

MAKALAH OPSI 2016

Rekayasa Aplikasi Generator Soal Essay “QuestiGo” Dengan Natural Language Processing(NLP) dan Random Equation Generator

Kelompok Bidang Penelitian : MATEMATIKA DAN REKAYASA

Bidang Ilmu : REKAYASA INFORMATIKA

Ketua Tim Peneliti		
Nama Lengkap	:	Muhammad Fahmi Rasyid
NIS	:	19989
Kelas	:	XI MIA 1
Anggota Peneliti		
1. Anggota 1		
Nama Lengkap	:	Dian Anggraini Pary Usamahu
NIS	:	19857
Kelas	:	XI IIS 1
Pembimbing		
Nama Lengkap	:	Drs. Suwondo
NIP	:	19640302 199103 1016
Bidang Studi yang diampu	:	Fisika

SMA MUHAMMADIYAH 1 YOGYAKARTA

KOTA YOGYAKARTA

2016

LEMBAR PENGESAHAN

1. Judul Makalah : Rekayasa Aplikasi Generator Soal Essay
: “QuestiGo” Dengan Natural Language
Processing dan Random Equation Generator
2. Kelompok Bidang : MATEMATIKA DAN REKAYASA
Penelitian
3. Bidang Ilmu : REKAYASA INFORMATIKA
4. Ketua Tim Penelitian:
Nama Lengkap: : Muhammad Fahmi Rasyid
NIS: : 19989
Kelas: : XI MIA 1
e-Mail: : rasyid.ufa@gmail.com
Asal Sekolah: : SMA Muhammadiyah 1 Yogyakarta
Alamat Sekolah: : (0274) 563739 telepon/faks: (0274) 519533

Menyatakan bahwa substansi ini, yang berjudul “Rekayasa Aplikasi Generator Soal Essay “*QuestiGo*” Dengan *Natural Language Processing* dan *Random Equation Generator*” belum pernah disertakan dalam lomba apapun, dan dikerjakan dengan melibatkan anggota peneliti sebanyak 1 orang, pembimbing sebanyak 1 orang, dengan rincian sebagai berikut:

Anggota Peneliti		
1. Anggota 1		
Nama Lengkap	:	Dian Anggraini Pary Usamahu
NIS	:	19857
Kelas	:	XI IIS 1
Pembimbing		
Nama Lengkap	:	Drs. Suwondo
NIP	:	19640302 199103 1016
Bidang Studi yang diampu	:	Fisika

Yogyakarta, 14 Juli 2016

Kepala Sekolah

Ketua Tim Peneliti

Tri Ismu Husnan Purwono S.H., M.M.
NIP/NBM: 634.951

Muhammad Fahmi Rasyid
NIS: 19989

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini,

Nama Lengkap : Muhammad Fahmi Rasyid
NIS : 19989
Kelas : XI MIA 1
Sekolah : SMA MUHAMMADIYAH 1 YOGYAKARTA
Alamat Sekolah : Jl. Gotong Royong II Petinggen, Karangwaru, Tegalrejo, Yogyakarta
Telepon/faks sekolah : (0274) 563739 / (0274) 519533
Alamat Rumah : Jl. Gotong Royong II Petinggen, Karangwaru, Tegalrejo, Yogyakarta
Telepon/HP : 085643094886

Menyatakan bahwa makalah ini, yang berjudul “Rekayasa Aplikasi Generator Soal Essay “*QuestiGo*” Dengan *Natural Language Processing* dan *Random Equation Generator*” adalah

- 1) Sepenuhnya ditulis oleh tim peneliti yang beranggotakan sebanyak 1 orang dengan rincian sebagai berikut

Anggota Peneliti		
Nama Lengkap	:	Dian Anggraini Pary Usamahu
NIS	:	19857
Kelas	:	XI IIS 1

- 2) Dikerjakan di bawah pembimbing,

Nama Lengkap	:	Drs. Suwondo
NIP	:	19640302 199103 1016
Bidang Studi yang diampu	:	Fisika

- 3) Orisinal karya tim peneliti ini, tanpa ada unsur plagiarisme baik dalam aspek substansi maupun penulisan.

Demikian surat pernyataan ini dibuat dengan sebenar-benarnya. Bila dikemudian hari ditemukan kekeliruan, maka kami bersedia menanggung semua risiko atas perbuatan yang kami lakukan sesuai dengan aturan yang berlaku.

Yogyakarta, 14 Juli 2016

Pembimbing Penelitian,

Yang membuat pernyataan
Ketua tim penelitian,

Materai 6000

Drs. Suwondo
NIP:19640302 199103 1016

Muhammad Fahmi Rasyid
NIS: 19989

Kepala Sekolah

Tri Ismu Husnan Purwono S.H., M.M.

NIP/NBM: 634.951

ABSTRAK

Kecurangan dalam ujian termasuk dalam permasalahan besar pendidikan di Indonesia, penyelesaian menggunakan soal yang beragam dapat dibilang efektif mengatasi kecurangan tersebut, namun kebiasaan itu masih saja dilakukan. Maka dari itu perlu pembiasaan dini soal yang beragam agar siswa tidak terbiasa menyontek hasil orang lain. Namun, pembuatan soal dalam paket-paket soal terkadang memiliki tingkat kesulitan yang tidak sama sehingga hasil penilaian tidak valid, dan pembuatan secara manual terbilang lama. Berdasar masalah tersebut kami berupaya merekayasa aplikasi yang dapat membuat soal dengan kecepatan tinggi, dan dengan tingkat kesulitan yang sama. Kami meneliti tingkat kesulitan soal berdasar pada beberapa pembentuknya, yaitu jumlah variabel, jumlah rumus, bentuk angka, dan kerumitan bahasa. Setelah ditemukan hubungannya, kami membuat algoritma khusus untuk menemukan tingkat kesulitan soal tersebut. Setelah itu kami menghitung kecepatan pembuatan soal, sehingga dapat ditemukan tingkat kesulitan yang seimbang dan pembuatan yang cepat. Dengan penelitian ini diharapkan dapat membantu guru dalam pembuatan soal untuk kegiatan evaluasi dan latihan dalam kegiatan belajar mengajar di sekolah.

Keyword: Natural Language Processing, Soal, Generator, Kesulitan Soal, Esai

KATA PENGANTAR

Puji syukur peneliti panjatkan ke hadirat Allah SWT berkat limpahan rahmat dan petunjuk-Nya penulis dapat menyusun tugas akhir penelitian dengan judul “Rekayasa Aplikasi Generator Soal Essay “QuestiGo” Dengan Natural Language Processing dan Random Equation Generator”. Karya tulis ini dibuat untuk mengikuti kompetisi Olimpiade Penelitian Siswa Indonesia(OPSI). Di samping itu karya ini dibuat untuk meningkatkan mutu belajar diri kami sendiri, sehingga dapat mengembangkan karya lain yang bermanfaat untuk bangsa, negara, serta agama. Amin.

