positioned side-by-side. All elements are centered within the main frame.

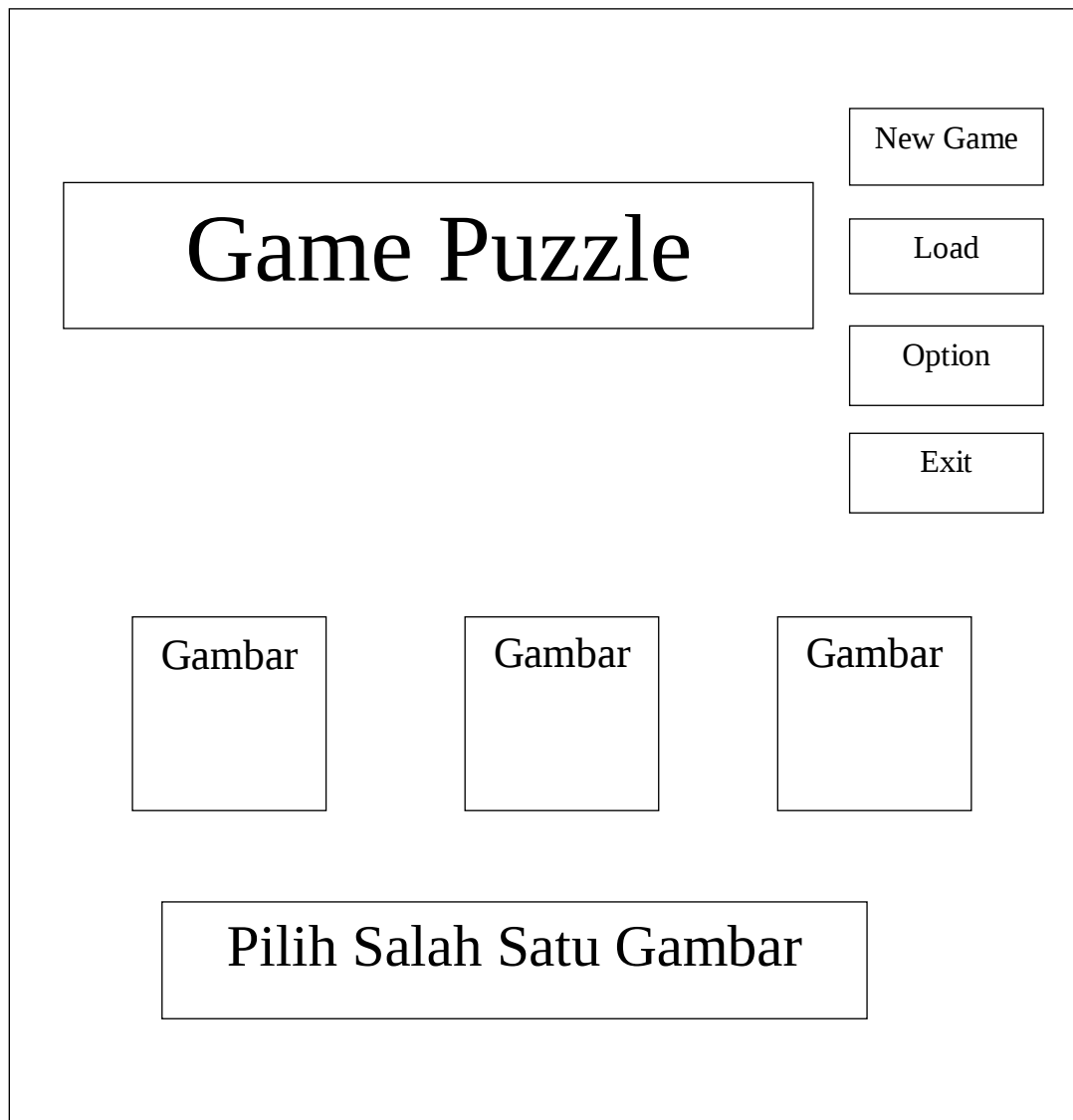
Game Puzzle

Nama User

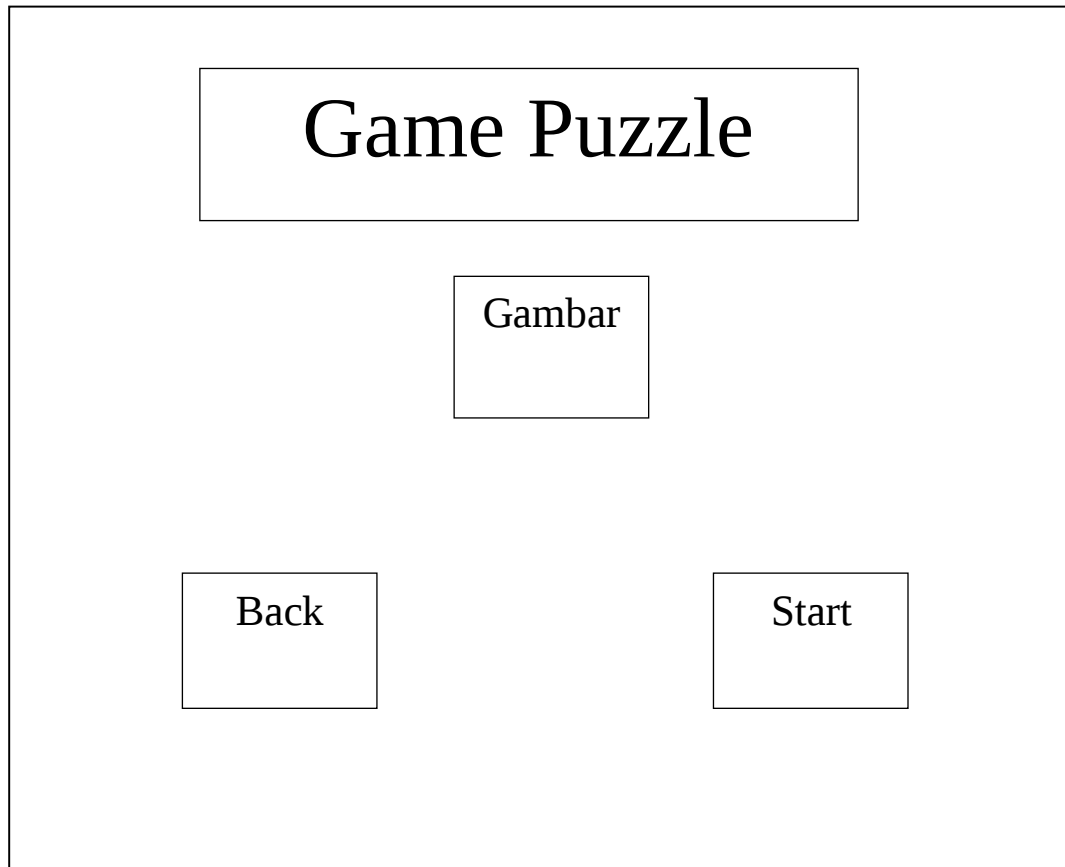
