

PROPOSAL SKRIPSI

PENERAPAN *STATE BASED CODE EDITOR* PADA SISTEM *E – LEARNING* BERBASIS WEB



Oleh :

Nama : Satria Efriyadi

NPM : G1A017069

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS BENGKULU

2022

1. Judul Penelitian

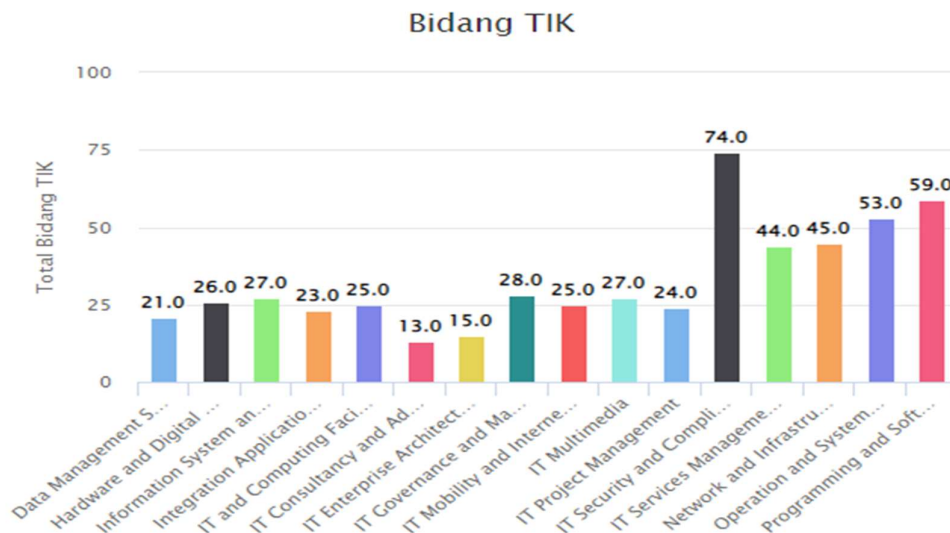
Penerapan *State based code editor* pada sistem *e – learning* berbasis web.

2. Bidang Ilmu

Bidang ilmu yang penulis akan teliti adalah sistem *e – learning* dan *learning design*.

3. Latar Belakang

Peta Okupasi Nasional dalam Kerangka Kualifikasi Nasional Indonesia (KKNI) merupakan dokumen yang disusun untuk memetakan jenis - jenis jabatan/okupasi/profesi yang ada di berbagai bidang, sub-bidang maupun area fungsi di semua jenis pekerjaan. Pengembangannya menggunakan pendekatan area fungsi dari proses kerja atau okupasi/jabatan/profesi suatu kegiatan usaha/industri/pekerjaan sejenis. Kemudian tampilannya berbentuk katalog yang memberikan deskripsi keahlian dan unit kompetensi di setiap okupasi/jabatan/profesi.



Gambar 1 grafik peta okupasi nasional Indonesia bidang TIK dikutip dari publikasi

Kementrian PPN/Bappenas dan dapat diakses pada :

<https://petaokupasi.bappenas.go.id/dashboard2/>

Pada gambar 1 digambarkan grafik peta okupasi nasional Indonesia bidang teknologi informasi dan komunikasi (TIK). Berdasarkan grafik tersebut dari seluruh lini kategori, Hampir semua lini kategori menunjukkan bahwa Indonesia masih kekurangan sumber daya manusia (SDM) di bidang teknologi informasi dan komunikasi. Dimana okupasi di bidang *IT Security and Compliance* menempati posisi pertama dalam hal okupasi yang paling banyak ditempati dengan poin 74.0 dari 100 dan okupasi paling sedikit adalah *IT Consultancy and Advisory* dengan poin 13.0 dari 100. Berdasarkan data tersebut dapat disimpulkan bahwa kondisi SDM bidang TIK dalam negeri cukup memprihatinkan dan salah satu penyebab utamanya adalah *skill-mismatch*.

Skill-mismatch adalah ketidaksesuaian antara keterampilan yang dicari oleh pemberi kerja dengan keterampilan yang dimiliki oleh individu. Sederhananya, ini adalah ketidaksesuaian antara keterampilan dengan pekerjaan. Ini berarti bahwa pendidikan dan pelatihan tidak menyediakan keterampilan yang diminta di pasar tenaga kerja. Untuk mengatasi permasalahan ini, dibutuhkan sebuah sistem daring yang dapat memberikan pendidikan sesuai kompetensi yang dibutuhkan di pasar tenaga kerja.

Jaya Kumar C. Koran (2002), mendefinisikan e-Learning sebagai sembarang pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau internet) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan, sedangkan Rosenberg (2001) menekankan bahwa e-Learning merujuk pada penggunaan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan. Hal ini senada dengan Cambell (2002), Kamarga (2002) yang intinya menekankan penggunaan internet dalam pendidikan sebagai hakikat *e-learning*.

Permasalahan yang umum terjadi pada pembelajaran pemrograman melalui e-Learning adalah konfigurasi lingkungan pengembangan yang