Keberhasilan penyusunan laporan penelitian ini tidak terlepas dari bantuan berbagai pihak, untuk itu pada kesempatan baik ini penulis mengucapkan banyak terima kasih kepada:

1. Bp. Tri Ismu Husnan Purwono, S.H. M. M. selaku Kepala Sekolah SMA Muhammadiyah 1 Yogyakarta.
2. Bp. Sadono S. Pd M. Pd, Bp. Darmansyah S.H., Bp. Zulbahri Sutan Bagindo S.E, Bp. Abdul Qodir S.Th.I dan Ibu Fitri Sari Sukmawati S. Pd. Selaku Wakasekur SMA Muhammadiyah 1 Yogyakarta.
3. Bp. Drs. Suwondo, sebagai mentor dan pembimbing kami.
4. Seluruh Keluarga yang memberikan bantuan baik moral maupun material.
5. Teman-teman MIA 1, IIS 1 dan KIR (Kelompok Ilmiah Remaja).
6. Semua pihak yang telah membantu dalam menyelesaikan penelitian ini yang tidak dapat disebutkan satu-persatu.

Penulis menyadari dalam penulisan penelitian ini masih banyak kekurangan dan jauh dari kesempurnaan, untuk itu masukan konstruktif dari pembaca sangat kami harapkan untuk perbaikan karya tulis kami di kemudian hari.

Wasalamualaikum Wr. Wb.

Yogyakarta, 4 Juli 2016

Tim Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	viii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah.....	2
C. Pertanyaan Penelitian.....	2
D. Tujuan Penelitian	2
E. Manfaat Penelitian	2
BAB II LANDASAN TEORI	3
A. Kecerdasan Artifisial	3
1. Sistem Pakar	3
2. Pemrosesan Bahasa Alami (<i>Natural Language Processing</i>)	4
B. Pembangkit Bilangan Acak (<i>Random Number Generator</i>)	4
C. <i>Regular Expression</i>	5
1. <i>Regular Expression Parser</i>	5
D. Pembangkit Persamaan Acak (<i>Random Equation Generator</i>)	5
E. Penilaian Kesulitan Soal	6
1. Konvensional.....	6
2. Metode QuestiGo (Generator).....	6
F. Kerangka Pikir	8
BAB III METODE PENELITIAN	9
A. Jenis Penelitian.....	9
B. Variabel Penelitian.....	9
1. Variabel Bebas	9
2. Variabel Terikat	9
3. Variabel Kontrol.....	9
C. Waktu dan Tempat.....	9
1. Timeline Penelitian	9
2. Tempat:	10

D.	Subjek Penelitian	10
E.	Alat dan Aplikasi yang Digunakan	10
1.	Alat: Laptop, Alat tulis dan Jaringan internet	10
2.	Aplikasi yang digunakan:.....	10
F.	Metode Pengumpulan Data.....	11
1.	Wawancara dan Pengamatan Sebelum Pembuatan Perangkat.....	11
G.	Teknik Analisis Data.....	12
1.	Tingkat Kesulitan Soal	12
2.	Kecepatan Pembuatan.	12
H.	Tahapan pembuatan aplikasi.....	13
BAB IV ANALISIS HASIL PENELITIAN		14
A.	Aplikasi QuestiGo.....	14
B.	Analisis	15
1.	Tingkat kesulitan soal	15
2.	Kecepatan Pembuatan Soal	18
C.	Hasil	19
1.	Tingkat kesulitan soal	19
2.	Kecepatan pembuatan soal	19
BAB V SIMPULAN DAN SARAN		20
A.	Kesimpulan	20
B.	Saran	20
DAFTAR PUSTAKA.....		21
LAMPIRAN		22

DAFTAR GAMBAR

Gambar 1: Implementasi Sistem Pakar	3
Gambar 2: Diagram Implementasi NLP	4
Gambar 3: Contoh Pola soal dan hasil	5
Gambar 4: Diagram Perbandingan Pembuatan soal.	7
Gambar 5: Diagram Kerangka Pikir	8
Gambar 6: Tampilan Antarmuka QuestiGo.....	14
Gambar 7: Diagram rekapitulasi variabel pembentuk kesulitan soal.	19

DAFTAR TABEL

Tabel 1: Timeline Penelitian	9
Tabel 2: Distribusi frekuensi hubungan jumlah variabel dengan tingkat kesulitan soal	15
Tabel 3: Distribusi frekuensi hubungan jumlah rumus dengan tingkat kesulitan soal.....	16
Tabel 4: Distribusi frekuensi hubungan penggunaan angka dengan tingkat kesulitan soal	16
Tabel 5: Distribusi frekuensi hubungan pemakaian bahasa dengan tingkat kesulitan soal.....	17
Tabel 6: Kecepatan pembuatan soal (ms/soal) dengan waktu kejadian(detik).....	18

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam dunia pendidikan yang sekarang berlangsung. Menyontek merupakan hal yang sering dikatakan wajar bagi sebagian besar pelajar dan bahkan sebagian kecil guru di Indonesia. Terbukti dari tingkat kecurangan dalam Ujian Nasional, yang setiap tahunnya meningkat 0,3 % dan keuntungan dari penjualan kunci jawaban mencapai 140 Juta per-tahun. Bayangkan kerugian negara akibat masalah ini. akan munculnya para pemuda yang tidak kompeten, dan berpotensi merusak generasi bangsa.

Usaha pemerintah dalam mengatasi praktik tersebut adalah salah satunya membuat paket soal berdasar pada jumlah peserta pada regional tersebut. Kebijakan ini sangat baik dan efektif. Namun, praktik menyontek masih pula terjadi. Hal ini seharusnya diatasi dari akar permasalahannya, yaitu kebiasaan menyontek pelajar yang mereka lakukan saat Ujian Sekolah, Ujian Tengah Semester, Ulangan Harian, bahkan Pekerjaan Rumah. Menyebabkan ketergantungan atau kebiasaan yang buruk pada Ujian Nasional atau bahkan di kehidupannya.

Pembuatan paket soal yang beragam merupakan metode yang efektif yang harusnya diaplikasikan juga pada sistem yang lebih kecil, yaitu pada sistem sekolah. Hal ini akan membantu mengatasi kebiasaan menyontek karena siswa tidak akan mudah menanyakan jawaban langsung kepada teman, namun apabila itu pekerjaan rumah, siswa dengan sendirinya akan menanyakan cara mengerjakan soal, bukan jawaban langsung. Apabila Ulangan harian, kemungkinan menyontek akan menjadi kecil. Tentu saja didukung pengawasan dari guru yang baik.

Namun pembuatan soal yang beragam sangat tidak memungkinkan jika dibuat dalam waktu singkat, dengan tingkat kesulitan yang sama juga bentuk soal yang bervariasi. Untuk itulah perlu dibuat komputerisasi pembuatan naskah soal yang fleksibel juga cepat agar mempermudah proses pembuatan soal yang beragam.

