

PEMBELAJARAN BAHASA INGGRIS DENGAN ANALISIS GRAMMAR

Rionaldo Putra¹⁾, Indah Fitri Astuti²⁾, Awang Harsa K³⁾

^{1,2,3)}Program Studi Ilmu Komputer, FMIPA, Universitas Mulawarman
Email : rionaldomorrell@gmail.com¹⁾, indahfitriastuti@yahoo.com²⁾, awangkid@gmail.com³⁾

ABSTRAK

Bahasa Inggris merupakan bahasa yang sangat penting dan paling banyak digunakan dalam pergaulan Internasional. Kebutuhan penguasaan Bahasa Inggris yang baik sangat diperlukan. *Grammar* merupakan Tata Bahasa dalam Bahasa Inggris. Salah satu hal yang sangat mungkin dirasakan rumit dalam memahami *Grammar* adalah *tenses*. *Tenses* dalam bahasa Inggris berarti 'waktu'. Dalam pembahasan *Grammar*, *Tenses* digunakan untuk membicarakan perbedaan atau perubahan bentuk dalam kalimat yang berhubungan dengan waktu terjadinya peristiwa yang diungkapkan pada suatu kalimat. Oleh karena itu pembelajaran *tenses* sangat penting dalam mengidentifikasi kalimat dalam suatu pembicaraan. Penelitian ini bertujuan untuk membangun suatu sistem yang dapat melakukan analisa terhadap suatu komponen masukkan pada *tenses* tertentu untuk kemudian dilakukan perbaikan secara otomatis. Dengan menerapkan Metode *Rule Base* untuk menentukan kriteria pembentukan dari masing-masing *tenses* kemudian dilakukan perubahan terhadap komponen-komponen tertentu pada *tenses* tersebut secara otomatis. Dari penelitian ini dihasilkan Program Pembelajaran Bahasa Inggris Untuk Level *intermediate*. Penggunaan program ini cukup mudah. *User* hanya perlu mengisi subjek, predikat dan objek untuk masing masing *tenses* yang kemudian akan diproses oleh sistem untuk dilakukan pembenaran serta penjelasan. Pengguna juga dapat melatih kemampuan mereka setelah mempelajari *tenses* menggunakan sistem ini dengan memilih fitur evaluasi untuk menjawab soal soal yang diberikan oleh sistem.

Kata Kunci : Sistem Pembelajaran Bahasa Inggris, Analisis *Grammar*, *Tenses*, *Rule Base*.

PENDAHULUAN

Bahasa Inggris merupakan bahasa yang sangat penting dan terbanyak digunakan dalam pergaulan Internasional. Setelah era globalisasi dimana batas-batas negara semakin luas dan semakin terbukanya pergaulan antar manusia dari belahan bumi menjadikan kebutuhan akan penguasaan Bahasa Inggris yang baik menjadi sangat diperlukan. *Grammar* merupakan Tata Bahasa dalam Bahasa Inggris. Salah satu hal yang sangat mungkin dirasakan rumit dalam memahami *Grammar* adalah *tenses*. *Tenses* dalam bahasa Inggris berarti 'waktu'. Dalam pembahasan *Grammar*, *tenses* digunakan untuk membicarakan perbedaan atau perubahan bentuk kalimat yang berhubungan dengan waktu terjadinya peristiwa yang diungkapkan pada kalimat tersebut. Sebagai contoh, suatu peristiwa yang sama akan diungkapkan secara berbeda apabila waktu terjadinya berbeda. Cara mengungkapkan peristiwa yang terjadi sekarang akan berbeda dengan yang terjadi di masa lalu, dan akan berbeda lagi dengan peristiwa yang terjadi di masa yang akan datang. Oleh karena itulah pembelajaran mengenai *tenses* sangat penting dalam mengidentifikasi kalimat dalam suatu pembicaraan.