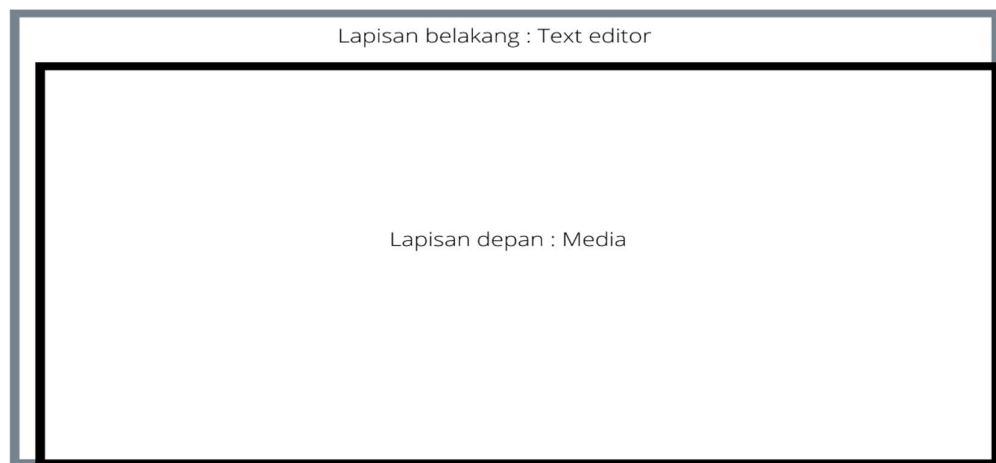
rumit. Hal ini disebabkan oleh spesifikasi komputer yang digunakan pendidik dan murid beragam sehingga menyebabkan cara konfigurasi lingkungan pengembangan yang berbeda. Selain itu cara penyampaian topik pembelajaran melalui media audio visual cenderung membuat murid hanya memperhatikan topik pembelajaran tanpa mempraktikkan hasil pembelajaran yang telah dipelajari sehingga pencapaian belajar murid menjadi kurang maksimal.

Salah satu usaha yang dapat dilakukan dalam mengatasi permasalahan ini adalah dengan mendesain *code editor* sedemikian rupa agar dapat meningkatkan pemahaman dalam proses pembelajaran. Contohnya seperti pada karya tulis ilmiah berjudul “*An Inquisitive Code Editor for Addressing Novice Programmer’s Misconceptions of Programs Behavior*”. Pada penelitian tersebut dibuat sebuah pengaya tambahan pada Atom, sebuah perangkat lunak *code editor*.

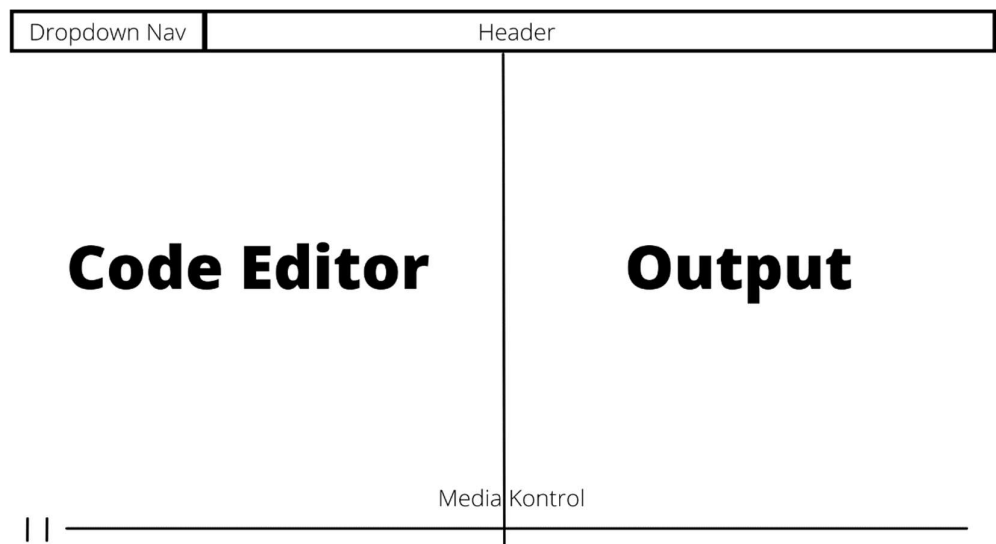
Pengaya tersebut dapat mengatasi permasalahan berupa kesalahpahaman pemrogram pemula terhadap bagaimana *source code* yang mereka buat bekerja dengan cara mendeteksi potongan kode yang berpotensi bermasalah kemudian mempertanyakan potongan kode tersebut kepada pengguna. Topik penelitian ini menarik dikarenakan pemrogram pemula cenderung beranggapan bahwa *source code* yang sudah mereka buat sudah benar jika tidak muncul eror ketika proses kompilasi, berbeda dengan pandangan pemrogram profesional yang menilai dari beberapa faktor seperti efisiensi *source code*, *readability* dan lain sebagainya.

Berdasarkan referensi diatas tercetuslah ide membuat sebuah *code editor* yang memiliki *state* atau keadaan didalamnya. *State based code editor* adalah *code editor* yang dapat menampilkan *source code* sesuai dengan *state* yang telah didefinisikan sebelumnya. Cara kerja *state based code editor* terinspirasi dari diagram transisi pada *Finite State Automata* (FSA) yang merupakan model matematika abstrak. *State based code editor*

bekerja dengan cara menerima masukan berupa *source code* yang disimpan di dalam sistem atau bisa disebut sebagai *state*. *Source code* tersebut kemudian dapat dipanggil dan dimunculkan kembali kedalam *code editor* pada sistem e-Learning berdasarkan waktu *code editor* mengalami perubahan pada video pembelajaran. Sehingga dengan adanya *state* di dalam *code editor* memungkinkan *code editor* untuk berubah secara dinamis mengikuti perubahan pada *code editor* pendidik.



