

# PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PEMBANGKIT STRUCTURED ENGLISH BERDASARKAN TABEL KEPUTUSAN

**Aris Tjahyanto, Sujianik**

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5913804  
Email: arist@its-sby.edu

## ABSTRAK

*Dalam pengembangan sebuah sistem informasi, pengguna seringkali memberikan beberapa aturan bisnis untuk diterapkan ke dalam sebuah sistem informasi. Aturan tersebut biasanya berupa aturan tertulis atau pun tidak tertulis yang menjadi kebijakan instansi. Pada tahap analisis, seorang sistem analis perlu mendokumentasikan aturan bisnis untuk keperluan perencanaan lebih lanjut. Dokumentasi bisa diwujudkan sebagai sebuah tabel keputusan yang bisa dengan mudah untuk menemukan adanya redundansi atau pun kontradiksi aturan. Selain itu, dengan tabel keputusan akan memudahkan programmer dalam mengimplementasikan aturan bisnis ke dalam program komputer.*

*Pada penelitian ini dikembangkan sebuah perangkat lunak yang memiliki kemampuan untuk membangkitkan pseudocode atau Structured English dari aturan bisnis yang telah diubah menjadi tabel keputusan. Sebelum aturan bisnis tersebut dimasukkan ke dalam tabel keputusan, sistem analis harus terlebih dulu menentukan condition dan action yang sesuai. Selain itu, perangkat lunak ini diharapkan juga mampu membuat, memvalidasi dan menyederhanakan sebuah tabel keputusan secara otomatis.*

*Berdasar uji coba yang dilakukan, terbukti perangkat lunak dapat bekerja dengan baik sesuai fungsi yang diharapkan. Pembuatan tabel keputusan dapat dilakukan baik untuk limited maupun extended entry decision table. Perangkat lunak juga mampu melakukan pengecekan validasi, penghapusan rule tabel keputusan, dan juga otomatisasi penyederhanaan tabel keputusan. Selain itu perangkat lunak berhasil membangkitkan pseudocode yang berbentuk Structured English dari tabel keputusan.*

**Kata kunci :** *Tabel Keputusan, Decision Table, Structured English, Aturan Bisnis, Business Rule, Analis Sistem, Analisis dan Desain Sistem Informasi.*

## 1. PENDAHULUAN

Proses pengembangan sistem informasi melewati beberapa tahapan mulai dari perencanaan sampai dengan implementasi sistem dan pemeliharaan sistem. Salah satu tahapan yang utama adalah analisis dan perancangan sistem. Dalam tahap analisis, seorang analis sistem menganalisis kebutuhan sistem dan juga menganalisis keputusan terstruktur sesuai dengan keinginan user. Dan dalam mendefinisikan kebutuhannya, user biasanya memasukkan aturan-aturan yang kompleks dimana aturan tersebut membutuhkan analisis keputusan untuk pemecahannya.

Dalam melakukan analisis keputusan, seorang analis sistem melibatkan keputusan-keputusan terstruktur. Untuk menganalisis keputusan-keputusan terstruktur tersebut, biasanya analis sistem menggunakan tiga metode yaitu Bahasa Inggris Terstruktur (Structured English), Tabel Keputusan (Decision Table) dan Pohon Keputusan (Decision Tree).

Sebuah aturan bisnis dapat diubah menjadi bentuk tabel keputusan oleh analis sistem melalui beberapa tahapan. Tetapi bentuk tabel keputusan ini tidak selalu bisa dipahami oleh programmer. Pada dasarnya seorang programmer akan lebih mudah

memahami suatu aturan jika dalam bentuk algoritma atau pseudocode. Adanya ambiguitas dapat terjadi sehingga mempersulit penerapan aturan ke dalam sistem yang akan dibuat.

Metode Structured English juga sangat penting karena metode ini merupakan pseudocode atau algoritma yang digunakan dalam membuat sebuah sistem. Dengan adanya Structured English diharapkan pengembangan sistem dapat dilakukan dengan lebih cepat dan programmer dapat mengerti apa yang dikehendaki analis sistem dan user.