Dengan membuat aplikasi ini, di harapkan dapat meningkatkan integritas sekolah dalam ujian nasional juga peningkatan pemahaman melalui kerja kelompok, sesuai dengan kurikulum 2013. Melalui pembiasaan soal yang tidak monoton dan seimbang.

B. Rumusan Masalah

1. Pembuatan soal yang beragam yang dapat mengurangi kebiasaan menyontek memakan waktu lama.
2. Pembuatan soal yang beragam oleh guru secara manual tingkat kesulitannya tidak sama.

C. Pertanyaan Penelitian

1. Bagaimana cara membuat aplikasi yang dapat mempermudah pembuatan soal pada guru?
2. Bagaimana membuat dengan tingkat kesulitan yang bisa ditentukan?

D. Tujuan Penelitian

1. Mengembangkan sebuah aplikasi yang digunakan untuk membuat soal yang beragam dengan cepat dan mudah.
2. Mengembangkan aplikasi yang dapat membuat soal dengan tingkat kesulitan yang bisa ditentukan.

E. Manfaat Penelitian

Penelitian ini diharapkan dapat memberi dampak positif:

1. Bagi Siswa, mampu membiasakan soal-soal variatif yang sangat berguna untuk mereka, juga mengurangi kebiasaan menyontek.
2. Bagi Pengajar, mampu mempermudah pengajar membuat soal secara cepat, dan mempermudah pengajar dalam memantau perkembangan siswa.
3. Bagi Peneliti, mampu meningkatkan pengetahuan dan ketrampilan meneliti, juga meningkatkan pengalaman peneliti.
4. Bagi pengembangan ilmu pengetahuan, diharapkan dapat menambah pengetahuan tentang teknik penilaian soal sebelum pembuatan, juga untuk digunakan di penelitian lanjutan

BAB II

LANDASAN TEORI

A. Kecerdasan Artifisial

Kecerdasan Artifisial merupakan bentuk rekayasa program yang memiliki tujuan agar sebuah komputer mampu berpikir selayaknya manusia. Seperti menentukan pilihan, mengolah data dan mencari pemecahan masalah terbaik. (McCarthy, 1989)

Menurut peneliti, berdasar dari kesimpulan di atas. Pembuatan kecerdasan artifisial sangat diperlukan, sehingga dalam penelitian ini, peneliti menggunakan penggabungan System Pakar dan *Natural Language Processing*.

1. Sistem Pakar

a. Definisi

Sistem Pakar (*Expert System*) merupakan salah satu jenis Kecerdasan Buatan yang berfungsi untuk memecahkan masalah heuristik yang berasal dari kesimpulan para pakar yang disimpan di dalam data komputer. (Bartee, 1988)

Sistem pakar pertama kali dibuat untuk kebutuhan medis sebagai penyelesaian dari pertanyaan pasien tentang gejala penyakit, kemudian di kembangkan ke berbagai bidang setelah itu.

Peneliti bermaksud ingin menggunakan sistem pakar, dalam konteks penelitian ini pakar yang dimaksud adalah guru, sebagai dasar pengetahuan program, sehingga dapat membangkitkan soal.

b. Implementasi



Gambar 1: Implementasi Sistem Pakar

2. Pemrosesan Bahasa Alami (*Natural Language Processing*)

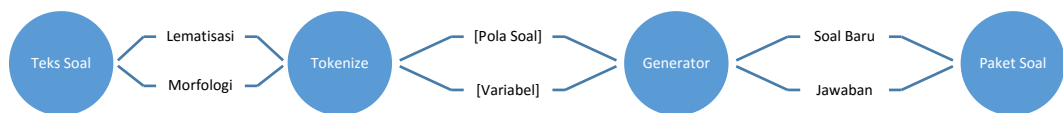
a. Definisi

Pemrosesan Bahasa Alami (*Natural Language Processing*) merupakan bentuk mekanisme pengolahan mutakhir dalam memroses data alami, baik dalam bentuk tulis maupun suara, sehingga dapat diolah oleh mesin (Chomsky, 1957)

Peneliti menggunakan NLP untuk membaca materi, yang nantinya akan diproses menjadi sebuah soal. Mengubah Pernyataan menjadi pertanyaan.

b. Implementasi

Pembuatan pola soal berdasarkan pemroses bahasa pemrograman dapat digambarkan sebagai berikut:



Gambar 2: Diagram Implementasi NLP

B. Pembangkit Bilangan Acak (*Random Number Generator*)

Random Number Generator (RNG) merupakan salah satu algoritma yang dipakai sebuah program untuk membangkitkan bilangan acak berdasarkan beberapa faktor buatan yang sering juga disebut *Pseudo-Random*. (Sedgewick & Wayne, 2011)

Sejak dulu, cara pertama untuk membuat bilangan acak dalam teknologi komputer dibuat berdasarkan *pseudo-random* generator, yang mulai muncul di awal tahun 50-an bersamaan dengan kemunculan mesin komputer elektronik pertama, dengan menggunakan tabel bilangan acak yang digunakan oleh para peneliti. (Corporation, 1955)

pseudo-random generator merupakan algoritma yang membuat barisan bilangan yang digunakan untuk memperkirakan sifat-sifat dari bilangan acak, yang sebenarnya ditentukan secara matematis berdasarkan vektor yang diberikan, yang disebut *seed* (Neumann, 1951), (Knuth, 1997)

Seed, sebenarnya merepresentasikan satu-satunya ketidakpastian dalam generator, yang merupakan set untuk awal sistem, umumnya tak salah jika memulai sebuah *pseudo-random generator* dengan barisan vektor acak sebenarnya, yang bisa diambil, sebagai contoh, dari bit terkecil dari waktu pada sistem. Tentu saja prosesnya tidak benar-benar acak, seperti yang dikatakan John Neumann, pelopor dari komputer modern dan teori dari permainan. Mengatakan “Semua orang yang memilih metode aritmatika untuk memproduksi bilangan acak, tentu saja ditemukan kesalahan di dalamnya (Neumann, 1951)

Nyatanya karena semua *pseudo-random generator* berjalan pada algoritma yang pasti, keluarannya pasti tak lepas dari pengulangan. Karena sistemnya berulang dan sangat dapat ditebak setelah beberapa periode. Kemudian, karena algoritma berdasar pada awalan yang diberikan, ia tak bergantung pada kejadian dari luar, namun hanya bergantung pada kondisi internal dalam sistem, menjadi mesin yang bisa ditebak, dengan periode yang sama, akan menghasilkan keadaan yang sama, yang mana akan mengulangi bilangan acak yang sudah ada, berulang, dan selamanya.

C. Regular Expression

RE merupakan salah satu perangkat komputer modern, yang sangat berguna membantu mengkualifikasikan teks menjadi beberapa bagian sesuai dengan *pattern* (pola) tertentu. (Sedgewick & Wayne, 2011)

RE digunakan peneliti untuk menemukan pola yang digunakan sebagai tokenizer pada parser.