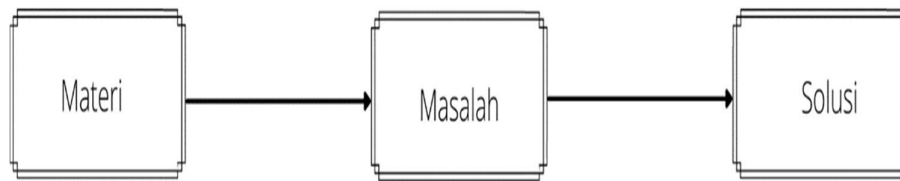
Gambar 2 Konsep antarmuka berlapis sistem e-Learning



Gambar 3 Konsep tampilan antarmuka pengguna sistem e-Learning

Pada gambar 2 dan 3 ditunjukkan konsep antarmuka yang akan digunakan pada sistem e-Learning. Desain tersebut bertujuan untuk memaksimalkan penerapan *state based code editor*. desain antarmuka pengguna sistem dirancang agar dapat bersinergi dengan *state based code editor*. Pada gambar 2 adalah desain antarmuka berlapis yang terdiri dari dua lapisan, yaitu lapisan depan dan lapisan belakang. Lapisan depan merupakan wadah untuk menampilkan materi pelajaran melalui media video pembelajaran dan kontrol media. Lapisan kedua merupakan wadah untuk pengguna melakukan praktik dari pembelajaran yang telah dipelajari, tampilan lapisan kedua sama dengan lapisan pertama hanya saja tampilan video digantikan oleh *state based code editor* untuk mempraktikkan materi pembelajaran. Pada gambar 3 adalah desain tampilan pengguna sistem dimana tampilan dibagi menjadi dua bagian yaitu bagian kiri tempat *code editor* dan bagian kanan yang merupakan tempat menampilkan *output*.

Berdasarkan konsep antarmuka diatas, metode pembelajaran yang diterapkan juga perlu didesain sedemikian rupa agar dapat mengeluarkan potensi penerapan *state based code editor* sebaik mungkin. Metode pembelajaran yang akan digunakan adalah kombinasi dari *problem based learning* yaitu berdasarkan Boud (2010:285) adalah pendekatan pembelajaran yang mengarah pada pelibatan murid dalam mengatasi masalah belajar dengan praktik nyata yang dekat dengan kehidupan sehari – hari dan *bite sized learning* yaitu pendekatan pembelajaran dimana materi pembelajaran dipecah menjadi bagian – bagian kecil agar dapat mudah dipahami oleh murid, dimana penyampaian materi pembelajaran disampaikan dengan cara informasi materi disajikan dalam format *bite sized learning* dan *problem based learning* akan diterapkan pada pertengahan sesi pembelajaran untuk meningkatkan pemahaman materi yang sedang dipelajari sehingga alur pembelajaran dapat digambarkan seperti pada gambar 4 berikut ini :



Gambar 4 ilustrasi alur pembelajaran

Berdasarkan paparan diatas, peneliti akan melakukan penelitian dengan topik “*Implementasi State Based Code Editor pada Sistem e-Learning berbasis Web*”. Penelitian ini berfokus pada cara membuat dan mengintegrasikan *State Based Code Editor* ke dalam sebuah *sistem e-Learning berbasis website*. Penelitian ini di harapkan dapat menjadi salah satu solusi untuk meningkatkan pemahaman murid dalam mempelajari bahasa pemrograman melalui media daring dan dapat membuat murid fokus ke materi pembelajaran tanpa perlu memikirkan permasalahan yang mungkin terjadi dalam mengkonfigurasi lingkungan pengembangan komputer untuk mengikuti materi yang sedang dipelajari.

4. Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat dirumuskan beberapa permasalahan sebagai berikut:

- 1) Bagaimana membuat dan merancang sistem e-Learning yang dapat memaksimalkan pemahaman murid dalam mempelajari bahasa pemrograman
- 2) Bagaimana pengaruh State Based Code Editor dalam e-Learning.
- 3) Bagaimana merancang dan mengimplementasikan sebuah State Based Code Editor yang dapat terintegrasi dengan Sistem e-Learning.

5. Batasan Masalah

Agar aplikasi ini tidak terlalu luas dalam proses pengembangannya, maka peneliti membuat batasan-batasan permasalahan sebagai berikut :

- 1) Sistem *e-Learning* terintegrasi dengan *State Based Code Editor* berbasis *Web*.
- 2) UI/UX Sistem *e-Learning* di aplikasikan menggunakan HTML,CSS dan JS beserta *Library* yang bersangkutan.
- 3) Sistem ini diuji cobakan pada user yang mendaftar pada sistem *e-Learning* berbasis web yang akan di buat.

6. Tujuan Penelitian

Tujuan penelitian yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

- 1) Menjadi salah satu solusi atas permasalahan kurangnya tenaga kerja di bidang teknologi dan informasi
- 2) Merancang dan membangun sistem *e-Learning* yang dapat memberikan materi yang mudah dipahami oleh pengguna.
- 3) Merancang dan mengimplementasikan *State Based Code Editor* kedalam sistem *e-Learning* berbasis web yang akan dibuat.
- 4) Melakukan analisa sistem yang telah dibuat berdasarkan metode *pieces (Performance, information, economy, control, efficiency and services)*.

7. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

- 1) Memudahkan dalam mempraktikkan materi yang sedang dipelajari.
- 2) Menghindari dari konfigurasi *Code editor* kompleks yang tidak ramah bagi pemula.
- 3) Menghasilkan sistem yang dapat membantu pengguna mempelajari bahasa pemrograman dan mempraktikkan ilmu yang telah dipelajari.
- 4) Menghasilkan sebuah sistem pembelajaran daring yang dapat menarik pengguna untuk ikut terlibat dalam proses pembelajaran.

8. Tinjauan Pustaka

8.1. *E-Learning*

Jaya Kumar C. Koran (2002), mendefinisikan e-Learning sebagai sembarang pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau internet) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan, sedangkan menurut Rosenberg (2001) e-Learning merujuk pada penggunaan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan.

Berdasarkan cara penyampaian e-Learning dapat dikelompokkan menjadi dua, yaitu komunikasi satu arah dan komunikasi dua arah, e-Learning yang akan penulis teliti adalah e-Learning berbasis web dengan komunikasi satu arah dengan materi yang diajarkan seputar pemrograman bagi pemula dan dibantu oleh *State Based Code Editor* sebagai salah satu media penyampaian materi.

Manfaat e-Learning adalah sebagai berikut :

1) Fleksibilitas

Pembelajaran konvensional mengharuskan siswa untuk hadir di kelas pada jam tertentu dan materi yang telah diajarkan tidak bisa diulang kembali karena penyampaian materi secara tatap muka, sedangkan dengan e-Learning materi disimpan dalam bentuk media dan dapat diakses kembali kapan saja selama media yang digunakan untuk mengakses e-Learning memiliki jaringan internet.