Oleh karena itu, jika ada suatu aturan tertentu dari user dimana dari aturan tersebut membutuhkan tabel keputusan dalam menyelesaikannya, maka perlu dikembangkan sebuah perangkat lunak yang mampu membuat tabel keputusan secara otomatis berdasarkan kondisi, alternatif-alternatif kondisi, tindakan serta rule-rule yang telah ditentukan. Dan untuk pengembangan tabel keputusan yang dibuat, maka tabel tersebut diintegrasikan dengan database yang sudah ada. Selain itu, juga perlu dikembangkan tabel keputusan yang dapat membangkitkan sebuah code yang memiliki struktur seperti pada Structured English.

**1.1. TUJUAN**

Tujuan penelitian ini adalah menghasilkan suatu perangkat lunak yang memiliki kemampuan membangkitkan Structured English dari aturan atau rule yang telah didefinisikan menjadi tabel keputusan. Selain itu aplikasi yang dibuat juga mampu membangun, mengisi, memanipulasi serta memvalidasi sebuah tabel keputusan yang telah diintegrasikan dengan database.

**1.2. MANFAAT**

Penelitian ini bermanfaat antara lain:

1. Membantu pengguna dalam melaksanakan aturan bisnis yang dijalankan dalam sistem yang akan dibuat
2. Membantu analis sistem dalam mendokumentasikan aturan bisnis.
3. Menjembatani antara pengguna, analis sistem dan programmer dalam memahami aturan, dan implementasinya.
4. Membantu programmer dalam memahami aturan bisnis melalui Structured English yang dihasilkan dari aplikasi yang dibuat pada penelitian ini.

**1.3. PERMASALAHAN**

Pada penelitian ini, beberapa permasalahan yang akan diselesaikan, yaitu:

- Bagaimana membuat tabel keputusan yang diintegrasikan dengan sebuah sistem basisdata relasional?
- Bagaimana membuat aplikasi tabel keputusan yang dapat mengecek redundansi dan inkonsistensi sebuah rule yang diinputkan?
- Bagaimana membuat suatu pembangkit Structured English dari sebuah tabel keputusan yang telah dibuat?

**2. TABEL KEPUTUSAN**

Tabel keputusan (*Decision Table*) adalah tabel yang digunakan sebagai alat bantu untuk menyelesaikan logika di dalam program. Algoritma yang berisi keputusan bertingkat yang banyak sekali sangat sulit untuk digambarkan langsung dengan Structured English atau pseudocode tetapi dapat diatasi jika terlebih dahulu dengan menggunakan tabel keputusan. Dengan demikian tabel keputusan efektif digunakan jika kondisi yang akan diseleksi didalam program jumlahnya cukup banyak dan rumit [1].

Struktur dari tabel keputusan terdiri dari empat bagian utama yaitu kondisi (*condition stub*), alternatif-alternatif kondisi (*condition entry*), tindakan (*action stub*) dan masukan-masukan tindakan (*action entry*). Condition stub berisi kondisi-kondisi yang akan diseleksi. Condition entry berisi kemungkinan-kemungkinan atau alternatif-alternatif dari kondisi yang diseleksi. Action stub berisi pernyataan-pernyataan yang akan dikerjakan atau

keputusan yang diambil berdasarkan terpenuhi atau tidaknya kondisi-kondisi dalam condition stub. Dan yang terakhir, action entry digunakan untuk memberi tanda tindakan mana yang akan dilakukan dan mana yang tidak dilakukan [1].

Keempat bagian pada tabel keputusan dapat digambarkan dalam bentuk baris dan kolom. Tabel keputusan terbagi menjadi 4 kuadran seperti yang ditunjukkan pada gambar 2.2. Kuadran kiri atas berisikan kondisi-kondisi (*condition stub*), kuadran kanan atas berisikan masukan alternatif-alternatif kondisi (*condition entry*). Separuh bagian bawah tabel tersebut berisikan tindakan-tindakan (*action stub*) yang berada di sebelah kiri dan rule-rule (aturan-aturan) pelaksanaan tindakan atau action entry diletakkan di sebelah kanan. Action stub berisi pernyataan-pernyataan tindakan yang akan dikerjakan baik kondisi yang diseleksi terpenuhi maupun tidak.