Dalam prakteknya, pembelajaran mengenai *tenses* sangat sulit untuk dipahami dikarenakan masing-masing *tenses* memiliki pola dan kegunaan yang berbeda-beda atau bahkan hampir mirip satu dengan lainnya, sehingga dibutuhkan perhatian

ekstra untuk dapat memahami dan mengaplikasikan suatu *tenses* ke dalam situasi yang berbeda-beda pula.

Banyak media pembelajaran yang dapat digunakan seperti buku, *compact disc* (CD), internet dan lain-lain. Komputer *Desktop* menyediakan *resource* yang cukup untuk membangun suatu aplikasi pembelajaran *tenses*. Oleh karena itu, penulis terdorong untuk merancang suatu sistem pembelajaran Bahasa Inggris sederhana. *Level Intermediate* merupakan suatu tingkat kemahiran dalam bahasa Inggris yang didasarkan dari pemahaman dasar dan komunikasi yang baik, namun masih memiliki ruang untuk berkembang di keseluruhan area. *Level Intermediate* dapat berbicara dan memberikan opini mengenai banyak hal dengan menggunakan bentuk *simple* dari masing – masing *tenses*. Dapat memulai percakapan menggunakan bahasa Inggris dalam menjalankan suatu aktivitas seperti berbelanja, bepergian atau reservasi kamar hotel. Pemahaman dalam *grammar* mencakup *adjectives*, *adverbs*, *comparatives*, *articles*, *possessives*, dan *basic prepositions*.

METODE PENELITIAN

a. *Tenses*

Dalam bahasa - bahasa barat, termasuk bahasa Inggris dan beberapa bahasa Asia dan Afrika, selalu dikenal istilah *tenses*. Meskipun kadang di tempat

yang berbeda dipakai istilah yang berbeda pula, maksud dan makna sebenarnya adalah sama.

Istilah *Tenses* dalam bahasa Inggris berarti "waktu". Dalam pembahasan tata bahasa atau gramatika, jika membicarakan *tenses* akan berarti membicarakan perbedaan - perbedaan atau perubahan bentuk kalimat sehubungan dengan waktu terjadinya peristiwa yang diungkapkan oleh kalimat - kalimat. Sebagai contoh, suatu peristiwa atau tindakan yang sama akan diungkapkan secara berbeda apabila waktu terjadinya berbeda. Cara mengungkapkan peristiwa yang terjadi sekarang akan berbeda dengan yang terjadi di masa lampau, dan akan berbeda lagi dengan yang terjadi di masa yang akan datang. Lihatlah ketiga kalimat contoh ini:

I am reading a novel now (=sekarang),

I read a novel yesterday (=kemarin),

I will read a novel tomorrow (=besok).

Perhatikan bahwa yang berbeda adalah bentuk kata kerja pada predikat kalimatnya. Tanda waktu 'sekarang' (kini), 'kemarin' (lampau), dan 'besok' (yang akan datang) menunjuk pada waktu yang berbeda. Peristiwa - peristiwa kini, lampau, dan yang akan datang itu masing-masing akan diwarnai dengan 'biasa', 'sedang', dan 'selesai'. Dalam bahasa Inggris, semua itu dikombinasikan sehingga membentuk beberapa gabungan dan yang lazim digunakan adalah 12 (dua belas) yang biasa disebut *tenses*.

b. Rekayasa Perangkat Lunak

Secara umum rekayasa perangkat lunak memiliki pengertian sebagai sebuah disiplin ilmu yang menerapkan teori-teori dasar bidang tertentu, ke dalam sebuah perangkat lunak yang digunakan sebagai alat bantu pemecahan masalah atau proses pengambil keputusan.

Menurut Stephen R. Schach dalam buku *Rekayasa Perangkat Lunak Pendekatan Praktis* menuliskan bahwa "rekayasa perangkat lunak adalah sebuah disiplin dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tepat waktu serta memuaskan keinginan pemakai".