2) Independent Learning

Dengan memanfaatkan sistem e-Learning memungkinkan siswa untuk memiliki kendali atas waktu memulai sesi belajar, menghentikan sesi belajar, bahkan memilih topik yang akan dipelajari terlebih dahulu, sehingga bisa dikatakan sebagian besar kesuksesan belajar dipegang oleh siswa.

3) Biaya

Banyak biaya yang dapat dihemat dengan e-Learning. Contohnya seperti biaya transportasi dan akomodasi, biaya administrasi, biaya pengadaan sarana dan fasilitas fisik (penyewaan atau penyediaan kelas, kursi, papan tulis) dan lain sebagainya.

8.2. *Passive Learning*

Passive Learning adalah metode belajar secara pasif dimana murid belajar dari mendengarkan dan mengobservasi, hal ini memberikan dampak yang buruk karena bagaimanapun juga, menulis kode adalah sebuah kemampuan, dan setiap kemampuan perlu di praktikkan.

8.3. *Active Learning*

Active Learning adalah pendekatan kegiatan belajar apapun dimana setiap murid berpartisipasi atau berinteraksi dengan proses pembelajaran. Metode ini dapat meningkatkan keterampilan berpikir tingkat tinggi, seperti penerapan terhadap ilmu yang telah dipelajari, analisis dan sintesis Metode ini berlawanan dengan *Passive learning*.

8.4. *Hyperteks Markup Language - Document Object Model (HTML - DOM)*

DOM atau *Document Object Model* adalah model standar penulisan sebuah dokumen XML atau HTML. DOM yang digunakan sebagai standar penulisan dokumen HTML disebut juga dengan HTML-DOM. HTML-DOM berfungsi untuk mengatur elemen-elemen html disusun memenuhi dan bagaimana untuk mendapatkan, mengubah, menambah, atau menghapus elemen html.

Elemen html adalah kode html yang dimulai dari sebuah *tag* pembuka dan diakhiri dengan *tag* penutup, *tag* itu sendiri adalah kode-kode tertentu yang menjadi pengenalan bahwa kode tersebut adalah kode html dan bisa diterjemahkan oleh browser. Ada tiga *tag* utama yang membangun sebuah halaman html, yaitu *tag* html (<html></html>), *tag* head (<head></head>) dan *tag* body (<body> </body>) (Mitra et al., 2017).

8.5. Learning Design

Learning Design adalah kerangka kerja yang mendukung pengalaman belajar, merujuk kepada pilihan yang disengaja tentang apa, kapan, di mana dan bagaimana suatu cara mengajar di terapkan, keputusan yang di perlukan tentang suatu konten ajar, struktur, waktu, strategipedagogis, urutankegiatan pembelajaran serta sifat teknologi yang digunakan untuk mendukung suatu pembelajaran.

8.6. State Based Code Editor

State Based Code Editor adalah Code Editor yang telah didefinisikan *state* atau keadaan di dalamnya, dengan adanya *state* di dalam *code editor* memungkinkan *code editor* untuk berubah secara dinamis mengikuti keadaan *code editor* pada materi suatu *course* bahasa pemrograman.

State Based Code Editor terinspirasi dari diagram transisi pada *Finite State Automata* (FSA), FSA adalah mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit yang dapat mengenali bahasa paling sederhana (bahasa reguler) dan dapat diimplementasikan secara nyata dimana sistem dapat berada pada salah satu dari sejumlah berhingga konfigurasi internal yang disebut sebagai *state*.

Secara sederhana dapat disimpulkan bahwa FSA adalah mesin paling sederhana untuk mengenali suatu pola. Beberapa contoh sistem yang memanfaatkan FSA antara lain pada mesin minuman otomatis, pengatur lampu lalu lintas, analisis leksikal, teks editor dan protokol komunikasi jaringan.

Suatu FSA mempunyai sekumpulan *state* dan aturan – aturan untuk berpindah dari suatu *state* ke *state* yang lain. *State* tersebut dapat terdiri dari, *Initial state* (keadaan awal), *Finite set of state* (himpunan keadaan berhingga) dan *Set of Final States* (himpunan dari keadaan akhir).

FSA dinyatakan dalam lima elemen atau dalam bahasa matematis disebut sebagai 5-tupel dan bila dinyatakan dalam spesifikasi formal mesin menjadi $\{ Q, \Sigma, q, F, \delta \}$, 5-tupel tersebut adalah :

Q : Himpunan state / kedudukan.

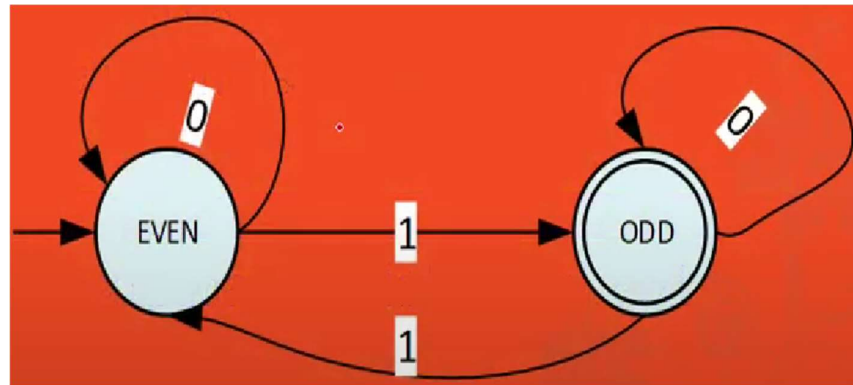
Σ : Himpunan simbol input/masukan/abjad.

q : state awal q_0 , dimana $q_0 \in Q$.

F : Himpunan state akhir, dimana $F \subseteq Q$.

δ : Fungsi transisi.