**Tabel 1. Struktur tabel keputusan**

Kondisi /Aksi	Rules			
	1	2	...	N
Condition Stub				Condition Entry
Action Stub				Action Entry

Ketika suatu tabel keputusan digunakan untuk menentukan tindakan apa yang perlu dilakukan, logikanya bergerak searah jarum jam mulai dari bagian kiri atas. Dan dalam membuat tabel keputusan adanya serangkain aturan (rule) untuk setiap tindakan sangat diperlukan. Aturan-aturan tersebut merupakan kombinasi dari semua kemungkinan kondisi yang memunculkan suatu tindakan. Dimana sebuah aturan adalah sebuah prosedur untuk pemeriksaan kondisi-kondisi yang berbeda dan pernyataan tindakan memberitahukan tindakan yang harus diambil jika sebuah aturan benar. Tabel keputusan dibaca sampai sebuah keadaan untuk sebuah aturan cocok dan tindakan yang dicantumkan diambil. Kemudian pemrosesan tabel diteruskan.

Tabel keputusan memiliki beberapa keuntungan, antara lain: (a) tabel keputusan mudah dibaca dan dimengerti oleh pengguna, (b) hubungan sebab akibat ditunjukkan oleh tabel keputusan, dengan demikian pengguna lebih mudah melakukan validasi yaitu pengecekan kontradiksi dan redundansi, dan (c) tabel keputusan sangat memungkinkan untuk dilakukan pemeriksaan kelengkapannya (semua kombinasi kondisi telah dipertimbangkan terlebih dahulu).

Ada dua jenis tabel keputusan, yaitu:

- **Limited-entry decision table.** Digunakan untuk tabel keputusan yang sederhana dimana condition entry diisi dengan nilai Y atau N. Jadi setiap kondisi yang diseleksi akan mempunyai 2 kemungkinan kejadian terpenuhi dan tidak terpenuhi. Bila kondisi

ada x kondisi yang akan diseleksi, maka akan terdapat N kemungkinan kejadian, yaitu  $N = 2^x$ .

• **Extended-entry decision table.** Digunakan untuk tabel keputusan yang lebih kompleks dengan condition entry diekspresikan sebagai suatu variable keputusan yang mana variable tersebut nilainya dapat berupa angka atau berupa range tertentu bisa lebih besar, lebih kecil dan lain sebagainya.

**Tabel 2. Contoh limited entry decision table**

Kondisi / Aksi	Rule			
	1	2	3	4
Dibawah \$50	Y	Y	N	N
Membayar dengan cek dengan 2 formulir	Y	N	Y	N
Menggunakan kartu kredit	N	Y	N	Y
Mencatat penjualan	X			
Memeriksa kartu kredit pada buku		X		
Meminta persetujuan supervisor			X	
Meminta otorasi bank atas kredit				X

Pada gambar di bawah ini merupakan sebuah tabel keputusan extended entry untuk jenis kasus permintaan pesanan (order) untuk inventory sebuah perusahaan. Dengan memakai *extended entry*, tabel keputusan yang dihasilkan menjadi lebih sederhana karena jumlah rule yang dihasilkan menjadi lebih sedikit bila dibandingkan dengan tabel keputusan berjenis *limited entry*.

