

Pengembangan Games Coding untuk Meningkatkan Kemampuan Programming Mahasiswa

Moh. Syafri Tuloli

ABSTRAK

Matakuliah yang melibatkan pemrograman sering sekali menjadi momok bagi mahasiswa teknik informatika. Padahal kemampuan untuk membuat sistem yang merupakan salah satu kompetensi inti dari mahasiswa jurusan teknik informatika.

Salah satu penyebab kurangnya kemampuan mahasiswa dalam menguasai pemrograman adalah masalah penguasaan sintaks yang digunakan dalam bahasa-bahasa pemrograman.

Penguasaan sintaks suatu bahasa pemrograman dapat dikembangkan dengan memperbanyak jam terbang mahasiswa dalam menggunakan suatu bahasa pemrograman baik dengan memperbanyak tugas rumah (*take-home assignment*) dan juga mendorong mahasiswa untuk melakukan latihan mandiri, tetapi hal ini mendapatkan kendala karena tingginya tingkat plagiarism tugas kuliah, dan kurangnya motivasi mahasiswa untuk melakukan latihan pemrograman secara mandiri.

Penelitian ini bertujuan meningkatkan penguasaan mahasiswa akan sintaks-sintaks bahasa pemrograman yang dikemas dalam bentuk perangkat lunak games yang menarik, sehingga mahasiswa termotivasi untuk melakukan latihan mandiri, dan karena bentuk latihannya adalah dengan memainkan games maka mahasiswa tidak memiliki kesempatan untuk melakukan plagiarism.

Kata Kunci : pemrograman, perangkat lunak games

PENDAHULUAN

Latar Belakang

Matakuliah-matakuliah pemrograman telah menjadi matakuliah yang dianggap sulit oleh sebagian mahasiswa teknik informatika, bahkan menjadi penyebab tingginya tingkat dropout mahasiswa teknik informatika/Computer Science (Garcia, dkk) dan penurunan tingkat peminat pada jurusan ini (Stiller). Beberapa penyebab hal ini adalah :

1. Kompleksitas dari materi pemrograman (Garcia, dkk).
2. Kurangnya motivasi mahasiswa dalam mempelajarinya (Garcia, dkk).
3. Rendahnya standard penerimaan mahasiswa, dan kurangnya persiapan saat memasuki jenjang perkuliahan (McFarland).
4. Perkembangan teknologi IT yang sangat cepat, yang tidak terkejar oleh kurikulum yang digunakan (McFarland).
5. Cara mengajarkan yang kurang efektif (Stiller).

Dalam lingkungan teknik informatika UNG sendiri kesulitan bertambah dengan

kurangnya penguasaan mahasiswa akan bahasa Inggris yang merupakan bahasa yang dipakai pada hampir semua alat bantu dalam pengajaran pemrograman.

Salah satu permasalahan yang akan dicoba dipecahkan melalui penelitian ini adalah masalah dalam pembelajaran bahasa pemrograman. Masalah ini sebenarnya bisa diatasi dengan meningkatkan jam terbang mahasiswa dalam menggunakan bahasa pemrograman tersebut, tetapi hal ini sulit dilakukan dikarenakan:

1. Keterbatasan waktu pembelajaran (berkaitan dengan alokasi waktu matakuliah dan berkaitan dengan ketersediaan laboratorium),
2. Kurangnya minat mahasiswa untuk melakukan latihan secara mandiri dalam penggunaan bahasa pemrograman (Garcia, dkk),
3. Kurang efektifnya penggunaan tugas kuliah (*take-home assignment*) yang disebabkan oleh:
 - a. tingginya tingkat plagiarisme yang dilakukan oleh mahasiswa teknik informatika (Sheard, dkk),
 - b. keterbatasan tenaga pengajar dalam memeriksa plagiarisme (Lesner, dkk) dan memberikan feedback terhadap tugas dari mahasiswa, padahal penelitian (Garcia) menunjukkan bahwa sistem yang memberikan feedback yang relevan dapat meningkatkan antusiasme dan motivasi mahasiswa untuk belajar pemrograman.