Bila digambarkan kedalam diagram transisi akan menjadi :



Gambar 5 Diagram transisi untuk pengecek pariti ganjil

Keterangan :

- *Initial state* ditandai dengan busur tanpa asal *state*
- Lingkaran menyatakan *state*
- Label pada lingkaran menyatakan *state*
- Busur menyatakan arah transisi/ arah perpindahan *state*
- Label pada busur adalah simbol masukan
- Lingkaran ganda menyatakan *Final state*

Pada gambar 5 pengecek pariti ganjil sistem akan menerima bila jumlah bit 1 adalah ganjil, ketika *state even* menerima masukan berupa bilangan 0 maka masukan tersebut akan berpindah ke arah *state even* itu sendiri dan kemudian jika *state even* menerima masukan 1 maka masukan

1 akan berpindah ke *state odd* sedangkan jika *state odd* menerima masukan 1 maka masukan akan berpindah ke *state odd* itu sendiri dan bila masukan adalah 0 maka masukan akan berpindah ke arah *state even*.

Maka dapat dituliskan menjadi :

$$Q = \{ODD, EVEN\}$$

$$\Sigma = \{0, 1\}$$

$$q_0 = EVEN$$

$$F = \{ODD\}$$

Catatan :

Jumlah F bisa lebih dari satu.

Fungsi transisi/ δ dapat ditulis menjadi :

- $\delta(EVEN, 0) = EVEN$
- $\delta(EVEN, 1) = ODD$
- $\delta(ODD, 0) = ODD$
- $\delta(ODD, 1) = EVEN$

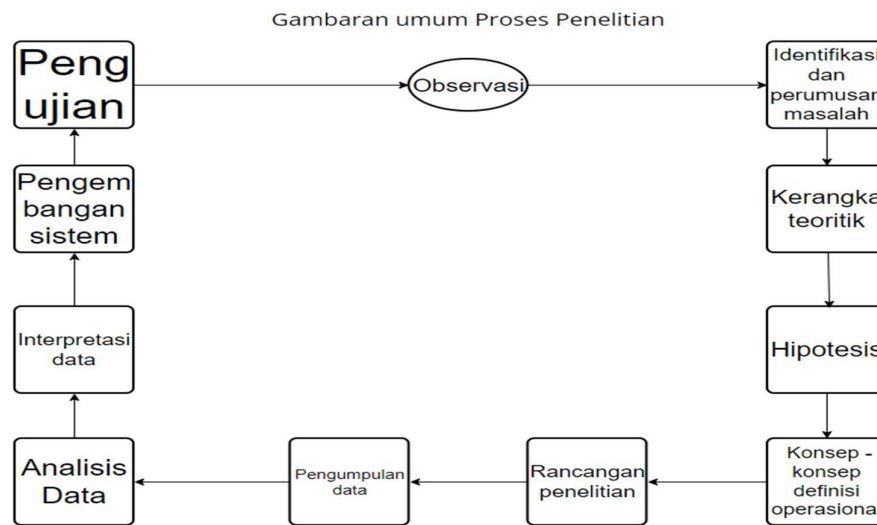
Sederhananya masukan akan berpindah kedalam *state* yang telah didefinisikan dan bila masukan berakhir pada *final state* maka masukan diterima. Dari cara kerja tersebut dapat dibuat *code editor* yang memiliki cara kerja mirip dengan Diagram transisi pada FSA, sehingga dihasilkan sebuah *code editor* yang dapat menampilkan baris kode sesuai dengan keadaan pada video pembelajaran pada sistem e-Learning.

8.7. Node.js

Node.js adalah Javascript Runtime Environment yang di bangun di atas *Chrome V8 Javascript Engine*, *Node.js* memungkinkan pengembang menggunakan bahasa pemrograman javascript untuk menulis perintah dan *server-side scripting*.

9. Metode Penelitian

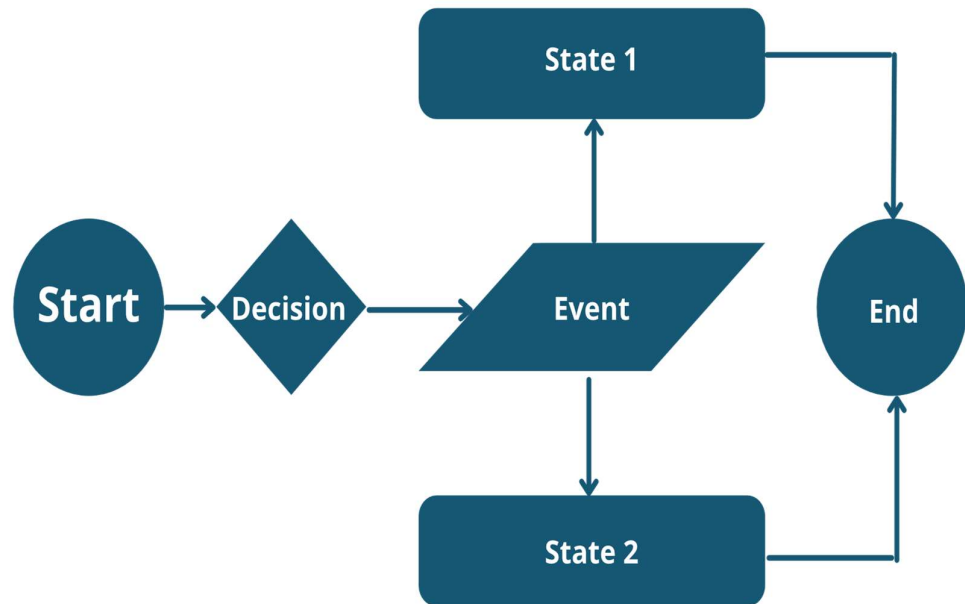
Menurut Sekaran (2000:20), “*Scientific research focus on the goal of problem solving and pursues a step-by-step logical, organized, and regiorious method to identify problems, gather the data, analyze them, and draw valid conclusions thereform.*” Maka langkah proses pengembangan penelitian yang penulis ambil dapat digambarkan sebagai berikut :