**Tabel 3. Contoh Extended Entry Decision Table**

Kondisi / Aksi	Rule					
	1	2	3	4	5	6
Harga setiap Item	-	A	B	C	D	C
Jumlah Pesanan	P	Q	Q	Q	R	R
Pesan dengan segera			X			
Menunggu sampai pesanan biasa ditempatkan	X	X				
Dicek oleh supervisor						
Kirim pembelian untuk ditawarkan				X	X	
						X

Keterangan:

- A = kurang dari 10
- B = lebih dari 10 dan kurang dari 50
- C = lebih dari 50
- D = kurang dari atau sama dengan 50
- P = kurang dari 50 unit
- Q = lebih dari 50 dan kurang dari 100
- R = lebih dari 100 unit

### 3. STRUCTURED ENGLISH

Structured English (SE) merupakan alat yang cukup efisien untuk menggambarkan suatu algoritma. Pseudocode merupakan alternatif dari Structured English. Kadang-kadang SE ini disebut dengan istilah PDL (Program Definition Language). Saat logika proses melibatkan rumus-rumus atau iterasi atau saat keputusan-keputusan terstruktur tidak terlalu rumit, maka teknik inilah yang sesuai untuk menganalisis proses keputusan [1]. Structured English digunakan untuk menggambarkan algoritma yang akan dikomunikasikan dengan programmer. Dengan SE ini akan memudahkan programmer untuk memahami dan menggunakannya.

Struktur keputusan (decision structure) atau struktur seleksi (selection structure) dapat berupa struktur IF-THEN atau struktur IF-THEN-ELSE atau struktur Case.

#### 3.1. STRUKTUR IF-THEN

Bentuk struktur ini adalah:

```
IF Kondisi
THEN Tindakan
    Contoh: IF nilai > 60
```

```
THEN Lulus = lulus + 1
```

Struktur IF-THEN-ELSE

Bentuk struktur ini adalah:

```
IF Kondisi
THEN Tindakan 1
ELSE Tindakan 2
```

Contoh: IF nilai > 60

```
THEN Lulus = lulus + 1
```

```
ELSE Ulang = Ulang + 1
```

#### 3.2. STRUKTUR CASE

Struktur ini akan memproses sebuah blok statemen pada suatu kasus yang mempunyai beberapa kondisi yang harus diseleksi. Berikut ini contoh struktur Case:

```
do case
case umur < = 5
    Tunjangan = 5000
case umur > 5 AND umur < 10
    Tunjangan = 15000
case umur > = 10
    Tunjangan = 50000
endcase
```

### 4. RANCANGAN PERANGKAT LUNAK

**Data masukan** yang dibutuhkan oleh perangkat lunak adalah data-data yang mendukung terbentuknya suatu tabel keputusan yaitu data untuk

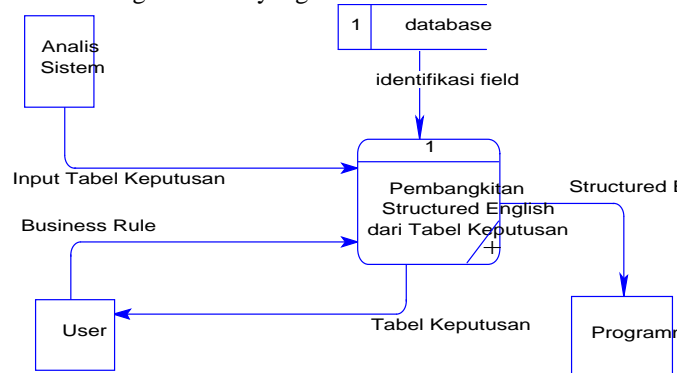
*condition stub*, *condition entry*, *action stub* dan *action entry* serta data masukan dari database yang berelasi dengan aturan pengguna.

**Data keluaran** yang dihasilkan dari perangkat lunak adalah sebuah *Structured English* dengan tipe struktur keputusan IF-THEN-ELSE dari tabel keputusan yang dibuat oleh perangkat lunak dan *Structured English* ini bisa disimpan dalam bentuk file txt. Selain itu setiap project yang dibuat bisa disimpan untuk proses selanjutnya ke dalam file prj.