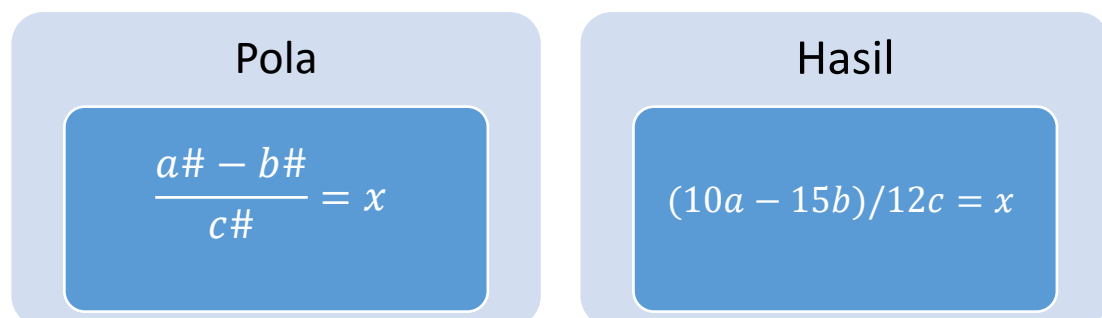
1. Regular Expression Parser

Parser merupakan program yang dibuat untuk mengolah sebuah data menjadi token-token yang nantinya akan diolah kembali menjadi barisan perintah yang dapat dipahami komputer (Alwi, 2002)

Dalam penelitian ini kami menggunakan Universal Parser Generator, Parser Generator yang dikembangkan oleh peneliti yang merupakan *Open-Source software*. Untuk membuat parser tata bahasa Indonesia sederhana.

D. Pembangkit Persamaan Acak (*Random Equation Generator*)

Pembangkit persamaan acak merupakan sebuah algoritma yang membuat persamaan berdasarkan pola persamaan yang diberikan. Dengan mengisi setiap variabel yang ada menggunakan pseudo-random generator. Pola yang diberikan seperti pada gambar:



Gambar 3: Contoh Pola soal dan hasil

E. Penilaian Kesulitan Soal

1. Konvensional

Pembuatan soal pada biasanya menggunakan teknik manual, dengan menerapkan masalah sehari-hari pada materi yang akan diberikan. Soal yang dibuat banyak, dan diperkirakan memiliki tingkat kesulitan yang sama. (Hamalik, 1989)

Pengujian soal sampai saat ini dilakukan dengan cara memberikannya kepada sejumlah siswa, kemudian menghitung indeks dengan membandingkan jumlah salah, dan jumlah benar (Clarke, 2003), dengan rumus di bawah ini:

a. Menghitung rata-rata skor untuk tiap butir soal dengan rumus :

$$Mean = \frac{\text{Jumlah skor peserta didik tiap soal}}{\text{Jumlah peserta didik}}$$

b. Menghitung tingkat kesukaran (Tk) dengan rumus :

$$Tk = \frac{Mean}{\text{Skor maksimum per soal}}$$

c. Membandingkan tingkat kesukaran dengan kriteria berikut :

0,00 – 0,30 = sukar

0,31 – 0,70 = sedang

0,71 – 1,00 = mudah

d. Membuat penafsiran tingkat kesukaran dengan cara membandingkan koefisien tingkat kesukaran (poin b) dengan kriteria (poin c).

2. Metode QuestiGo (Generator)

Pembuatan soal secara modern merupakan cara baru yang bisa dibuat dengan mempertimbangkan metode dan algoritma yang memungkinkan untuk membuat soal dengan tingkat kesulitan sama dan dengan kecepatan tinggi.

Algoritma yang digunakan adalah *Random Equation Generator*. Dengan menggunakan pola yang sudah dibuat sebelumnya, program akan mengisi tiap variabel untuk ditemukan ke dalam pola soal. Kemudian membandingkan dengan hasil.

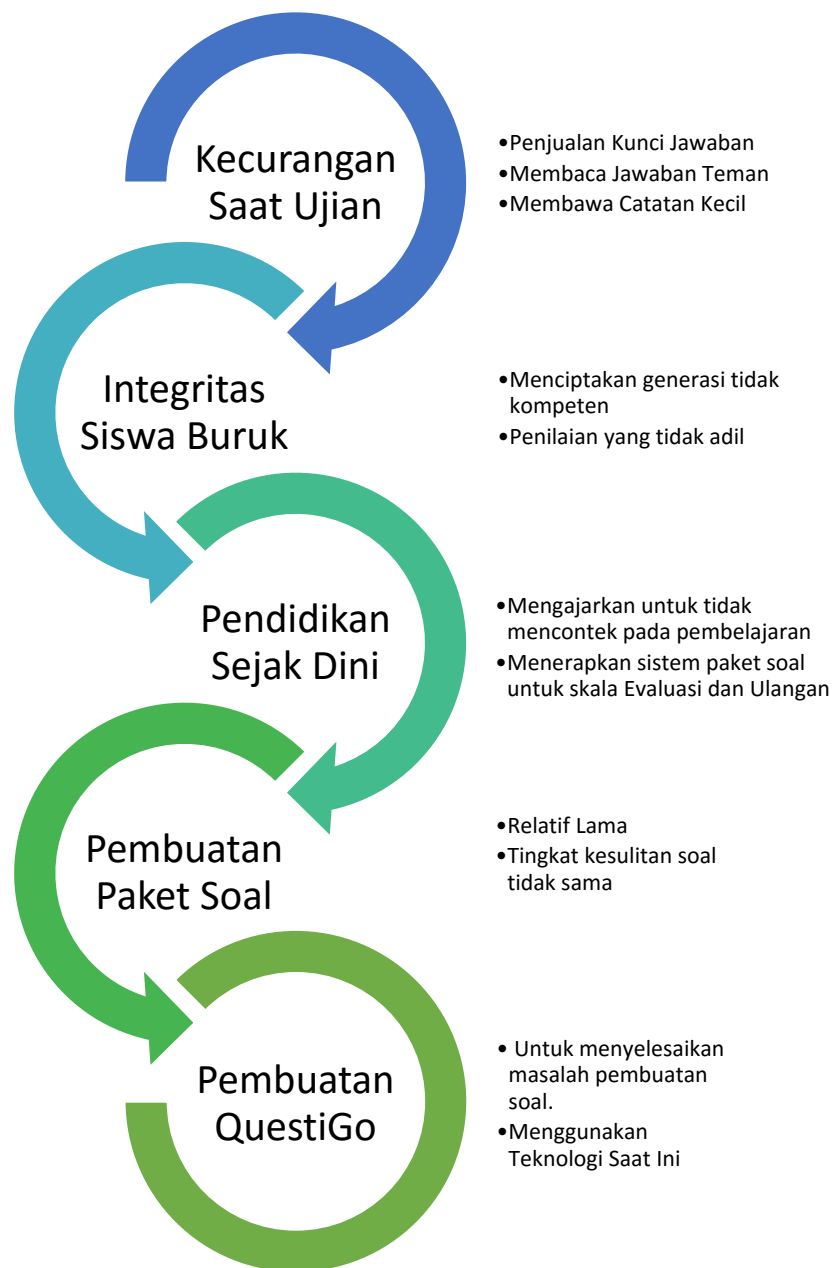


Gambar 4: Diagram Perbandingan Pembuatan soal.