Metode

Penelitian ini dilakukan dalam beberapa langkah, yaitu tinjauan pustaka, analisa kelemahan, kebutuhan dan kelayakan sistem, perancangan sistem, pembuatan prototipe sistem, pengujian prototipe sistem, implementasi dan sosialisasi serta evaluasi. Penjabaran dari langkah-langkah penelitian tersebut yaitu :

1. **Tinjauan pustaka**, untuk menemukan teori-teori dan penelitian sebelumnya mengenai pengajaran pemrograman. Keluaran dari tahapan ini adalah teridentifikasi permasalahan dalam pemecahan permasalahan pengajaran pemrograman dan keefektifan metode-metode yang telah diujicobakan untuk memecahkan permasalahan tersebut.
2. **Analisa**, untuk melakukan analisis kelemahan pada sistem yang telah ada, analisis kebutuhan dan kelayakan teknologi untuk pengembangan sistem dalam bentuk survey. Pada tahap ini, akan dibahas mengenai perlunya pengembangan suatu perangkat lunak untuk mempermudah dalam pemecahan permasalahan pengajaran pemrograman. Keluaran dari tahapan ini adalah daftar kebutuhan dan kelayakan dalam

pengembangan perangkat lunak pengajaran pemrograman. Target atau indikator keberhasilan pada tahap ini adalah teridentifikasinya kebutuhan dan kelayakan terhadap pengembangan perangkat lunak pengajaran pemrograman.

3. **Perancangan sistem**, untuk merancang model pemecahan permasalahan. Keluaran dari tahapan ini adalah model pemecahan, rancangan arsitektur perangkat lunak, rancangan antarmuka perangkat lunak, rancangan prosedural perangkat lunak.
4. **Implementasi prototype**, untuk pengimplementasian rancangan sistem yang telah dibuat kedalam bentuk suatu program prototype yang bisa dijalankan pada lingkungan pengembangan. Keluaran dari tahapan ini adalah terdapatnya suatu perangkat lunak yang bisa dijalankan pada lingkungan pengembangan dan dapat mensimulasikan fungsi-fungsi dari sistem.
5. **Pengujian sistem**, untuk menguji program prototype yang telah dibuat pada tahap implementasi. Keluaran dari tahapan ini adalah teridentifikasinya kesalahan-kesalahan yang menjadi dasar untuk melakukan perbaikan dan perubahan.
6. **Perbaikan dan perubahan sistem**, pada tahapan ini dilakukan perbaikan dan atau perubahan, baik pada program prototype juga pada rancangan-rancangan sistem yang telah dibuat sebelumnya. Keluaran dari tahapan ini adalah program prototype yang siap diinstall pada lingkungan pengguna.

Hasil dan Pembahasan

1. Identifikasi Permasalahan

Tabel 1. Permasalahan yang teridentifikasi

No.	Permasalahan	Pemecahan terhadap masalah
1	Kurangnya penguasaan kemampuan pemrograman oleh mahasiswa yang disebabkan oleh kurangnya latihan mandiri	Pengembangan aplikasi yang dapat memberikan latihan pemrograman pada mahasiswa
2	Beratnya beban menilai tugas pemrograman mahasiswa	Aplikasi dapat menilai secara otomatis nilai hasil kerja mahasiswa
3	Tingginya tingkat plagiat terhadap tugas pemrograman	Aplikasi dapat mencegah dilakukannya plagiat terhadap latihan yang diberikan

4	Mahasiswa kurang termotivasi melakukan latihan mandiri	Aplikasi menggunakan daya tarik games untuk mengemas materi latihan yang diberikan
---	--	--

2. Analisa Kebutuhan Sistem

Berdasarkan hasil identifikasi permasalahan maka dilakukan analisis kebutuhan sistem yang akan dibangun yaitu sistem yang memiliki fitur dan karakteristik:

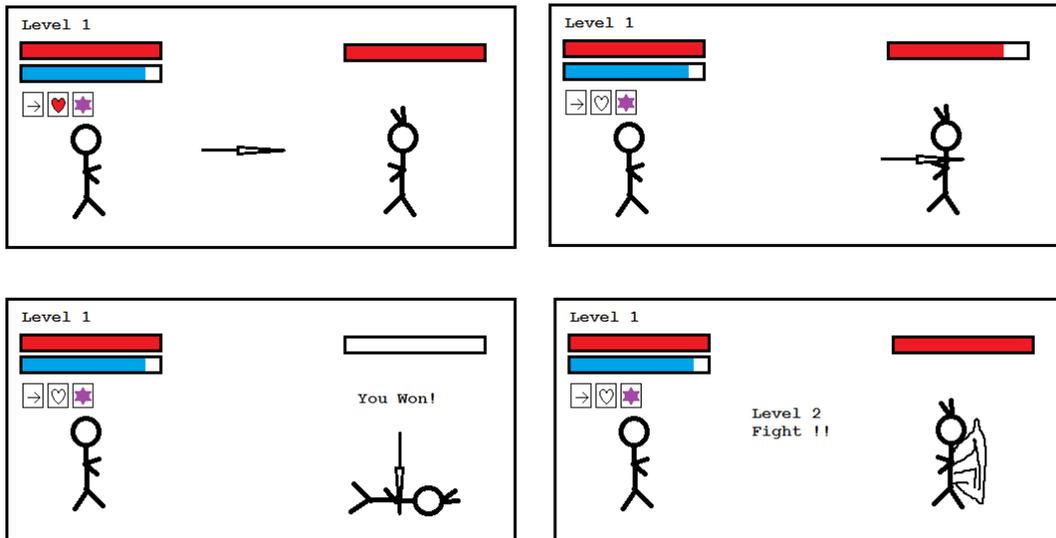
- Aplikasi dapat memberikan latihan pemrograman pada mahasiswa, hal ini agar tidak membatasi jumlah latihan yang dapat dilakukan oleh mahasiswa, sehingga dapat mempercepat tingkat penguasaan kemampuan pemrograman tanpa memerlukan effort yang lebih dari pengajar.
- Aplikasi dapat menilai hasil kerja mahasiswa, hal ini agar mahasiswa dapat secara langsung mendapatkan umpan balik dari hasil kerjanya yang dapat meningkatkan motivasinya untuk belajar dari kesalahan yang dilakukan dan untuk memotivasi dalam melanjutkan latihan mandiri.
- Aplikasi menggunakan elemen-elemen games untuk meningkatkan motivasi dan ketertarikan mahasiswa untuk mencoba aplikasi ini, dan untuk mengubah persepsi mahasiswa tentang pembelajaran pemrograman.

3. Perancangan Sistem

3.1. Disain Model Permainan

Model permainan digunakan menggunakan storyboard yang menggambarkan pengimplementasian elemen-elemen games yang akan diterapkan pada sistem. Elemen-elemen games yang dimasukkan dalam sistem :

- Pemain memiliki pilihan serangan/pertahanan yang akan digunakan, setiap aksi memerlukan Magic Point.
- Menggunakan Hit Point sebagai penanda kehidupan karakter pemain dan musuh.
- Menggunakan Magic Point yang menunjukkan jumlah energi magic yang digunakan untuk menyerang.
- Menggunakan tingkatan kemampuan musuh yang akan terus meningkat saat pemain berhasil memenagkan suatu tingkat musuh.



Gambar 1. Story Board

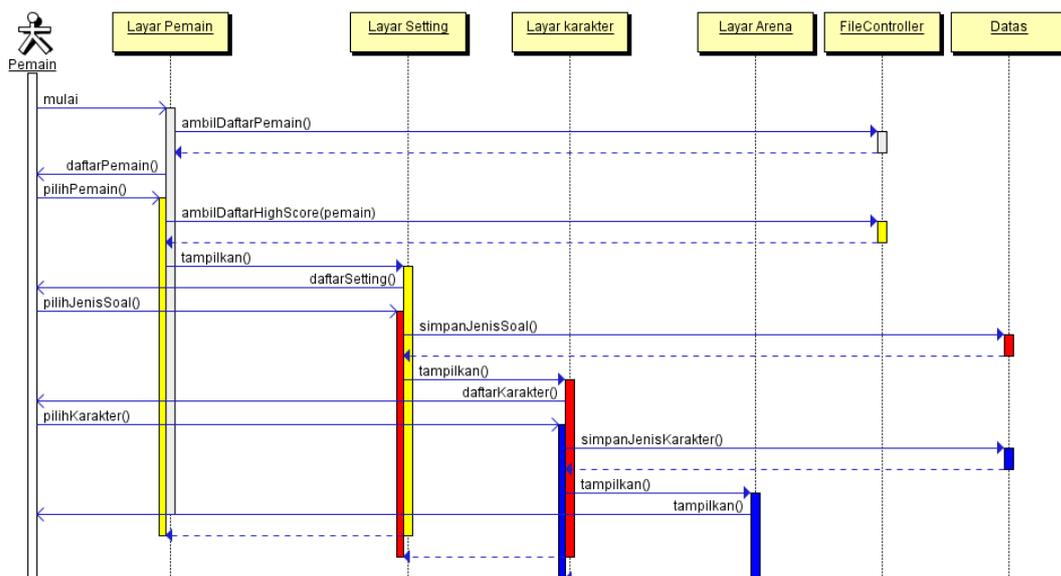
3.2. Disain Berkas Penyimpanan

Berkas Penyimpanan yang digunakan berupa file teks yang didekripsi, hal ini mengingat berkas yang akan disimpan hanya berupa data nama pemain dan nilai tertinggi yang dicapainya. Adapun format berkas file sebelum dienkripsi adalah sebagai berikut:

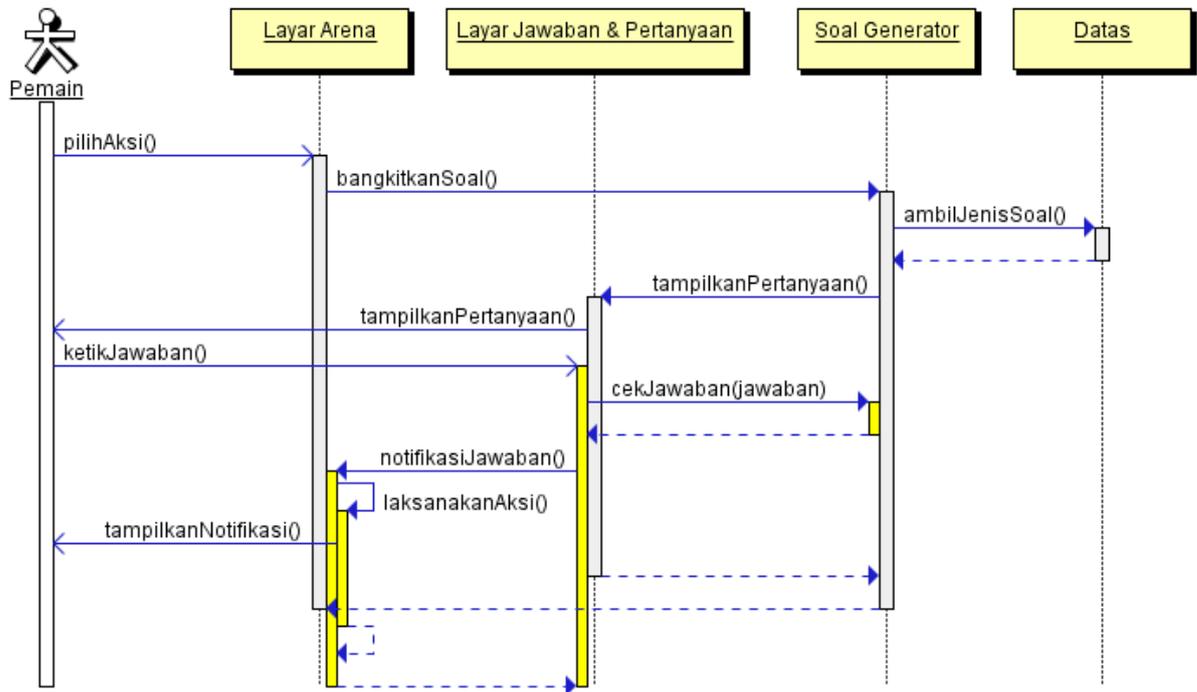
Nama Pemain#Kode Jenis Soal#Nilai tertinggi
 Nama Pemain2#Kode Jenis Soal#Nilai tertinggi
 Nama Pemain3#Kode Jenis Soal#Nilai tertinggi