Gambar 6 ilustrasi proses pengembangan penelitian

Pada gambar 6 dijelaskan proses pengembangan penelitian yang dimulai dari observasi untuk mengidentifikasi dan merumuskan masalah. Kemudian barulah dibuat kerangka teoritik sebagai bentuk kesimpulan mentah dari masalah dan selanjutnya dapat diambil hipotesis dari kesimpulan tersebut. Setelah hipotesis didapatkan barulah rancangan definisi operasional dibuat untuk mendapatkan definisi dari variabel yang akan diteliti agar dapat diukur dan diamati. Langkah selanjutnya adalah proses umum dalam penelitian seperti merancang penelitian, mengumpulkan data kemudian data diolah melalui proses analisis dan interpretasi kemudian variabel yang diteliti dikembangkan dalam hal ini adalah sistem dan terakhir dilakukan pengujian dan observasi terhadap variabel yang diteliti.

Flow Chart state based code editor



Gambar 7 Diagram Flowchart *State Based Code Editor*

Pada gambar 7 Diagram Flowchart *State Based Code Editor*, alur kerja *State Based Code Editor* dalam sistem e-Learning berawal dari notasi *Start* yang melambangkan bahwa materi dalam e-Learning mulai diakses. Kemudian notasi *Decision* mewakili keputusan sistem untuk menampilkan salah satu *State* berdasarkan aturan yang telah didefinisikan. Notasi *Event* mewakili proses transisi sistem dari *Default State* ke *State* tujuan. Dan notasi *End* mewakili akhir dari alur kerja yang menandakan bahwa sesi pembelajaran telah diakhiri.

Sedangkan metodologi penelitian yang digunakan peneliti dalam tugas akhir ini terdiri dari:

a. Metode Pengumpulan Data

1) Studi Pustaka

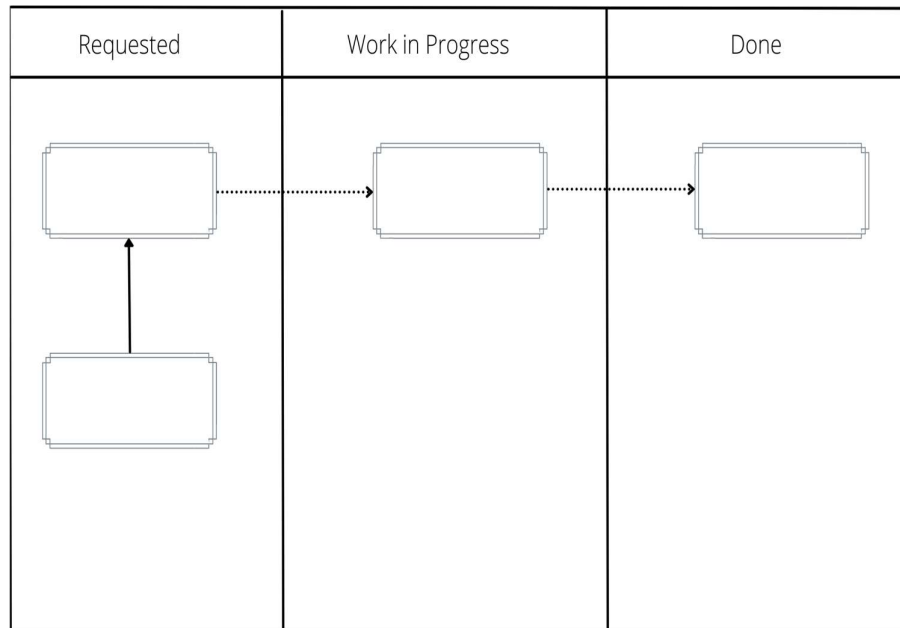
Metode ini dilakukan dengan cara mengumpulkan data yang diperoleh dari berbagai literatur, seperti buku, jurnal dan internet yang membahas topik yang relevan dengan penelitian, sehingga dapat membantu proses pengerjaan tugas akhir.

2) Kuesioner

Kuesioner diberikan kepada mahasiswa yang mengerjakan tugas akhir di Fakultas Teknik Universitas Bengkulu. Hasil dari kuesioner diharapkan dapat digunakan sebagai data yang menunjang penelitian dan hal-hal yang diperlukan untuk membangun sistem ini

b. Metode Pengembangan Sistem

Ilya Bibik (2018:9) menjelaskan bahwa *kanban* adalah segala hal tentang memvisualisasikan sebuah proses, jadi ungkapan “Sebuah gambar bernilai seribu kata” sangat mendefinisikan proses *kanban* secara akurat. Metode *Kanban* ditemukan oleh Toyota pada tahun 1940-an untuk mengurangi waktu menganggur selama manufaktur. Dalam pengembangan perangkat lunak, itu berarti melaksanakan pekerjaan yang datang dan menggunakan *signboard* untuk melacak kemajuan dan hambatan. Pekerjaan tidak harus dilakukan dalam bentuk iterasi ketika menggunakan metode *kanban*, karena metode *kanban* berfokus kepada proses kerja. Kelebihan utama dalam menggunakan metode *kanban* adalah visualisasi dalam proses kerja yang dapat membantu untuk mengidentifikasi hambatan dan kesempatan untuk mengurangi waktu menganggur. Metode *kanban* membatasi dengan ketat jumlah pekerjaan yang dapat dikerjakan dalam satu waktu, hal ini dilakukan agar dapat meminimalisir kemungkinan terjadinya hambatan dalam mengerjakan proyek.