Menurut Fritz Bauer dalam buku *Rekayasa Perangkat Lunak Pendekatan Praktis* menuliskan bahwa "Rekayasa perangkat lunak adalah penetapan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata."

Menurut IEEE 610.12 dalam buku *Rekayasa Perangkat Lunak Pendekatan Praktis* menuliskan bahwa "Rekayasa perangkat lunak adalah sebuah studi pendekatan dan aplikasi secara sistematis, disiplin pengembangan operasi dan pemeliharaan perangkat lunak yang kesemuanya itu merupakan aplikasi rekayasa yang berkaitan dengan perangkat lunak".

Dari ketiga pengertian tersebut, arti yang diberikan Fritz Bauer sangat sesuai dengan tujuan dan sasaran dari rekayasa perangkat lunak. Pernyataan Stephen R. Schach terbatas pada menanggulangi kekurangan yang terjadi jika tidak menerapkan rekayasa perangkat lunak. Pengertian yang diberikan IEEE 610.12 paling baik dalam menyampaikan wujud dari rekayasa perangkat lunak.

Rekayasa perangkat lunak merupakan ilmu yang penting untuk diperdalam karena teknologi ini memberikan stabilitas, kontrol, dan organisasi aktifitas yang jika tidak terkontrol dengan baik dapat menjadi sangat kacau. Ada beberapa tujuan dilakukan rekayasa perangkat lunak antara lain yaitu :

Untuk membangun *software* yang benar dan benar sebuah *software* (*Right Software and Software Right*).

Untuk membangun *software* yang tepat (*correct*).

Dikelola dengan baik untuk pemeliharaan kebenarannya (*correctness*).

PENELITIAN DAN PEMBAHASAN

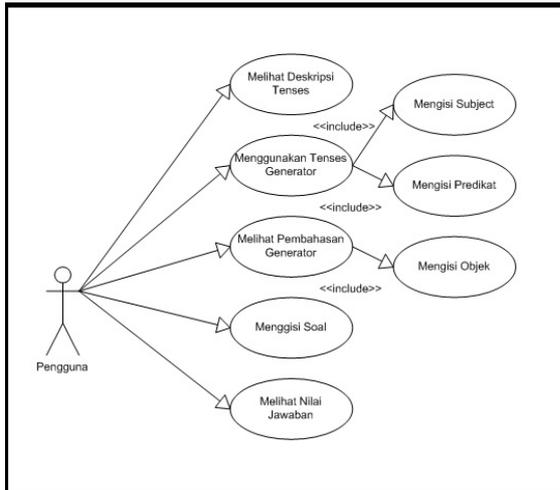
Sistem Pembelajaran Bahasa Inggris Dengan Analisis *Grammar* merupakan program pembelajaran yang membahas mengenai *tenses* pada Bahasa Inggris dan diperuntukkan bagi pengguna yang memiliki tingkat pemahaman Bahasa Inggris di *level intermediate*.

Program ini dibangun menggunakan Microsoft Visual Studion. NET dan Microsoft Access sebagai basis data. *Tenses* yang dibahas adalah :

- a) *Present Perfect Tense*
- b) *Present Perfect Continuous Tense*
- c) *Past perfect Tense*
- d) *Past perfect Continuous Tense*
- e) *Future Perfect Tense*
- f) *Future Continuous Tense*