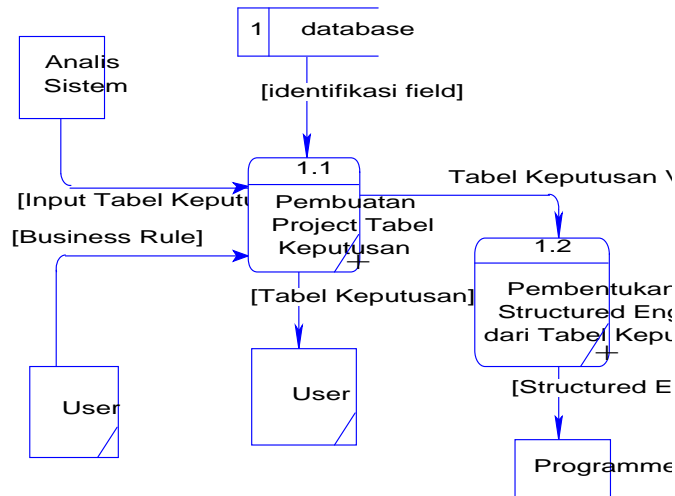
Perancangan proses dari perangkat lunak ini menggunakan pendekatan fungsional yang

direpresentasikan dengan menggunakan *Data Flow Diagram* (DFD) atau Diagram Aliran Data (DAD). Diagram aliran data ini menjelaskan alur proses mulai dari level 0 sampai dengan level 2.

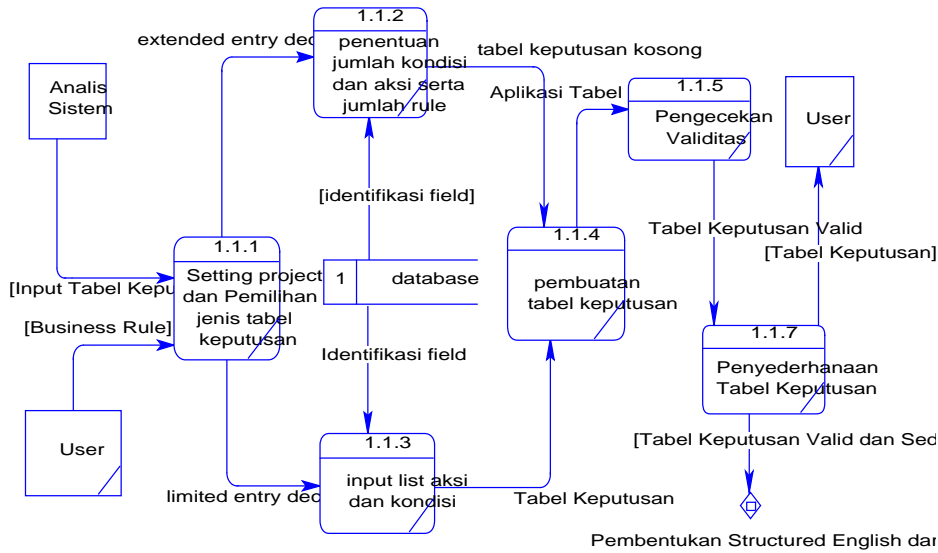
Pada DAD level 0 pada gambar berikut menunjukkan bahwa proses pembangkitan *Structured English* membutuhkan data masukan yang berupa tabel keputusan yang sudah dipersiapkan terlebih dahulu oleh analis sistem.



Gambar 1. Dad level 0



Gambar 2. Dad level 1



Gambar 3. Dad level 2

Pada DAD level 1 seperti pada gambar 5, proses pembangkitan Structured English dijabarkan menjadi 2 subproses, yaitu proses pembuatan project tabel keputusan dan proses pembentukan Structured English dari tabel keputusan.