Gambar 8 Ilustrasi metode pengembangan *Kanban*

Pada gambar 8 ilustrasi metode pengembangan *kanban*, terdapat *signboard* yang berisikan tiga kolom yaitu *Requested*, *Work in Progress (WIP)*, dan *Done*, kolom *Requested* adalah permintaan atau bisa juga pekerjaan yang akan di lakukan. Kemudian *Requested* akan masuk ke kolom WIP, kolom WIP adalah tempat mendefinisikan proses kerja dan di dalam kolom WIP terdapat limitasi jumlah WIP yang dapat di proses untuk menghindari kemungkinan terjadinya hambatan, kemudian setiap pekerjaan yang sudah selesai akan masuk kedalam kolom *done*, Ketika di konstruksi dan dikelola dengan benar, *kanban* dapat berfungsi sebagai *Real-time Information Repository* sehingga dapat menyoroti hambatan dalam sistem dan apapun yang mungkin dapat mengganggu jalannya pekerjaan.

c. Pengujian Sistem

Pengujian sistem dilakukan untuk melihat apakah program sudah berjalan dengan sebagaimana mestinya atau belum, Pengujian sistem dilakukan dengan menggunakan dua metode yaitu metode *White box* dan *Black Box*.

d. Alat dan Bahan

Dalam melaksanakan penelitian ini, dibutuhkan beberapa sarana pendukung yang merupakan perangkat-perangkat yang di butuhkan untuk penerapan *state based code editor* pada sistem e-Learning berbasis website. Untuk melaksanakan penelitian ini, peneliti akan menggunakan alat dan bahan sebagai berikut:

Alat:

- Komputer Desktop
- Processor AMD Ryzen™ 5 2400G
- RAM 8 GB
- RX 6600 8 GB

Bahan:

- Browser
- Sistem Operasi
Windows 10
- Visual Studio Code

10. Penelitian Terkait

No.	Peneliti	Judul	Tahun	Metode/ Objek	Hasil	Perbedaan
1.	Eka Wahyu Hidayat, Andi Nur Rachman, Muhammad Fauzan Azim	Penerapan <i>Finite State Machine</i> pada <i>Battle Game</i> Berbasis <i>Augmented Reality</i>	2019	Metode: <i>Finite State Automata</i> (FSA)	Aplikasi permainan berbasis android menggunakan teknologi <i>Augmented Reality</i> (AR) dimana metode FSA dipakai sebagai kecerdasan buatan pada karakter permainan.	Objek penelitian adalah aplikasi permainan berbasis android menggunakan teknologi <i>Augmented Reality</i> dan menggunakan <i>Finite State Automata</i> sebagai kecerdasan buatan pada NPC (<i>Non Playable Characters</i>). Penelitian yang akan dilakukan mengintegrasikan <i>State based code editor</i> pada sistem e – learning berbasis website.

2.	Moatsu m Alawida , Je Sen Teh, Azman Samsud in, Wafa, Hamdan Alshour a	An Image <i>Encryption Scheme based on Hybridizing Digital Chaos and Finite State Machine</i>	2018	Metode : <i>Digital Chaos dan Finite State Automata</i>	Diperoleh metode <i>Chaotification</i> baru untuk meningkatkan peta digital <i>tent maps</i> , metode ini adalah hasil kombinasi dari <i>tent map</i> dan <i>deterministic finite state machine</i>	Bidang penelitian ini adalah kriptografi dengan tema skema enkripsi gambar berdasarkan gabungan Teori <i>Chaos/Chaos system</i> dan <i>Finite State Machine</i> . Penelitian penulis fokus kepada desain pembelajaran pada sistem e-Learning dengan cara mengintegrasikan <i>state based code editor</i> kedalam sistem e-Learning berbasis website.
3.	M. Syahrul Rizal, Mufariz uddin,	<i>Modeling the Semantics of States and State Machines</i>	2020	Objek : <i>State dan State Machine</i>	Penelitian bertujuan untuk menetapkan definisi yang tepat dari notasi <i>State</i> dan <i>State Machines</i> . Didapatkan hasil analisa berdasarkan metodologi <i>Thinging Machine</i> (TM) yaitu berupa kesimpulan bahwa <i>State</i> adalah notasi utama dari	Penelitian bertujuan untuk menetapkan definisi yang tepat dari notasi <i>State</i> dan <i>State Machines</i> , Penelitian penulis bertujuan untuk merancang dan mengintegrasikan <i>State</i>

	Yola Aprilia Koto				<i>State Machines</i> , yang mana <i>Events</i> mendorong perubahan <i>state</i> .	<i>Based Code Editor</i> yang cara kerjanya terinspirasi dari <i>State Machines</i> (FSA) ke dalam sistem e-Learning berbasis website.
4.	Patrícia Salajová	<i>Web Editor of Finite Automata</i>	2021	Objek : Web Editor	Hasil dari penelitian ini adalah editor berbasis web untuk membantu dalam pengajaran materi tentang <i>Finite Automata</i> .	<p>Hasil dari penelitian adalah sebuah editor berbasis web yang dapat membantu membuat diagram transisi untuk membantu pengajaran tentang <i>Finite Automata</i>.</p> <p>Hasil dari penelitian yang penulis lakukan adalah sistem e-Learning yang terintegrasi dengan <i>State Based Code Editor</i> dimana cara kerjanya terinspirasi dari diagram</p>

						transisi pada <i>Finite State Automata</i> (FSA).
--	--	--	--	--	--	---

11. Waktu Pelaksanaan

Jadwal dan waktu pelaksanaan penelitian ini, yaitu :

No	Kegiatan	Bulan					
		Jan 2022	Feb 2022	Mar 2022	Apr 2022	Mei 2022	Juni 2022
1	Studi Kepustakaan						
2	Penerimaan Proposal Skripsi						
3	Pengumpulan dan Analisis <i>Data</i>						
4	Pembuatan Sistem/Program						
5	Pengujian Sistem/Program						
6	Penyelesaian Laporan Akhir						

