PERANCANGAN PERANGKAT LUNAK KAMUS INDONESIA-JEPANG

SKRIPSI

Diajukan untuk menempuh Ujian Akhir Sarjana
Fakultas Teknik
Program Strata I Jurusan Teknik Informatika
Universitas Komputer Indonesia

DANI SUHARDANI
10199022



PROGRAM STRATA I
JURUSAN TEKNIK INFORMATIKA
FAKULTASTEKNIK
UNIVERSITAS KOMPUTER INDONESIA

BANDUNG
2004
LEMBAR PENGESAHAN

PERANCANGAN PERANGKAT LUNAK KAMUS INDONESIA - JEPANG

Penyusun : Dani Suhardani

NIM : 10199022

Pembimbing I	Pembimbing II
Bambang Siswoyo, S.T., M.T.	**Mira Kania Sabariah, S.T.**
NIP. 4127 70 06 010	**NIP. 4127 70 06 008**

Mengetahui,

Ketua Jurusan Teknik Informatika

Budhi Irawan, S.Si.
NIP. 4127 70 06 03

LEMBAR PENGESAHAN

PERANCANGAN PERANGKAT LUNAK
KAMUS
INDONESIA - JEPANG

Penyusun : Dani Suhardani

NIM : 10199022

Penguji I **Penguji II**

Bambang Siswoyo, S.T., M.T.
NIP. 4127 70 06 010

Mira Kania Sabariah, S.T.
NIP. 4127 70 06 008

Penguji III

Muhammad Nasrun, S. Si.
NIP. . 4127 70 06 011

Di saat keadaan ku mulai terpuruk, kau hadir dalam kehidupan ku. Kau berikan aku angin segar agar ku bangkit dan maju. Kau membuat hidup ini menjadi lebih indah dan berarti, semoga pengalaman ini menjadikan ku lebih baik dari diriku sebelumnya.

Ya Allah ampunilah dosa-dasa ku dan kedua orang tuaku

•g Ya Allah bimbinglah diriku ini agar menjadi manusia shaleh yang berjalan dan berjuang dijalan Mu•h

•gTunjukilah kami jalan yang lurus. Yaitu jalan orang-orang yang telh Engkau anugerahkan nikmat kepada mereka, bukan (jalan) orang-orang yang dimurkai dan bukan (pula jalan) mereka yang sesat•h. (A Faatihah 5-6)

Skripsi ini kupersmbahkan untuk kedua Orang Tua ku tersayang yang telah sabar mendidik, membesarkan, mengajar ku hingga sekarang, serta kakak-kakakku tercinta yang menjadi panutan ku A Uun, Teh Ninin & Wiran.

KATA PENGANTAR

Assalamualaikum wr. wb

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan Skripsi dengan judul : **PERANCANGAN PERANGKAT LUNAK KAMUS INDONESIA - JEPANG** pada waktunya.

Adapun maksud dari penyusunan skripsi ini adalah untuk memenuhi persyaratan ujian akhir program strata satu jurusan Teknik Informatika di Universitas Komputer Indonesia (UNIKOM) Bandung.

Dalam penulisan skripsi ini penulis menyadari dengan keterbatasan ilmu dan kemampuan yang dimiliki, masih banyak kekurangan dan kesalahan dalam penyusunannya sehingga dibutuhkan kritik dan saran dalam memperbaiki karya ilmiah di masa mendatang.

Tak lupa pula penulis ucapkan terima kasih kepada:

1. Bapak Ir. Eddi Soeryanto Soegoto, M.Sc selaku Rektor Universitas Komputer Indonesia.
2. Bapak Prof. Dr. Ir. Ukun Sastraprawira, M.Sc selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia.
3. Bapak Budhi Irawan, S.Si selaku Ketua Jurusan Teknik Informatika Universitas Komputer Indonesia.
4. Bapak Bambang Siswoyo, M.T. S.T selaku Dosen Pembimbing I Teknik Informatika.
5. Ibu Mira Kania Sabariah, S.T selaku Dosen Pembimbing II Teknik Informatika, yang tidak bosan-bosannya membimbing penyusun selama masa bimbingan hingga akhir penulisan ini.
6. Seluruh Dosen Pengajar dan Staff di Universitas Komputer Indonesia.
7. Bapak dan Ibu, kakak-kakakku tersayang yang telah mendoakan dan memberikan dukungan baik spriritual maupun materiil.