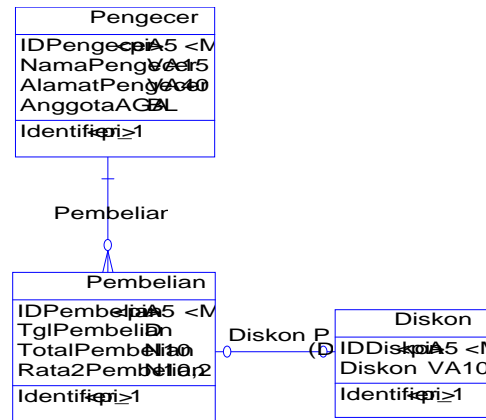
Penjabaran lebih detil dari proses pembuatan project tabel keputusan ditunjukkan pada DAD level 2 pada gambar 6.

**5. UJICOBA DAN ANALISIS**

Pada uji coba ini ada beberapa skenario yang diujikan, yaitu: pembuatan tabel keputusan, pengecekan validitas, penghapusan rule, dan penyederhanaan tabel keputusan, pembangkitan Structured English dari tabel keputusan.

Untuk melakukan ujicoba pembuatan *limited entry decision table*, berikut ini diberikan suatu aturan yang akan diubah menjadi *limited entry decision table*. Aturan ini berkaitan dengan pemberian diskon kepada setiap pembelian yang dilakukan oleh pedagang pengecer dan struktur database *Retail.mdb* seperti pada gambar 7.

“Diskon minimum 5 % diberikan kepada semua pembelian. Tetapi jika seorang pedagang eceran mampu mempertahankan volume rata-rata pembeliannya per bulan paling sedikit Rp 100.000, diskon 15 % akan diberikan kepada pengecer tersebut dan jika mereka merupakan anggota AGA. Ketika volume pembelian pedagang eceran di bawah Rp 100.000, diskonnya adalah 12 persen untuk anggota AGA dan 7 persen untuk bukan anggota. Pengecer yang bukan anggota AGA, tetapi dapat memelihara volume pembelian Rp 100.000 per bulan akan diberikan diskon 10%, kecuali jika total pembelian mereka kurang dari Rp 35.000,-“

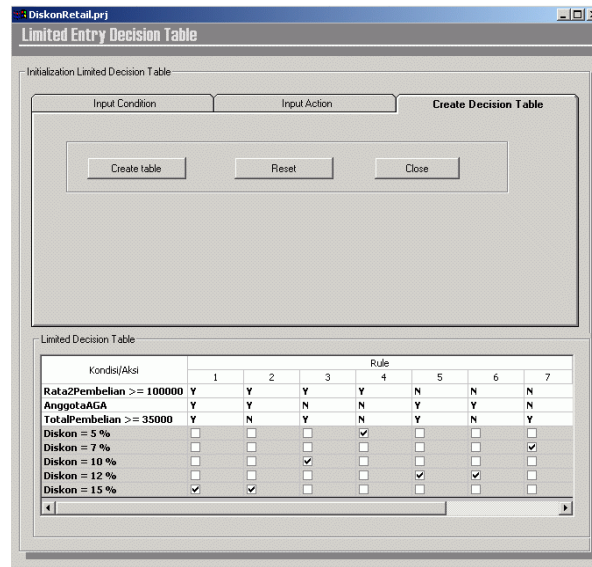


Gambar 4. Struktur database retail.mdb

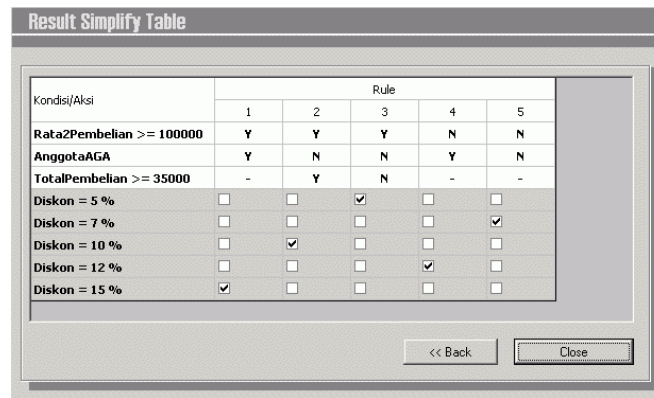
Tabel 4. Bentuk Tabel Keputusan Limited Entry

Condition/Action	Rules			
	1	2	...	N
Rata2 pembelian >= 100000				
Anggota AGA				
Total pembelian >= 35000				
Diskon 5%				
Diskon 7%				
Diskon 10%				
Diskon 12%				
Diskon 15%				

Setelah semua kondisi dan aksi diinisialisasi, Klik *button create table* pada *Tab Create Decision Table*. Dan akhirnya sebuah tabel keputusan dengan tipe *limited entry* berhasil dibuat. Hasil akhir pembuatan tabel keputusan ini dapat dilihat pada gambar 9.