Menurut (Qualifications and Curriculum Authority, 2003) faktor yang mempengaruhi tingkat kesulitan soal ada beberapa macam antara lain: Tata Bahasa dan Variabel yang diberikan.

Peneliti menambahkan Banyaknya rumus, dan Angka yang akan dihitung sebagai penguat algoritma. Dikarenakan perbedaan mendasar dari pembuatan soal konvensional dengan Questigo adalah, dapat membuat soal yang beragam dengan kecepatan sama dan tingkat kesulitan yang bisa diatur **sebelum** soal dibuat.

F. Kerangka Pikir



Gambar 5: Diagram Kerangka Pikir

Menurut Gambar 5, disebutkan bahwa masalah utama dalam pembuatan soal konvensional relatif lama dan tingkat kesulitan yang dibuat tidak selalu sama. Sehingga, tidak efisien dan tidak adil dalam penilaian.

BAB III METODE PENELITIAN

A. Jenis Penelitian

Untuk memecahkan masalah penelitian ini, peneliti melakukan Rekayasa Aplikasi dan Eksperimen, peneliti membuat Aplikasi kemudian mengujinya kecepatan pembuatan, setelah itu peneliti mengujikan hasil dari aplikasi tersebut kepada siswa SD, SMP, dan SMA, untuk menguji kesamaan bobot soal melalui Survei langsung dan Online.

B. Variabel Penelitian

Beberapa variabel yang digunakan dalam penelitian berikut adalah:

1. Variabel Bebas

Variabel bebas dalam penelitian ini adalah Jumlah soal dan Jumlah variabel tiap soal

2. Variabel Terikat

variabel terikat yang terdapat pada penelitian ini adalah tingkat kesulitan soal, dan kecepatan pembuatan soal.

3. Variabel Kontrol

Variabel kontrol dalam penelitian ini adalah kecepatan komputer yang digunakan.

C. Waktu dan Tempat

1. Timeline Penelitian

Tabel 1: Timeline Penelitian

No.	Keterangan	1				2				3				4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Perencanaan Ide																
2	Kajian Awal																
3	Bimtek GPIR																
4	Mengikuti GPIR																
5	Kajian Pustaka																
6	Pembuatan Proposal																
7	Pembuatan Aplikasi																
8	Pengujian																
9	Pembuatan Perangkat Pameran																

No.	Keterangan	3				4				5				6				7			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
10	Persiapan <i>Sagasitas Research Exhibition</i>																				
11	<i>Sagasitas Research Exhibition</i>																				
12	Eksperimen																				
13	Penyusunan Laporan																				
14	Pengesahan Laporan																				
15	Pengiriman Naskah																				

2. Tempat:

- SMA Muhammadiyah 1 Yogyakarta, Jl. Gotong royong II, Petinggen Tegalrejo Karangwaru, Kota Yogyakarta.
- Rumah peneliti
 - Asrama As-Sakinah SMA Muhammadiyah 1 Yogyakarta.
 - Campursari, Bulu, Temanggung, Jawa Tengah
 - Jl. Nologaten Perum. Jogja Town House 2 No. B6
- Perpustakaan Kota Yogyakarta +
- Perpustakaan Daerah Istimewa Yogyakarta

D. Subjek Penelitian

Subjek penelitian ini adalah siswa SMP yang memiliki kriteria umum sebagai berikut.

- Memiliki tingkat kecerdasan normal dan di atas normal.
- Berumur 14-16 tahun.
- Mampu bertanggungjawab atas sikap dan tindakan yang dilakukan.

E. Alat dan Aplikasi yang Digunakan

1. Alat: Laptop, Alat tulis dan Jaringan internet

2. Aplikasi yang digunakan:

a. Sistem Operasi Windows

Perangkat Sistem operasi yang digunakan merupakan OS Windows, berbasis desktop yang memungkinkan kemudahan dalam proses analisis dan pengolahan data secara besar, selain itu memudahkan dalam pencetakan soal.

b. NET FRAMEWORK 4.5

Net Framework merupakan sebuah teknologi pemrograman yang berisi antarmuka pustaka dasar (*Library*) yang disediakan untuk memudahkan *programmer* membuat Aplikasi yang dapat di jalankan di semua sistem operasi tanpa mengubah kode yang sudah ada. *Framework* versi 4.5 sudah terpasang di komputer berbasis Windows 7 ke atas, sehingga tidak perlu menginstal lagi.

c. Visual Studio Community 2015

Peneliti menggunakan IDE VSC, selain karena gratis, IDE (*Integrated Development Environment*) ini memiliki banyak fitur yang membantu mengumpulkan data kecepatan, dan data lain yang mendukung.

Pembuatan Kode dan Antarmuka aplikasi sepenuhnya melalui VSC, menggunakan *Framework* WPF(*Windows Presentation Foundation*)

F. Metode Pengumpulan Data

1. Wawancara dan Pengamatan Sebelum Pembuatan Perangkat

Kami melakukan pengumpulan data sebelum membuat perangkat lunak ini, dengan mencari literasi dan wawancara dengan beberapa guru SMA di sekolah kami. Untuk mengetahui apa hal yang penting untuk diperhitungkan dalam pembuatan soal.

Tabel 4. Data wawancara sebelum pembuatan aplikasi

No.	Nama	Pendapat
1.	Suwondo S. Pd	Fungsi soal memiliki 3 jenis, sebagai latihan, sebagai evaluasi belajar dan pengayaan. Tingkat kesulitan ditentukan oleh jumlah turunan rumus, dan banyaknya variabel.
2.	Ichsan Yunianto Nuansa Putra S. Pd	Kompleksitas Bahasa dan Angka mempengaruhi tingkat kesulitan soal.

G. Teknik Analisis Data

Teknik analisis data yang digunakan adalah Teknik Deskriptif Kualitatif. Setelah data dikumpulkan, selanjutnya adalah verifikasi data. Setelah mengumpulkan data penilaian soal, kemudian dengan verifikasi data, diharapkan dapat ditemukan hasil yaitu faktor pembentuk tingkat kesulitan soal.

Kemudian dengan faktor tersebut, dibuat algoritma untuk membuat soal dengan tingkat kesulitan yang ditentukan. Langkah yang akan dilakukan adalah sebagai berikut:

1. Tingkat Kesulitan Soal

Dengan melakukan survei tingkat kesulitan soal, diberikan 5 soal untuk setiap jenjang pendidikan, SD, SMP dan SMA. Tiap soal dengan Indeks Kesulitan yang dirahasiakan. Subjek diminta untuk menilai soal tersebut berdasarkan variabel, kemudian menghubungkan dengan hasil yang sebenarnya.