8. Spesial untuk Lia Purnama Sari (Lumut) tercinta yang setia mendukung dan memberi semangat dalam pengerjaan skripsi ini.
9. Anak-anak IF-1 angkatan •f99, yang telah menempuh hidup baru di dunia kerja : Ary Abdillah, Dede Tito, Ade Supriadi, Tantri, Agnes, Gunawan. Kepada Dian saya do•fakan semoga cepat mendapat pekerjaan. Teman seperjuangan dalam seminar dan sidang tahun 2004 : Agung, Iwan, Kinkin, Budi, Wilman, Dede wahyudi, Dede Abdul, Yuyun, dan teman seperjuangan lainnya. Terima kasih kalian telah membuat penulis untuk bersemangat dalam pengerjaan skripsi ini dan telah mendorong ketika penulis putus asa dalam penyusunan skripsi ini. Dan tidak lupa kepada Icha dan Ebi yang masih betah di UNIKOM semoga semester depan kalian bisa mengikuti seminar dan sidang selanjutnya dan lulus. Dan rekan-rekan yang lain di angkatan •f99 yang tidak disebutkan namanya terimakasih telah memberikan kenangan selama kuliah di UNIKOM.
10. Bapak dan Ibu di kelas IF-10 karyawan angkatan •f00 yang telah memberikan kenangan dalam satu tahun terakhir ini dan masukan-masukannya.

Dengan adanya laporan skripsi ini diharapkan dapat berguna bagi penulis sendiri dan juga bagi pembaca pada umumnya. Akhir kata penulis ucapkan terima kasih lagi kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, semoga pahala dari Allah SWT diberikan-Nya kepada semua pihak. Amin.

Wassalamualaikum wr. wb.

Bandung, Agustus 2004

Penulis

Dani Suhardani

DAFTAR ISI

KATA PENGANTAR i
DAFTAR ISI iii
DAFTAR GAMBAR v
DAFTAR TABEL vii
ABSTRACT viii
ABSTRAKSI ix
BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Ruang Lingkup 3
1.4 Tujuan 3
1.5 Metodologi Penelitian 3
1.6 Sistematika Penulisan 4
BAB II LANDASAN TEORI
2.1 Kamus 6
2.2 Pengertian Basis Data 6
2.3 Keuntungan Basis Data 7
2.4 Pohon Sintaks 7
2.5 Analisis Leksikal 13
2.5.1 Tugas Scanner 13
2.5.2 Besaran Leksik 14
2.6 Metode Parsing 15
2.6.1 Parsing Dengan Recursive Decent Parser 16
BAB III ANALISIS DAN PERANCANGAN SISTEM
3.1 Analisis Kebutuhan Sistem 25
3.2 Analisis Sistem 25
3.3 Data Flow Diagram 26

3.3.1. Diagram Konteks 26
3.3.2. DFD Level 0 27
3.4 Aturan Produksi 28
3.5 Analisis Masalah 33
3.6 Algoritma 37
3.6.1 Algoritma Scanning 37
3.6.2 Algoritma Parsing 38
3.6.3 Algoritma Pertukaran Objek Predikat 39
3.7 Batas Perancangan Sistem 40
3.8 Perancangan Perangkat Lunak 40
BAB IV IMPLEMENTASI SISTEM
4.1 Implementasi 43
4.1.1 Proses Masukan File 43
4.1.2 Proses Keluaran File 44
4.2 Analisa Hasil 47
BAB V KESIMPULAN DAN SARAN
5.1 Kesimpulan 48
5.2 Saran 48
DAFTAR PUSTAKA 49
LAMPIRAN