12. Daftar Referensi

- Boud, D., & Feletti, G. I. (2013). The challenge of problem-based learning. In *The Challenge of Problem-based Learning*. <https://doi.org/10.4324/9781315042039>
- Dalziel, J. (2007). Implementing learning design : the Learning Activity Management System (LAMS). Downar.
- Hmelo-Silver, C. (2004). Problem Based Learning: What and How Do Students Learn? Educational Psychology Review. *Educational Psychology Review*. Springer, 16 (3).
- Jonassen, D. H., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development*, 47(1). <https://doi.org/10.1007/BF02299477>
- Jonassen, D., Mayes, T., & McAleese, R. (1993). A Manifesto for a Constructivist Approach to Uses of Technology in Higher Education. In *Designing Environments for Constructive Learning*. https://doi.org/10.1007/978-3-642-78069-1_12
- Oliver, R. (2001). Seeking best practice in online learning: Flexible Learning Toolboxes in the Australian VET sector. *Australasian Journal of Educational Technology*, 17(2). <https://doi.org/10.14742/ajet.1791>
- Asmawi, Syafei, & Yamin, M. (2019). Pendidikan Berbasis Teknologi Informasi Dan Komunikasi. *Prosiding Seminar Nasional Pendidikan*, 3.
- Hanum, N. S. (2013). Keefetifan e-learning sebagai media pembelajaran (studi evaluasi model pembelajaran e-learning SMK Telkom Sandhy Putra Purwokerto). *Jurnal Pendidikan Vokasi*, 3(1). <https://doi.org/10.21831/jpv.v3i1.1584>
- Handani, S. W., Suyanto, M., & Sofyan, A. F. (2016). PENERAPAN KONSEP GAMIFIKASI PADA E-LEARNING UNTUK PEMBELAJARAN ANIMASI 3 DIMENSI. *Telematika*, 9(1). <https://doi.org/10.35671/telematika.v9i1.413>
- Nurhayati. (2013). Pemberdayaan E-learning Sebagai Media Pembelajaran Ramah Lingkungan. *Jurnal Saintech*, 05(01).

- Mutia, I., & Leonard. (2015). Kajian Penerapan E-learning dalam Proses Pembelajaran di Perguruan Tinggi. *Faktor Exacta*, 6(4).
- Aditya, A. N. (2011). Jago php & MySQL. In *Ijns.Org*.
- A. C. Prof. Dr. Sri Mulyani. (2017). Metode Analisis dan Perancangan Sistem. In *Abdi SisteMatika*.
- Abdul Kadir. (2014). Pengenalan Sistem Informasi Edisi Revisi. In *Edisi Revisi*.
- Kristanto, A. (2018). Perancangan Sistem Informasi dan Aplikasinya (Edisi Revisi). *Gava Media*.
- Churcher, C. (2007). Beginning database design. In *Beginning Database Design*.
<https://doi.org/10.1007/978-1-4302-0366-7>
- Al-Fraihat, D., Joy, M., Masa'deh, R., & Sinclair, J. (2020). Evaluating E-learning systems success: An empirical study. *Computers in Human Behavior*, 102.
<https://doi.org/10.1016/j.chb.2019.08.004>
- Fujita, N. (2020). Transforming online teaching and learning: towards learning design informed by information science and learning sciences. *Information and Learning Science*, 121(7–8). <https://doi.org/10.1108/ILS-04-2020-0124>
- Henley, A. Z., Fleming, S. D., & Luong, M. v. (2017). Toward principles for the design of navigation affordances in code editors: An empirical investigation. *Conference on Human Factors in Computing Systems - Proceedings, 2017-May*.
<https://doi.org/10.1145/3025453.3025645>
- Dange, S., Kasture, R., Kadhao, A., Thorat, A., & Mhamane, S. (2020). Build: Web Services based Source Code Editor Integrate with Community Question Answer. *Proceedings of the 4th International Conference on Inventive Systems and Control, ICISC 2020*. <https://doi.org/10.1109/ICISC47916.2020.9171162>
- Kemdikbud, badan bahasa. (2016). Pedoman Umum Ejaan Bahasa Indonesia. In *Pedoman Umum Ejaan Bahasa Indonesi* (Vol. 4).
<https://doi.org/10.5851/kosfa.2016.36.6.807>
- Agung, B., 2017. CNN Indonesia. [Online]
Available at: <https://www.cnnindonesia.com/teknologi/20170728094848-185->

230919/indonesia-darurat-tenaga-programmer

[Accessed 17 1 2022].

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137. <https://doi.org/10.1016/j.jss.2017.11.045>

Bibik, I. (2018). How to Kill the Scrum Monster. In *How to Kill the Scrum Monster*. <https://doi.org/10.1007/978-1-4842-3691-8>

Cross, K. P., & Boud, D. (1987). Problem-Based Learning in Education for the Professions. *The Journal of Higher Education*, 58(4). <https://doi.org/10.2307/1981327>

Sekaran, U., & Bougie, R. (2009). Research Method for Business Textbook: A Skill Building Approach. *John Wiley & Sons Ltd*.

Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR)*, 22(4), 299-319.

Al-Fedaghi, S. (2020). Modeling the semantics of states and state machines. *Journal of Computer Science*, 16(7). <https://doi.org/10.3844/JCSSP.2020.891.905>

Hidayat, E. W., Rachman, A. N., & Azim, M. F. (2019). Penerapan Finite State Machine pada Battle Game Berbasis Augmented Reality. *J. Edukasi dan Penelit. Inform*, 5(1), 54.

Alawida, M., Teh, J. S., & Samsudin, A. (2019). An image encryption scheme based on hybridizing digital chaos and finite state machine. *Signal Processing*, 164, 249-266.

Jha, S., Behrens, J., Gkountouvas, T., Milano, M., Song, W., Tremel, E., ... & Birman, K. P. (2019). Derecho: Fast state machine replication for cloud services. *ACM Transactions on Computer Systems (TOCS)*, 36(2), 1-49.

Waseem, H. M., & Khan, M. (2019). A new approach to digital content privacy using quantum spin and finite-state machine. *Applied Physics B*, 125(2), 1-14.