2. Kecepatan Pembuatan.

Dengan menjalankan program, diberikan pola soal sederhana, biasa, dan kompleks dengan masukan jumlah soal 1000 buah yang akan dibuat, dan menghitung berapa waktu yang diperlukan untuk membuat 1 soal.

H. Tahapan pembuatan aplikasi.

1. Parser

Pembuatan parser terdiri dari 3 kelas. Tokenizer, Lemmatizer dan Parser itu sendiri. Kelas tokenizer berisi kelas pembagi kata dan kalimat. Sehingga membentuk Parse Tree. Parse Tree merupakan perangkat yang sangat penting untuk mengolah token-token, sehingga dapat digunakan sebagai masukan pola. Untuk mempermudah *reverse tracking*. Pola yang dibuat berupa larik kalimat (*Array of Sentence*) di dalamnya terdapat larik kata(berisi token-token atau. Kata tunggal dan tanda baca.

Parser mengolah kalimat agar dapat diubah menjadi objek yang mudah diolah di kelas selanjutnya.

a. Tokenizer.

Kelas ini mengubah teks menjadi token-token, mengubah struktur menjadi sistem yang lebih mudah. Hasil token-token akan dimasukkan ke dalam bentuk *List<Token>*

b. Lemmatizer.

Lemmatizer berfungsi untuk mengubah kata yang berimbuhan menjadi kata dasar. Sehingga akan memperkecil ukuran kalimat. Berguna juga untuk membuat kalimat lebih fleksibel.

c. Parser.

Kelas parser berisi kumpulan kelas di atas, dengan memasukkan teks, diolah kemudian diolah menjadi token-token, lalu tiap token diolah dengan Lemmatizer. Sehingga dapat dibuat pola soal yang akan digunakan.

3. *Expression Solver*

Expression solver hanya terdiri dari 1 kelas. Berguna untuk mencari hasil dari sebuah ekspresi matematis. Sehingga jawaban yang dibuat oleh generator valid. Penyelesai Ekspresi dapat juga digunakan sebagai penyelesai soal fisika dan soal lainnya. Sehingga lebih mudah untuk menentukan tingkatan maksimum atau batasan pada soal.

4. *Randomizer*

Kelas *randomizer* merupakan kelas pengacak yang berfungsi untuk mengolah bilangan acak, memanggilnya untuk dimasukkan ke dalam soal.

5. Generator

Menggunakan komponen di atas, generator mengubah teks soal menjadi *template* kemudian menyusun pola sesuai *template*, lalu mengubahnya menjadi soal kembali, setelah itu dilakukan proses pengecekan soal menggunakan Equation Solver

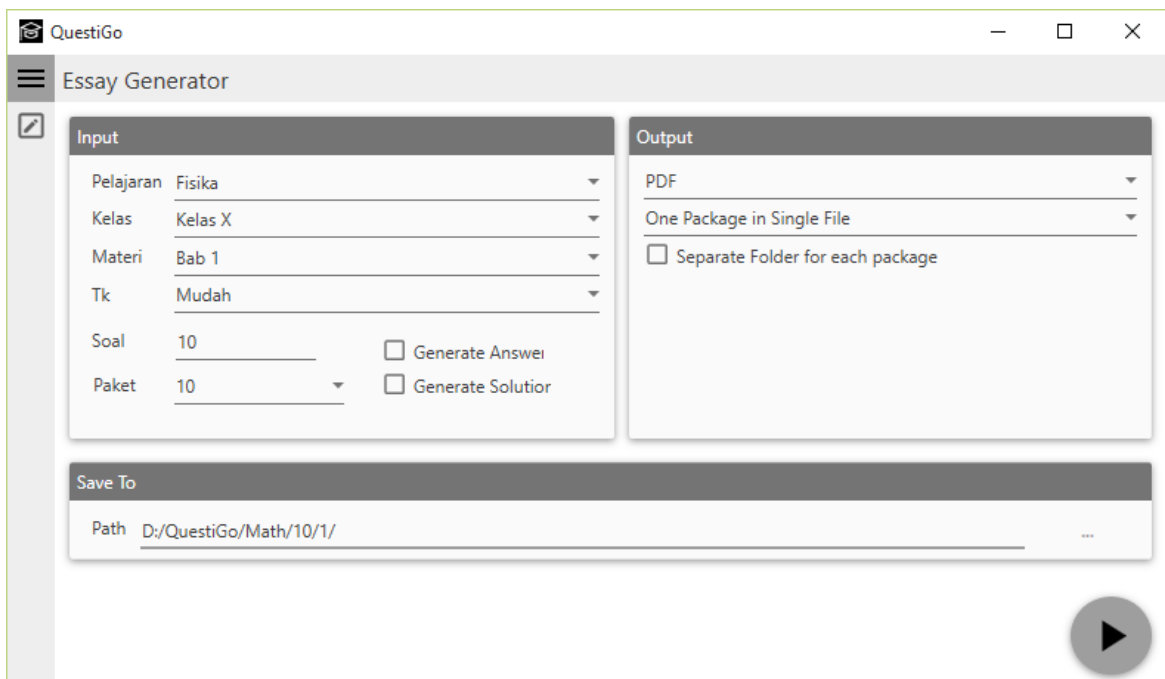
BAB IV

ANALISIS HASIL PENELITIAN

Pada bab ini akan dikemukakan data penelitian dan analisisnya, serta hasil yang didapatkan dari penelitian. Bab ini dibagi menjadi 2 analisis dan hasil, sedangkan yang di olah ada 2 yaitu tingkat kesulitan soal dan kecepatan pembuatan.

A. Aplikasi QuestiGo

Aplikasi QuestiGo berbentuk formulir, dengan tiga masukan. *Input*, *Output*, dan *Save To*. Grup *Input*, berisi tentang pilihan Pelajaran, Kelas, Materi bab, Kesulitan, jenis soal yang akan dibuat, kemudian jumlah soal yang akan dibuat. Group *Output* berisi tentang keluaran yang akan dibuat oleh aplikasi, bentuk keluaran (PDF, DOCX, TXT, HTML) cara pembagian paket. Grup *SAVE TO*, berisi lokasi hasil generator. Dan yang terakhir tombol *START* untuk memulai generator. Di bagian samping kiri, terdapat *sidebar* yang berisi perangkat lain sebagai pembantu QuestiGo.



The screenshot shows the 'QuestiGo' application window titled 'Essay Generator'. It features a sidebar on the left with a menu icon. The main area is divided into three sections: 'Input', 'Output', and 'Save To'. The 'Input' section contains dropdown menus for 'Pelajaran' (Fisika), 'Kelas' (Kelas X), 'Materi' (Bab 1), and 'Tk' (Mudah), along with text input fields for 'Soal' (10) and 'Paket' (10), and checkboxes for 'Generate Answer' and 'Generate Solution'. The 'Output' section has dropdowns for 'PDF' and 'One Package in Single File', and a checkbox for 'Separate Folder for each package'. The 'Save To' section includes a 'Path' field with the value 'D:/QuestiGo/Math/10/1/'. A large play button is located at the bottom right.