DAFTAR GAMBAR

1. Gambar 2.1. Sebuah Pohon Sintaks 9
2. Gambar 2.2. Pohon Penurunan untuk untai •eaabbbl 10
3. Gambar 2.3. Pohon penurunan untuk untai •eaabbaa 11
4. Gambar 2.4. Pohon penurunan untuk untai •eaabbabbba¶ 13
5. Gambar 2.5. Diagram Sintaks Program 22
6. Gambar 2.6. Diagram Sintaks Block 23
7. Gambar 2.7. Diagram Sintaks Statement 23
8. Gambar 2.8. Diagram Sintak Exp 23
9. Gambar 3.1. Diagram Konteks 26
10. Gambar 3.2. DFD Level 0 27
11. Gambar 3.3. Pohon Sintaks Bahasa Indonesia 30
12. Gambar 3.4. Pohon Sintaks Bahasa Jepang 31
13. Gambar 3.5. Subpohon Bahasa Indonesia 31
14. Gambar 3.6. Diagram Keadaan 34
15. Gambar 3.7. Diagram Sintaks Kalimat 35
16. Gambar 3.8. Diagram Sintaks Subjek 35
17. Gambar 3.9. Diagram Sintaks Kata Benda 36
18. Gambar 3.10. Diagram Sintaks Predikat 36
19. Gambar 3.11. Diagram Sintaks Kata Kerja 36
20. Gambar 3.12. Diagram Sintaks Objek 36

21. Gambar 3.13. Diagram Sintaks K Benda dan K Sifat 37
22. Gambar 3.14. Tampilan Menu Utama 41
23. Gambar 3.15. Tampilan Menu edit Database 42
24. Gambar 4.1. Tampilan Awal Sistem 43
25. Gambar 4.2. Proses Masukan File 44
26. Gambar 4.3. Tampilan Database 45
27. Gambar 4.4. Tampilan Tanda Baca 45
28. Gambar 4.5. Hasil Proses 46
29. Gambar 4.6. Tampilan Open 47

DAFTAR TABEL

1. Tabel 3.1. Daftar Proses 27

ABSTRACT

Langguage is a communication tools thats people use in their live. Lots of langguage makes a different to a place to another one•fs even to one nation to another. We can learn another langguage using a dictionary, but to find the meaning words it takes to much time.

ABSTRAKSI

Bahasa merupakan alat komunikasi yang selalu dipakai dalam kehidupan sehari-hari. Banyaknya beragam bahasa membuat perbedaan diantara suatu daerah dengan daerah lainnya ataupun suatu bangsa dengan bangsa yang lainnya. Agar dapat mempelajari suatu bahasa kita dapat mempelajarinya menggunakan kamus. Tetapi pada proses pencarian arti memakan waktu yang lama.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Bahasa merupakan salah satu alat komunikasi. Di dunia ini banyak terdapat beragam bahasa salah satunya adalah bahasa indonesia•C selain itu ada bahasa lainnya seperti bahasa jepang, bahasa inggris, bahasa jerman, dan sebagainya. Salah satu cara mempelajari bahasa adalah dengan menggunakan kamus.

Ketertarikan pada bahasa ini membuat penulis ingin mengimplementasikan materi yang pernah didapat untuk membuat kamus. Kamus yang dipilih adalah kamus indonesia-jepang, mengambil kamus indonesia-jepang itu sendiri dikarenakan penulis sedang melakukan pembelajaran terhadap bahasa jepang, selain itu bahasa jepang mempunyai huruf-huruf unik seperti hiragana, katakana, dan kanji.

Penggunaan kamus indonesia-jepang ini ditujukan untuk pengguna komputer yang ingin mempelajari bahasa asing khususnya bahasa jepang. Memang kamus biasa lebih praktis untuk dibawa-bawa tetapi dalam proses pencarian akan membutuhkan waktu dan belum lagi bisa kotor atau sobek juga bisa terkena air oleh karena itu penulis ingin mencoba membuat kamus indonesia-jepang ini untuk komputer, dimana proses pembuatan menggunakan basis data untuk penanganan data-datanya agar tersimpan, tersaji, dan terorganisir dengan baik dan juga dalam proses pencarian tidak memerlukan waktu yang lama.

Berdasarkan uraian diatas maka penulis akan mengambil judul

PERANCANGAN PERANGKAT LUNAK KAMUS INDONESIA-JEPANG•h

1.2. Rumusan Masalah

Dalam aplikasi pembuatan kamus ini diperlukan sebuah basis data yang akan digunakan untuk membuat kesimpulan. Pada proses pencarian dilakukan dengan metode scan dan parsing, dimana kalimat yang akan dicari artinya akan discan kemudian akan dicocokan kedalam database yang ada dan akan ditampilkan kalimat yang dicari.