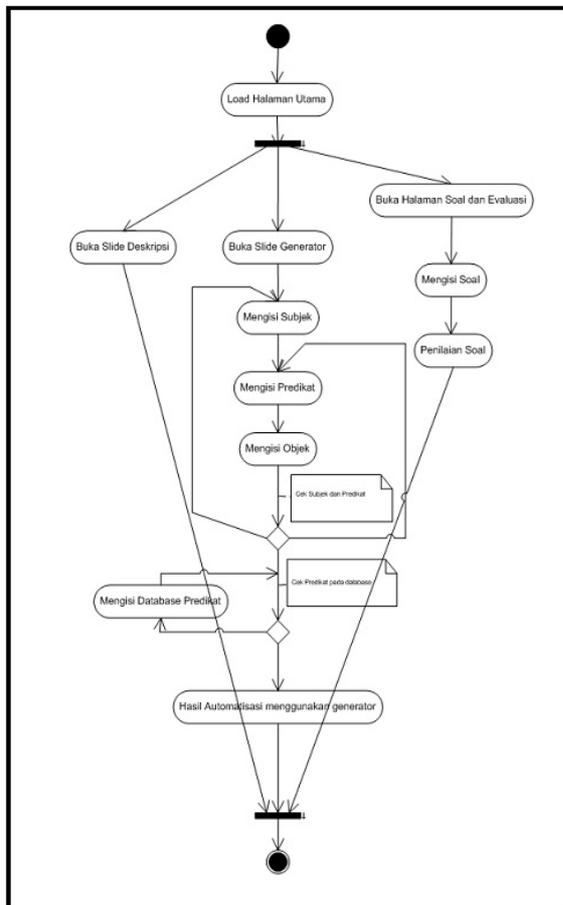
Pada program ini terdapat fitur *Tenses Generator* yang dapat meng-automatisasi masing-masing *tenses* berdasarkan masukan dalam bentuk kata atau kalimat utuh yang sesuai dengan bentuk masing - masing *tenses*. Program ini memiliki fitur 'soal dan evaluasi' sebagai bahan evaluasi dari pembelajaran yang dilakukan sehingga diharapkan meningkatkan pemahaman pengguna dalam mempelajari *tenses*.

Use case diagram merupakan tahap awal dan utama dalam proses pengembangan sistem, dimana dalam tahapan ini dijelaskan dan didefinisikan fungsi serta fitur apa saja yang dapat disediakan oleh sistem. Pada use case diagram mempunyai satu aktor yaitu pengguna. Agar lebih jelas bisa dilihat pada gambar 4.1.



Gambar 4.1 Use Case Diagram Pembelajaran Bahasa Inggris Dengan Analisis Grammar

Activity Diagram menjelaskan mengenai alur-alur kegiatan yang dapat dilakukan oleh pengguna terhadap sistem yang telah dikembangkan, berdasarkan use case diagram yang telah dibuat sebelumnya. Pada diagram ini membahas mengenai aktivitas pengguna bisa dilihat pada gambar 4.2.



Gambar 4.2 Activity Diagram Pembelajaran Bahasa Inggris Dengan Analisis Grammar

IMPLEMENTASI SISTEM

Metode Rule Base digunakan untuk menentukan aturan yang disesuaikan dengan grammar / tata bahasa dari masing masing tenses, diantaranya :

- a) Jenis subjek apakah subjek tersebut plural atau singular.
- b) Pembentukan verb / kata kerja atau komponen masing masing tenses yang dapat dihubungkan dengan jenis subjek.

Berikut adalah gambar penerapan metode Rule Base pada beberapa aspek di dalam program ini :

```
Function isPlural(ByVal subject As String)
Dim threeLastChar As String
Dim twoLastChar As String
If subject.Length >= 3 Then
threeLastChar = subject.Substring(subject.Length - 3, 3)
twoLastChar = subject.Substring(subject.Length - 2, 2)
Else
threeLastChar = ""
twoLastChar = ""
End If

Dim oneLastChar As String = subject.Substring(subject.Length - 1, 1)
If subject.Contains(" and ") Then
Return True
ElseIf subject.Contains(" or ") Then
Return False
Else
If threeLastChar = "ies" Then
Return True
ElseIf twoLastChar = "es" Then
Return True
ElseIf oneLastChar = "s" And twoLastChar <> "ss" Then
Dim charfronts As String = subject.Substring(subject.Length - 2, 1)
If oneLastChar = "s" And Not IsVocal(charfronts) Then
Return True
Else
Return False
End If
End If

If Plural_NounTableAdapter.Singular(subject).Rows.Count > 0 Then
Return False
ElseIf Plural_NounTableAdapter.Public Overridable Overloads Function Singular(Singular1 As String) As String
Return True
Else
Return False
End If
End If
End Function
```