Gambar 2. Format File Teks

3.3. Interaksi Antar Objek



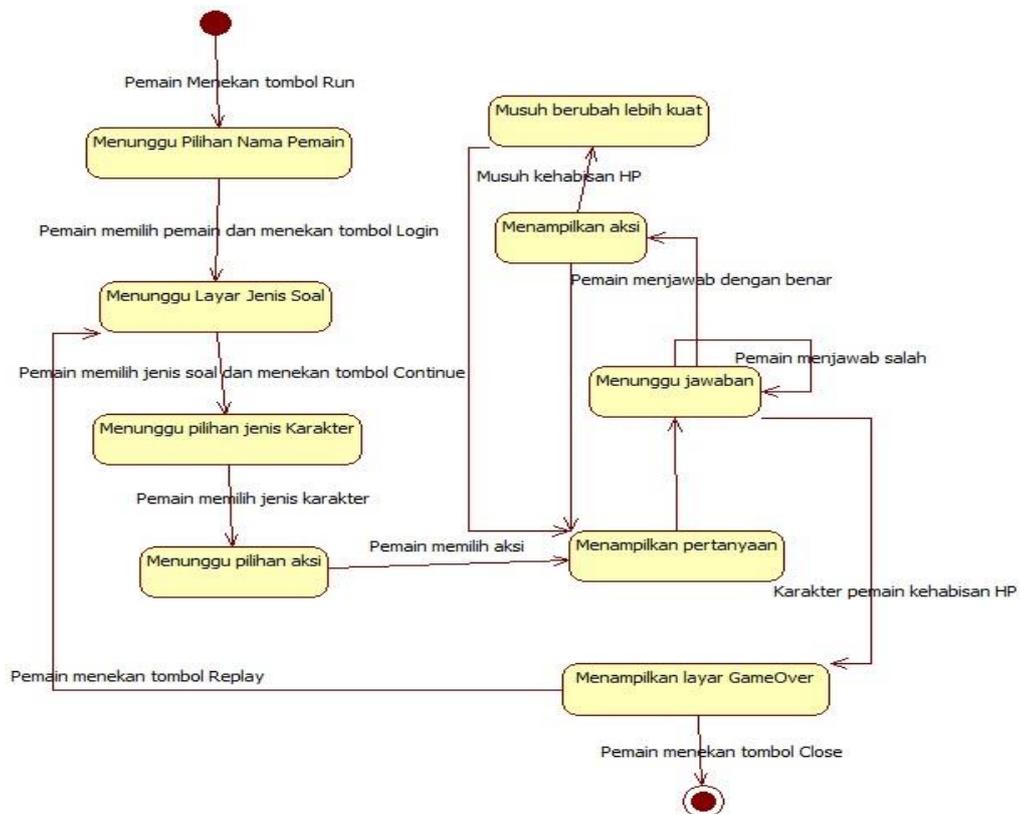
Gambar 3. Sequence Diagram Pilihan Pemain sebelum bermain



Gambar 4. Sequence Diagram Aksi Pemain saat bermain

3.4. Disain Transisi State

Disain transisi antar state dalam aplikasi :



Gambar 5. State Transition Diagram

KESIMPULAN

- a). Sistem yang dibangun dapat menilai jawaban yang diberikan oleh mahasiswa sehingga dapat mengurangi beban kerja pengajar (Uji U-1-05 dan U-1-06).
- b). Sistem menggunakan elemen games untuk meningkatkan motivasi belajar mahasiswa dengan menggunakan umpan balik yang cepat, dan menggunakan aksi yang melakukan animasi (U-1-04), penggunaan highscore (nilai tertinggi) (Uji U-1-07), dan tingkatan musuh (Uji U-1-09).
- c). Sistem dapat menekan terjadinya plagiat dengan meningkatkan variabilitas soal yang dibangkitkan (U-1-08).

DAFTAR PUSTAKA

García, Gines-Mateos, dkk, 2009, *A Course on Algorithms and Data Structures Using On-line Judging*, ACM.

Stiller, Evelyn, 2009, *Teaching Programming Using Bricolage*, Consortium for Computing Sciences in Colleges.

McFarland, Ronald D, 2003, *Teaching Students to Learn in The Computer Science and Information Systems Curriculum: Creating A Distinction Between Content and Methods*, Consortium for Computing Sciences in Colleges.

Sheard, Judy, dkk, 2002, *Cheating and Plagiarism: Perceptions and Practices of First Year IT Students*, ACM.

Lesner, Boris, dkk, Boris, dkk, 2010, *A Novel Framework to Detect Source Code Plagiarism: Now, Students Have to Work for Real!*, ACM.

Flood, Raymond, dkk, 2005, *Teaching Programming Collaboratively*, ACM.

Warren, Peter, 2001, *Teaching Programming Using Scripting Languages*, Consortium for Computing in Small Colleges.

Arshad, Naveed, 2009, *Teaching Programming and Problem Solving to CS2 Students using Think-Alouds*, ACM.

Pears, Arnold, dkk, *A Survey of Literature on the Teaching of Introductory Programming*, ACM.

Palumbo, D, 1990, *Programming language/problem-solving research: a review of relevant issues. Review of Educational Research*.