Gambar 6: Tampilan Antarmuka QuestiGo

B. Analisis

1. Tingkat kesulitan soal

Pada penelitian ini dicari rumus yang dapat digunakan untuk menentukan tingkat kesulitan soal berdasarkan hubungan komponen soal terhadap tingkat kesulitannya. Rumus tersebut digunakan untuk membuat algoritma pembuatan soal.

a. Hubungan jumlah variabel dengan kesulitan soal

Dari hasil penyebaran kuesioner diperoleh persepsi responden terhadap soal-soal yang telah diberikan dan hubungan jumlah variabel dengan kesulitan soal, dapat dilihat dari tabel berikut:

Tabel 2: Distribusi frekuensi hubungan jumlah variabel dengan tingkat kesulitan soal

Nilai	Sulit	Sedang	Mudah
1	18	30	22
2	25	30	26
3	45	19	23
4	35	34	30
5	25	20	46
6	23	47	25
7	16	17	22
8	18	20	18
9	35	23	28

Jumlah variabel yang dimaksud adalah banyaknya angka yang akan dihitung, semisal $[y=3x+z]$ terdapat 3 variabel, dan $[f=m*v/s]$ terdapat 4 variabel. Dengan memberikan soal dengan jumlah variabel yang berbeda. Dicari hubungan kesulitan soalnya.

Dalam Tabel 2 terdapat tiga jenis soal yang berikan, dalam hubungan dengan jumlah variabel, ditemukan 45 responden (19%) memberikan nilai 3 pada soal sulit, lalu 47 responden (20%) memberi nilai 6 untuk soal sedang, dan 46 responden (19.5%) memberikan nilai 5 untuk soal mudah.

Dapat dilihat bahwa variabel rendah (3 poin) menghasilkan kesulitan soal tinggi, kemudian soal sedang dan mudah memiliki hubungan dengan variabel yang hampir sama (6 dan 5 poin). Sehingga dapat disimpulkan bahwa jumlah variabel berbanding terbalik dengan tingkat kesulitan.

b. Hubungan jumlah rumus dengan kesulitan soal

Dari hasil penyebaran kuesioner diperoleh persepsi responden terhadap soal-soal yang telah diberikan dan hubungan jumlah rumus dengan kesulitan soal, dapat dilihat dari tabel berikut:

Tabel 3: Distribusi frekuensi hubungan jumlah rumus dengan tingkat kesulitan soal

Nilai	Sulit	Sedang	Mudah
1	13	22	33
2	11	26	34
3	11	23	47
4	15	30	31
5	11	46	20
6	5	25	23
7	42	22	12
8	72	18	23
9	60	28	17

Dalam penelitian ini, jumlah rumus yang dimaksud adalah rumus yang diperlukan untuk menyelesaikan tiap soal. Misal $[a/b = c]$ dan $[b*r = d]$ maka jumlah rumus tergantung soal yang ditanyakan, apakah memakai keduanya. Atau hanya salah satu. Inilah yang dihitung dalam pembuatan algoritma.

Dalam Tabel 3 terdapat tiga jenis soal yang berikan, dalam hubungan dengan jumlah rumus, ditemukan 42 responden (18%) memberikan nilai 7 pada soal sulit, lalu 46 responden (20%) memberi nilai 5 untuk soal sedang, dan 47 responden (19.5%) memberikan nilai 3 untuk soal mudah.

Dapat dilihat bahwa rumus yang banyak (7 poin) menghasilkan kesulitan soal tinggi. Kemudian, rumus sedang (5 poin) juga menghasilkan kesulitan soal Sedang, dan tingkat kesulitan Mudah memiliki rumus yang sedikit (3 poin). Sehingga dapat disimpulkan bahwa jumlah rumus yang harus digunakan berbanding lurus dengan tingkat kesulitan.

c. Hubungan bentuk angka dengan kesulitan soal

Dari hasil penyebaran kuesioner diperoleh persepsi responden terhadap soal-soal yang telah diberikan dan hubungan bentuk angka dengan kesulitan soal, dapat dilihat dari tabel berikut:

Tabel 4: Distribusi frekuensi hubungan penggunaan angka dengan tingkat kesulitan soal

Nilai	Sulit	Sedang	Mudah
1	49	17	7
2	44	32	11
3	47	46	14
4	37	50	12
5	7	43	47
6	17	17	61
7	15	16	50
8	11	8	26
9	13	11	12

Dalam Tabel 4 terdapat tiga jenis soal yang diberikan, dalam hubungan dengan bentuk angka, ditemukan 49 responden (21%) memberikan nilai 1 pada soal sulit, lalu 46 responden (20%) memberi nilai 3 untuk soal sedang, dan 61 responden (26%) memberikan nilai 6 untuk soal mudah.

Bila diperhatikan, tingkat kesulitan yang tinggi mempunyai sedikit angka (1 poin). Lalu, Soal dengan tingkat kesulitan Sedang memiliki beberapa angka yang diberikan (5 poin), dan tingkat kesulitan Mudah memiliki angka lebih banyak (6 poin). Sehingga dapat disimpulkan bahwa jumlah variabel dengan angka yang diberikan berbanding terbalik dengan tingkat kesulitan.

d. Hubungan faktor bahasa dengan kesulitan soal.

Dari hasil penyebaran kuesioner diperoleh persepsi responden terhadap soal-soal yang telah diberikan dan hubungan faktor bahasa dengan kesulitan soal, dapat dilihat dari tabel berikut:

Tabel 5: Distribusi frekuensi hubungan pemakaian bahasa dengan tingkat kesulitan soal

Nilai	Sulit	Sedang	Mudah
1	24	6	4
2	12	2	6
3	42	8	10
4	46	18	14
5	39	24	32
6	37	42	23
7	28	62	43
8	8	49	43
9	4	29	65

Tabel di atas merupakan tabel distribusi frekuensi berdasar hasil survei, kami menggunakan modus data untuk menentukan hubungan antara penilaian bahasa dan tingkat kesulitan.

Dalam Tabel 5 terdapat tiga jenis soal yang diberikan, dalam hubungan dengan struktur bahasa, ditemukan 46 responden (20%) memberikan nilai 4 pada soal sulit, lalu 62 responden (26%) memberi nilai 7 untuk soal sedang, dan 65 responden (27%) memberikan nilai 9 untuk soal mudah.

Menurut survei tersebut, ditemukan bahwa tingkat kesulitan yang tinggi mempunyai pengantar yang sedikit (4 poin). Lalu, Soal dengan tingkat kesulitan Sedang memiliki bahasa pengantar yang cukup (7 poin), dan tingkat kesulitan Mudah memiliki bahasa pengantar yang mudah dan banyak (9 poin). Sehingga dapat

disimpulkan bahwa faktor bahasa pengantar berbanding terbalik dengan tingkat kesulitan.