Gambar 4.3. Metode Rule Base untuk menentukan subjek tunggal atau jamak

Gambar 4.3 merupakan implementasi metode Rule Base untuk menentukan subjek tunggal atau jamak dengan berbagai kriteria, mulai dari apakah subjek menggunakan konjungsi/kata hubung 'and' atau 'or' atau jumlah karakter mulai dari tiga karakter dibelakang subjek sampai 1 karakter dibelakang subjek hingga satu karakter dibelakang subjek apakah mengandung '-ies', '-es' atau '-s' yang kemudian akan dianggap sebagai subjek jamak, apabila kata kata tersebut tidak mengandung unsur diatas maka akan dianggap subjek yang dimasukan adalah subjek tunggal.

Gambar 4.4. merupakan implementasi metode Rule Base pada salah satu bentuk tenses yaitu Present Perfect Tenses. Yang menjabarkan aturan pembentukan tenses itu sendiri.

```
Dim sng As String
Select Case sng Like "it:te"
Case "Present Perfect Tense"
Select Case subject
Case "he", "she", "it", "you"
AddResult(subject yang digunakan adalah " + subject + ", maka have yang merupakan koponen perfect tense mengalami konjungsi sesuai subject yaitu "have")
Case "we", "they", "I"
AddResult(subject yang digunakan adalah " + subject + ", maka have yang merupakan koponen perfect tense mengalami konjungsi sesuai subject yaitu "has")
Case Else
If IsPlural(subject) Then
AddResult(subject yang digunakan adalah " + subject + " termasuk bentuk [jamak/plural], maka have yang merupakan koponen perfect tense mengalami konjungsi
Else
AddResult(subject yang digunakan adalah " + subject + " termasuk bentuk [tunggal/singular], maka have yang merupakan koponen perfect tense mengalami konj
End If
End Select

Dim v3 As String = web_ListTableAdapter.get3(tstPredicate.Text, tstPredicate.Text, tstObject.Text)
If v3 < "" Then
AddResult.Text = subject & " " & getwordmas(subject) & " " & v3 & " " & tstObject.Text
Else
AddResult("No Match for Predicate on Database. Please Insert to Database ", HighObjIn.Exclamation, "Warning")
Form1.ShowMsgBox()
AddResult.Text = subject & " " & getwordmas(subject) & " " & v3 & " " & tstObject.Text
Exit Sub
End If

AddResult("Present Perfect Tense menggunakan kata kerja bentuk ke 3, sehingga " + tstPredicate.Text + " mengalami perubahan bentuk menjadi " + v3)
AddResult("Tidak dalam Present Perfect Tense dapat menggunakan konjungsi seperti for, since atau before")
AddResult("Contoh konjungsi = 'She has been there since 9 o'clock'")
```

Gambar 4.4. Metode Rule Base pada Present Perfect Tense Generator

```

Function AddSuffixIng(predicate As String) As String
    predicate = LCase(predicate)
    Dim twoLastChar As String = predicate.Substring(predicate.Length - 2, 2)
    Dim oneLastChar As String = predicate.Substring(predicate.Length - 1, 1)

    If predicate = "go" Or predicate = "be" Or predicate = "do" Then
        Return predicate & "ing"
    Else
        If twoLastChar(0) = twoLastChar(1) And IsVocal(twoLastChar(0)) And IsVocal(twoLastChar(1)) Then
            Return predicate & "ing"
        Else
            If twoLastChar = "ie" Then
                Return predicate.Substring(0, predicate.Length - 2) & "ying"
            Else
                If IsVocal(oneLastChar) Then
                    Return predicate.Substring(0, predicate.Length - 1) & "ing"
                Else
                    Return predicate & "ing"
                End If
            End If
        End If
    End If
End Function
    
```