Gambar 5. Hasil pembuatan tabel keputusan *limited entry*



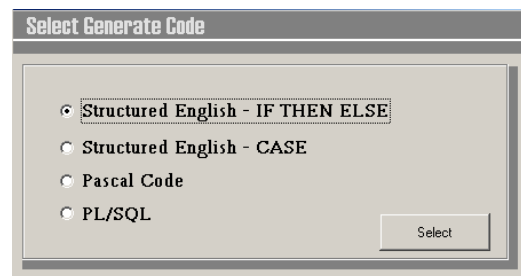
Gambar 6. Hasil penyederhanaan tabel

Ujicoba untuk pengecekan validitas dilakukan setelah tabel keputusan *limited entry* terbentuk. Ujicoba ini dilakukan dengan cara klik menu **Object/Validation Checking** atau *button check* pada toolbar. Hasil pengecekan akan ditampilkan pada *Form Check Validation Decision Table* yang menunjukkan sebuah pesan bahwa tabel keputusan yang dibuat adalah valid dimana tidak ada redundansi dan kontradiksi.

Ujicoba selanjutnya adalah penyederhanaan table keputusan yang dilakukan setelah database valid, caranya klik menu **Object/ Simplify Table** atau *button simplify* pada toolbar. Hasil penyederhanaan ditampilkan pada gambar 10.

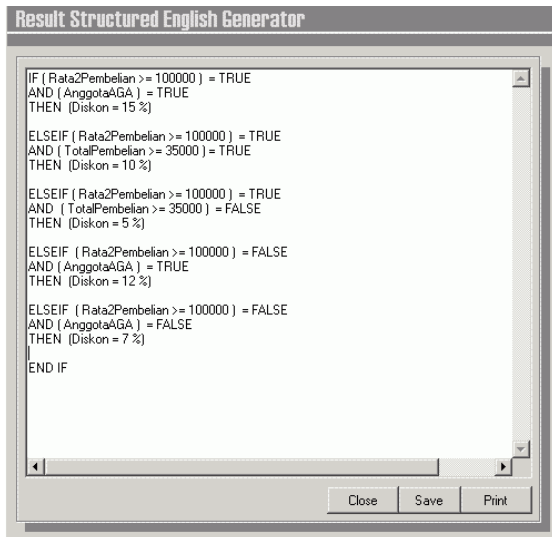
Ujicoba pembangkitan *Structured English* dapat dilakukan pada table keputusan yang telah disederhanakan maupun pada table keputusan kompleks. Pada ujicoba ini akan dilakukan pembangkitan *structured English* dari tabel keputusan yang telah disederhanakan. Untuk ujicoba ini klik menu **Object/Generate Code** atau tombol *Simplify* pada toolbar. Sebelum proses pembangkitan

dilakukan, hal yang diperlu dilakukan adalah pemilihan jenis code yang akan dihasilkan. Namun pada perangkat lunak ini hanya di default untuk Structured English – IF THEN ELSE. Pemilihan ini dapat dilihat pada gambar 11.