2. Kecepatan Pembuatan Soal

Dari hasil uji pada pembuatan soal, dihitung berapa kecepatan pembuatan soal. Dihitung berapa mili detik yang dibutuhkan untuk membuat 1 soal. Dari proses yang dijalankan selama 100 detik. Sampel kecepatan pembuatan ditampilkan seperti pada tabel di bawah ini:

Tabel 6: Kecepatan pembuatan soal (ms/soal) dengan waktu kejadian(detik).

Waktu	Sulit	Sedang	Mudah
10	50	45	22
20	65	40	26
30	53	46	17
40	59	35	26
50	67	41	25
60	55	43	14
70	58	33	29
80	63	34	29
90	57	45	17
100	60	49	28

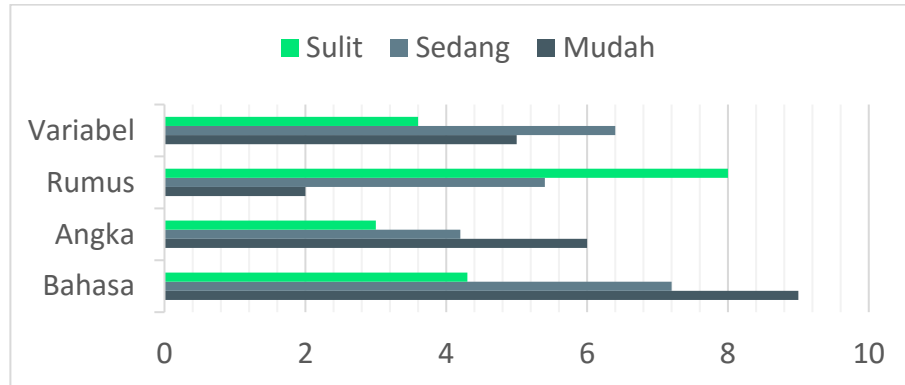
Tabel di atas merupakan sampel kecepatan pembuatan soal, diambil nilai rata-rata tiap 10 detik, misalkan pada detik ke 10, kecepatan untuk soal sulit adalah 50 ms per soal. Atau bila dikonversikan menjadi 20 soal per detik.

Berdasar Tabel 6 di atas, kecepatan pembuatan untuk jenis soal sulit bervariasi dengan rata-rata 58.7 ms/soal, kemudian untuk jenis soal sedang berkisar dengan rata-rata 41.1 ms/soal, lalu untuk jenis soal mudah memiliki kecepatan rata-rata 23.3 ms/soal.

C. Hasil

Hasil penelitian terhadap variabel yang diteliti dalam penelitian ini akan diberikan kesimpulan dengan menentukan skor interval kelas terlebih dahulu pada variabel penelitian

1. Tingkat kesulitan soal



Gambar 7: Diagram rekapitulasi variabel pembentuk kesulitan soal.

Berdasarkan diagram di atas, dapat dibentuk rumus pembuatan soal berdasarkan masukan jumlah variabel, banyaknya rumus, angka dan bahasa yang digunakan. Sehingga tingkat kesulitan soal dapat ditentukan, seperti pada rumus berikut:

$$Tk = \frac{R}{V + A + B}$$

Penjelasan:

Tk	: Tingkat Kesulitan Soal	A	: Bentuk angka
R	: Rumus yang digunakan	B	: Pola kerumitan bahasa.
V	: Variabel yang diberikan		

2. Kecepatan pembuatan soal

Berdasar analisis di atas, kecepatan pembuatan untuk jenis soal sulit bervariasi dengan rata-rata 58.7 ms/soal, kemudian untuk jenis soal sedang berkisar dengan rata-rata 41.1 ms/soal, lalu untuk jenis soal m memiliki kecepatan rata-rata 23.3 ms/soal.

Dapat disimpulkan semakin tinggi tingkat kesulitan soal yang akan dibuat, maka semakin banyak variabel yang harus dibuat, dan semakin banyak waktu yang diperlukan untuk membuat 1 soal.

Jika kecepatan pengacakan tiap variabel:

$$v(x) = n(x) * 1 \text{ ms}$$

Maka kecepatan total:

$$\sum \frac{v(R)}{v(V) + v(A) + v(B)}$$

BAB V

SIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan analisis data, dapat disimpulkan bahwa, aplikasi ini dapat membuat soal dengan tingkat kesulitan yang sama, didapat pula rumus yang menentukan tingkat kesulitan soal secara sederhana.

Kemudian, aplikasi juga mampu membuat soal dengan kecepatan cukup tinggi, sehingga memungkinkan pembuatan paket soal dalam skala besar, dengan waktu yang singkat.

B. Saran

Berdasar pada penelitian yang sudah kami lakukan, kami ingin mengemukakan saran sebagai berikut:

1. Dibutuhkan penelitian lanjutan, penyempurnaan program aplikasi dan sistematika, maupun algoritma yang lebih baik untuk membuat soal lebih cepat dan bervariasi.
2. Untuk pendidik, dengan menggunakan rumus yang ditemukan diharap dapat digunakan untuk membuat soal di kemudian hari, secara manual maupun menggunakan generator ini. Pengembangan rumus juga perlu dilakukan.
3. Penelitian ini pastinya memiliki keterbatasan, karena itu penelitian sejenis lebih lanjut perlu dilakukan, dengan meningkatkan kualitas metodologi, sehingga dapat ditemukan hasil optimal dan dapat digeneralisasikan.

DAFTAR PUSTAKA

- Alwi, H. (2002). *Kamus Besar Bahasa Indonesia* (3rd ed.). Jakarta: Balai Pustaka.
- Bartee, T. C. (1988). *Expert Systems and Artificial Intelligence: Applications and Management* (1st ed.). Indianapolis: Howard W. Sams & Co.
- Chomsky, N. (1957). *Syntactic Structures*. (1st ed.). Gravenhage: Mouton & Co.
- Clarke, S. (2003). *Enriching Feedback in The Primary Classroom*. London : Hodder.
- Corporation, R. (1955). *A Million Random Digits with 100,000 Normal Deviates*. New York: Free Press.
- Hamalik, O. (1989). *Teknik Pengukuran dan Evaluasi Pendidikan*. Bandung : Mandar Maju.
- Knuth, D. E. (1997). *The Art of Computer Programming,. Seminumerical Algorithms (3rd edition)*.
- McCarthy, J. (1989). *Artificial Intelligence, Logic and Formalizing Common Sense," in Philosophical Logic and Artificial Intelligence*. (R. Thomason , Penyunt.) Dordrecht: Kluwer Academic.
- Neumann, J. v. (1951). Various Technique Used in Connection with Random Digits. (G. E. Forsythe, Penyunt.) *Applied Math Series*, 36–38.
- Qualifications and Curriculum Authority. (2003). *Assessment for Learning : Using Assessment to Raise Achievement in Mathematics*. London: Great Britain.
- Sedgewick, R., & Wayne, K. (2011). *Algorithm* (4th ed.). Boston: Princeton University.

LAMPIRAN

Data Kecepatan (soal/ms)

[illegible]