Gambar 4.5. Metode *Rule Base* untuk menentukan gerund / *Verb-ing*

Gambar 4.5. merupakan implementasi metode *Rule Base* pada program untuk menentukan perubahan bentuk *verb* menjadi *verb-ing / gerund*. Berdasarkan kriteria dua hingga satu karakter dibelakang *verb*

```

Function getHaveOrHas(ByVal subject As String) As String
    Select Case subject
        Case "We", "They", "I", "You"
            Return "have"
        Case Else
            If IsPlural(subject) Then
                Return "have"
            Else
                Return "has"
            End If
    End Select
End Function
    
```

Gambar 4.6. Metode *Rule Base* untuk menentukan penggunaan *Have / Has* berdasarkan subjek

Gambar 4.6. merupakan implementasi metode *Rule Base* untuk menentukan penggunaan kata *Have* atau *Has* yang berarti sudah dalam tenses tertentu berdasarkan jenis subjek, apakah subjeknya tunggal atau jamak. Dimana *Have* digunakan untuk subjek jamak dan *Has* digunakan untuk subjek tunggal

Pada program ini pengguna langsung melihat tampilan program secara keseluruhan dan dapat langsung berinteraksi dengan fitur – fitur yang disediakan. Pengguna dapat memilih untuk mengklik tombol *tenses* yang ada dibagian kiri program untuk melihat penjelasan dari masing – masing *tenses* atau mengerjakan soal dengan mengklik tombol *Soal dan Evaluasi*. Tombol *Back* dan *Next* dapat digunakan untuk berpindah slide dari slide satu ke slide lainnya. Tampilan utama dari program ini bisa dilihat pada Gambar 4.7.



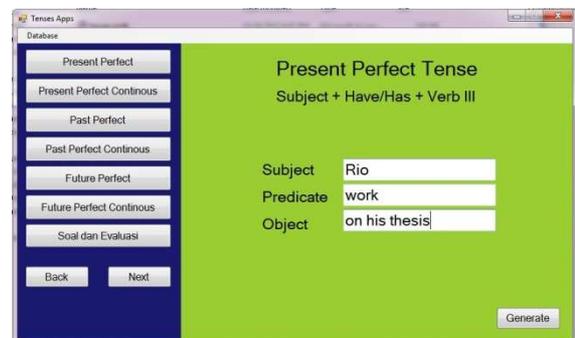
Gambar 4.7. Tampilan utama program pembelajaran *tenses*

Dengan mengklik tombol – tombol yang berada pada Panel Menu di sisi kiri program pengguna dapat mengakses penjelasan dari masing – masing *tenses* yang berada pada Panel 2 disebelah kanan dengan tampilan sliding layaknya presentasi menggunakan Microsoft Power Point. Contoh, jika pengguna mengklik tombol *Present Perfect* pada panel 1 maka akan muncul penjelasan mengenai *Present Perfect Tense* pada panel 2 seperti pada Gambar 4.8.



Gambar 4.8 Tampilan penjelasan *Present Perfect Tense* pada Panel 2

Jika kita mengklik tombol *Next* maka akan muncul *Tenses Generator* untuk *Present Perfect Tense* pada panel 2 , seperti pada gambar 4.9.