NEXT

EXIT

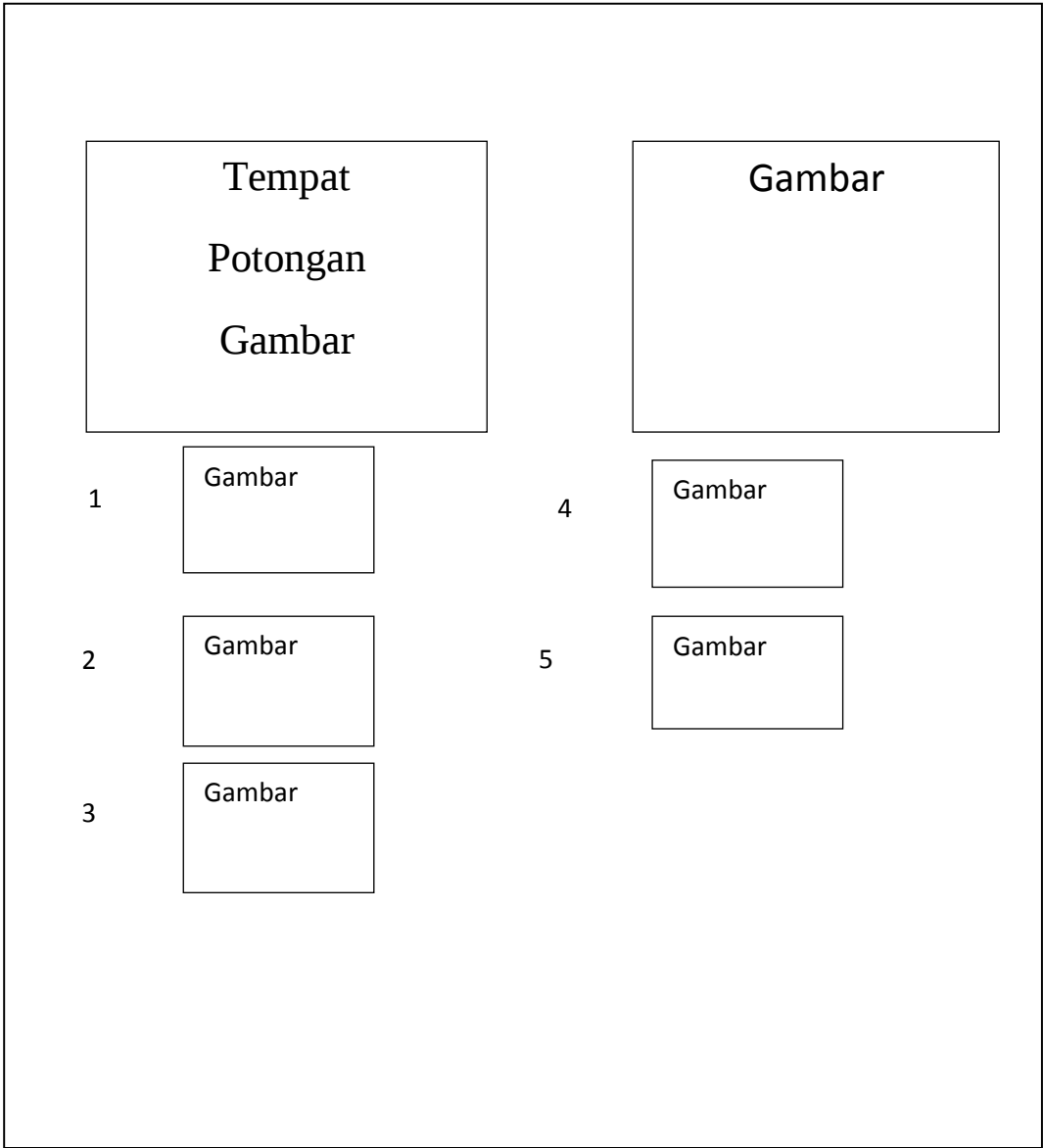
3.3. Desain Menu Utama



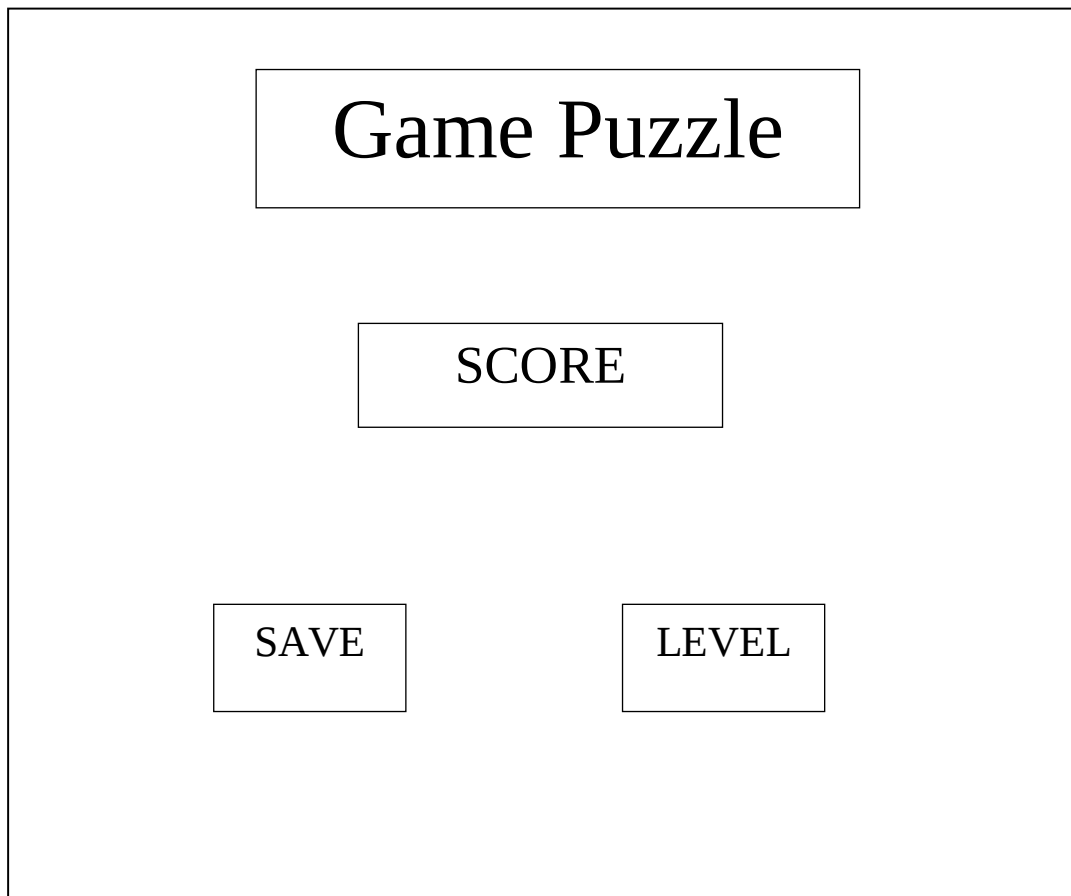
3.4. Desain Menu Memulai Permainan



3.5. **Desain Menu Bermain**



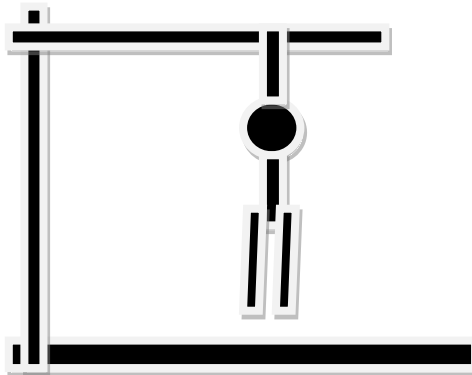
3.6. Desain Menu Skor



3.7. Desain Menu Game Over

Game Puzzle

GAME OVER



EXIT