Gambar 7. Pemilihan bentuk code

Setelah itu, proses pembangkitan dilakukan dan hasil pembangkitan *Structured English* dapat dilihat pada gambar berikut ini.



```

Result Structured English Generator
IF ( Rata2Pembelian >= 100000 ) = TRUE
AND ( AnggotaAGA ) = TRUE
THEN ( Diskon = 15 % )

ELSEIF ( Rata2Pembelian >= 100000 ) = TRUE
AND ( TotalPembelian >= 35000 ) = TRUE
THEN ( Diskon = 10 % )

ELSEIF ( Rata2Pembelian >= 100000 ) = TRUE
AND ( TotalPembelian >= 35000 ) = FALSE
THEN ( Diskon = 5 % )

ELSEIF ( Rata2Pembelian >= 100000 ) = FALSE
AND ( AnggotaAGA ) = TRUE
THEN ( Diskon = 12 % )

ELSEIF ( Rata2Pembelian >= 100000 ) = FALSE
AND ( AnggotaAGA ) = FALSE
THEN ( Diskon = 7 % )

END IF
Close Save Print

```

Gambar 8. Hasil pembangkitan *structured english*

Dari uji coba yang dilakukan terhadap perangkat lunak yang dibuat, terlihat perangkat lunak telah bekerja dengan baik dan menjalankan semua fungsi yang diharapkan.

Pada skenario pertama hal ini terlihat bahwa tabel keputusan berhasil dibuat dengan baik oleh perangkat lunak baik untuk *limited* maupun *extended entry decision table* sesuai dengan format dari tabel keputusan acuan.

Selanjutnya skenario kedua, pengecekan validitas yang meliputi pengecekan redundansi dan kontradiksi bisa dilakukan oleh perangkat lunak dalam penelitian ini baik *limited* maupun *extended*. Demikian juga untuk penyederhanaan tabel yang dapat disederhanakan oleh perangkat lunak secara otomatis, hasilnya sama dengan jika proses penyederhanaan dilakukan secara manual. Selain itu, perangkat lunak juga bisa melakukan penghapusan rule-rule yang tidak terpakai dengan baik.

Untuk skenario ketiga, perangkat lunak mampu membangkitkan tabel keputusan ke dalam bentuk *Structured English* dengan struktur keputusan IF THEN ELSE. Pembangkitan ini bisa dilakukan pada tabel keputusan dengan tipe *limited entry*, *extended*

*entry decision table* maupun untuk tabel keputusan yang telah disederhanakan.

## 6. KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari Penelitian ini adalah:

- Adanya perangkat lunak ini bisa membantu analisis sistem dalam mengembangkan sebuah tabel keputusan, sekaligus menjembatani antara analisis sistem, user dan *programmer* dalam memahami dan implementasi suatu aturan atau *business rule* dan juga membantu dalam mendokumentasikan aturan tersebut.
- Perangkat lunak yang dibuat mampu melakukan pembuatan project tabel keputusan baik untuk *limited* maupun *extended entry decision table*.
- Perangkat lunak mampu melakukan pengecekan validitas tabel, penghapusan rule dan juga penyederhanaan tabel keputusan secara otomatis sehingga memudahkan analisis sistem dalam melakukan analisis aturan atau *business rule* yang berbentuk tabel keputusan.
- Pembangkitan *Structured English* dari tabel keputusan mampu dilakukan oleh perangkat lunak sehingga memudahkan *programmer* dalam menerapkan aturan ketika pengembangan sistem, tetapi *Structured English* yang dihasilkan hanya berupa struktur keputusan IF-THEN-ELSE saja dan bentuk *struktur english* ini hanya merupakan sebuah pemrograman linier karena tidak mendefinisikan *nested IF* pada hasil pembangkitannya.

## 7. DAFTAR PUSTAKA

- Jogiyanto HM. "Analisis dan Desain Sistem Informasi", ANDI Yogyakarta, 1995
- Kendall, Kenneth E. dan Kendall, Julie E., "System Analysis and Design", Fifth Edition, Prentice Hall, 2002
- Pressman, Roger S. "Software Engineering: A Practitioner Edition", Fourth Edition, The McGraw-Hill, 1997
- Whitten, Jeffrey L dan Bentley, Lonnie D. "Systems Analysis and Design Methods", Fourth Edition, McGraw-Hill, 1998