Gambar 4.9. Tampilan *Tenses Generator* untuk *Present Perfect Tense*

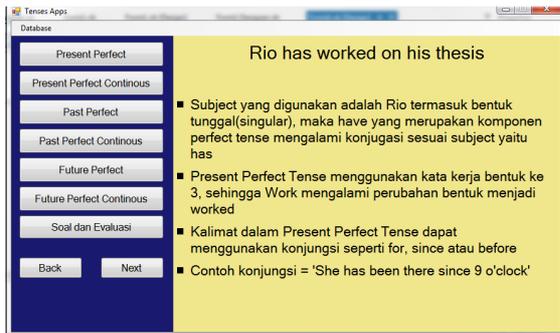
Tenses Generator ini akan melakukan otomatisasi pada kata – kata atau frase yang

dimasukkan ke dalam *generator* menjadi sebuah kalimat utuh berdasarkan pola pembentukan kalimat seperti yang ditampilkan di atas. Ketika pengguna mengklik tombol *Generate* maka generator akan menampilkan hasil otomatisasi ke slide selanjutnya yang berada pada Panel 3, namun apabila Predikat yang diinputkan tidak terdapat di dalam *database* maka pengguna akan diminta untuk memasukkan terlebih dahulu predikat tersebut kedalam *database*. seperti pada gambar 4.10.



Gambar 4.10. Tampilan program ketika memasukkan predikat kedalam *database*.

Setelah berhasil memasukkan predikat kedalam *database* maka program secara otomatis meng-*generate* berdasarkan masukkan sebelumnya dan menampilkan hasil otomatisasi ke slide selanjutnya beserta pembahasan bagaimana kalimat dalam *tense* tersebut terbentuk. Dapat dilihat pada gambar 4.11.

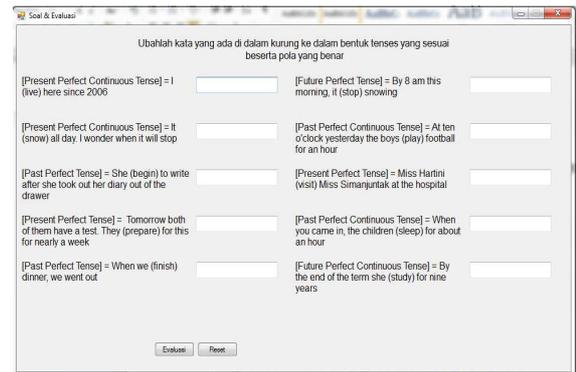


Gambar 4.11. Tampilan *Result* dari *Present Perfect Tense* beserta pembahasan

Ketika pengguna ingin mempelajari *tense* yang lain maka pengguna hanya cukup menekan tombol *tenses* lainnya untuk menampilkan pembahasan *tenses* yang sesuai dengan pilihan. Kemudian jika pengguna merasa sudah cukup paham dengan penjelasan masing- masing *tenses* beserta pembahasan mengenai bagaiman suatu kalimat dalam *tenses* tersebut terbentuk pengguna dapat mencoba untuk menyelesaikan soal yang ada pada program ini dengan mengklik tombol Soal dan Evaluasi yang terdapat pada panel menu. Bentuk soal yang digunakan adalah esai dengan mengganti

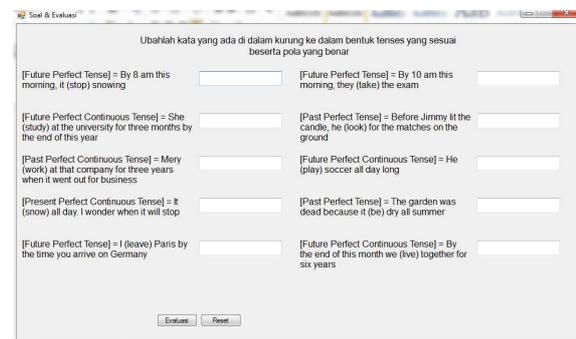
kata yang berada didalam kurung dengan perubahan bentuk yang sesuai dengan *tenses* yang diperintahkan beserta pola dasar kalimatnya. Soal dan Jawaban diambil dari *database* yang telah dimasukkan sebelumnya Soal yang ditampilkan akan selalu berubah setiap kali fitur Evaluasi ditutup dan ketika fitur Evaluasi digunakan maka windows program utama akan ditutup untuk mencegah user menggunakan generator dalam mengerjakan soal.

Gambar 4.12. menunjukkan fitur Soal dan Evaluasi ketika dibuka



Gambar 4.12. Tampilan Fitur Evaluasi ketika dibuka pertama kali

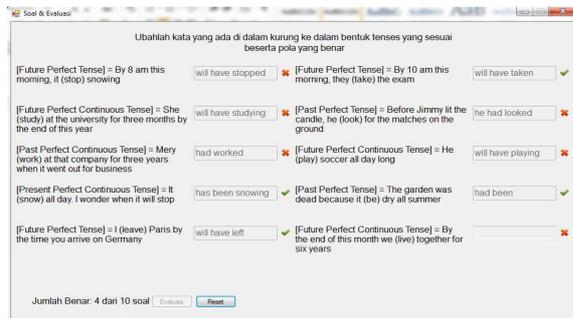
Gambar 4.12. menampilkan 10 soal dari 60 set soal yang dimasukkan kedalam *database* sebelumnya. Sistem akan menilai skor pengguna ketika pengguna menekan tombol 'Evaluasi'. Tombol 'Reset' berfungsi untuk mengosongkan kembali kolom isian sehingga pengguna dapat melakukan pengisian ulang terhadap soal-soal yang ditampilkan. Apabila pengguna memilih untuk menggunakan fitur Evaluasi kembali maka soal yang diberikan akan teracak secara otomatis, seperti pada gambar 4.13.



Gambar 4.13. Tampilan fitur evaluasi setelah dibuka untuk yang kedua kalinya

Gambar 4.14. adalah tampilan fitur evaluasi yang telah diisi, untuk melihat skor yang diraih ketika selesai mengerjakan soal pengguna hanya cukup mengklik tombol 'Evaluasi'. Pengguna akan mengetahui berapa jumlah benar dari semua soal

yang dikerjakan dan akan tampil ikon-ikon yang menunjukkan soal-soal yang dijawab benar atau soal-soal yang dijawab salah



Gambar 4.14. Tampilan fitur evaluasi dengan isian dan skor

KESIMPULAN

Dari hasil penelitian yang telah dilakukan oleh penulis, maka dapat diambil beberapa kesimpulan antara lain:

a. Membangun Program Pembelajaran Tenses dengan Analisis Grammar yang dapat digunakan untuk mempelajari bagaimana suatu kalimat pada tense tertentu dapat terbentuk dengan fitur *Tenses Generator*.

b. Program ini dapat menampilkan suatu kalimat berdasarkan inputan yang dimasukkan pada fitur *Tense Generator* dan juga dapat digunakan untuk mengasah kemampuan pengguna dengan fitur 'Soal dan Evaluasi', sehingga diharapkan kemampuan pengguna dalam memahami Tenses meningkat.

DAFTAR PUSTAKA

- [1] Andi. (2005). Menggunakan Microsoft Office Access. Semarang: Wahana Komputer.
- [2] Fowler, M. (2004). UML Distilled Edisi 3 Panduan Singkat Bahasa Permodelan Objek Standar. Yogyakarta: ANDI.
- [3] Martinus, S. (2008). Cara Mudah Belajar dan Memahami Tenses. Yogyakarta: IndonesiaTera.
- [4] Murphy, R. (1998). Grammar In Use Reference And Practice For Intermediate Students Of English. New York: Cambridge University.
- [5] S.Pressman, R. (2012). Rekayasa Perangkat Lunak Pendekatan Praktis Edisi 7 Buku 1. Yogyakarta: ANDI.
- [6] Sommerville, I. (2003). Software Engineering Rekayasa Perangkat Lunak Jilid 1. Jakarta: Erlangga.
- [7] Putra, Rionaldo. (2014). "Pembelajaran Bahasa Inggris Dengan Analisis Grammar". Skripsi Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Mulawarman