1.3. Ruang Lingkup

Kamus Indonesia ? jepang ini bisa menterjemahkan kalimat tetapi pada hasil output tidak terdapat partikel yang mengikutinya, dan bentuk outputpun berupa *jishokei* bukan bentuk-bentuk lain. Input masukan bahasa indonesia, diusahakan pada masukan kalimat bahasa indonesia berbentuk SPO (Subjek Predikat Objek), dimana input dengan menggunakan bahasa indonesia akan mengeluarkan keluaran tulisan hiragana dan dapat pula berupa tulisan katakana. Jika kalimat atau kata tidak ada didalam database maka kata yang dimasukan tadilah yang akan ditampilkan.

1.4. Tujuan

Tujuan perancangan *software* ini adalah untuk memudahkan pengguna mencari kata ataupun kalimat dalam bahasa jepang sehingga tidak perlu melihat kedalam kamus manual.

1.5. Metodologi Penelitian

Metodologi yang digunakan dalam penulisan skripsi ini adalah :

1. Studi Literatur

    1. Studi kepustakaan

        yaitu teknik pengumpulan data dengan cara mengumpulkan buku-buku yang ada kaitannya dengan permasalahan yang diambil.

    2. Searching internet

        Yaitu pencarian data melalui internet dengan mengunjungi alamat website yang berkaitan dengan masalah.

1.6. Sistematika Penulisan

BAB I Pendahuluan

Bab ini mengemukakan tentang gambaran keseluruan mengenai Latar belakang, Rumusan masalah, Ruang lingkup, Tujuan, Metodologi Penelitian, serta Sistematika Penulisan.

BAB II Landasan Teori

Dalam bab ini dijelaskan lebih lanjut mengenai teori-teori dasar mengenai basis data dan parsing.

BAB III Analisis Perancangan Sistem

Berisi tentang perancangan sistem yang akan dibangun, identifikasi masukan, identifikasi keluaran dan antar muka yang meliputi proses masukan dan proses keluaran, aturan produksi, batasan perancangan sistem dan algoritma.

BAB IV Implementasi

Dalam bab ini, penulis menyajikan tentang teknik implementasi serta, menentukan kebutuhan input dan output.

BAB V Kesimpulan dan Saran

Dalam bab ini, penulis menyajikan tentang kesimpulan dan saran-saran terhadap skripsi ini.

BAB II

LANDASAN TEORI

2.1. Kamus

Kamus merupakan salah satu jenis buku yang dipergunakan manusia sebagai alat pembelajaran mengenai bahasa, dan juga kamus dapat dipergunakan sebagai alat penterjemah dari satu bahasa kedalam bahasa yang lainnya.

2.2. Pengertian Basis Data

Basis data merupakan kumpulan data tentang suatu organisasi yang disusun sedemikian rupa sehingga memudahkan dalam pengelolaan dan pengaksesan dimana data-data tersebut saling berhubungan.

Database adalah kumpulan dari data-data yang berhubungan satu sama lainnya yang digunakan untuk pencarian suatu data tertentu pada saat query dijalankan. Sebuah database dirancang, dibuat dan ditempati oleh data dengan tujuan tertentu. Di dalam sistem database relasional, tabel-tabel dari database saling berhubungan satu sama lainnya. Dan sebuah tabel database akan selalu memiliki *attribute names* (nama-nama attribute), *relation names* (nama-nama relasi), dan *tuples* (record-record).

2.3. Keuntungan Basis Data

Basis data mempunyai berbagai keuntungan antara lain:

1. Cepat proses pencarian.
2. Mudah dalam meng edit data.

3. Praktis dalam peremajaan.

4. Keamanan data.

5. Mengurangi kejemuan.

2.4. Pohon Sintaks

Sebuah pohon (tree) adalah suatu *graph* terhubung tidak sirkuler, yang memiliki satu simpul (node)/ *vertex* disebut akar (root) dan dari situ memiliki lintasan ke setiap simpul. Gambar 2.1 memberikan contoh sebuah tree yang menguraikan kalimat dalam bahasa Inggris.

The quick brown fox jumped over the lazy dog

Pohon Sintaks/ pohon penurunan (syntax tree/derivation tree/parse tree) berguna untuk menggambarkan bagaimana memperoleh suatu string (untai) dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal. Setiap simbol variabel diturunkan menjadi terminal, sampai tidak ada yang belum tergantikan.

Misal terdapat tata bahasa bebas konteks dengan aturan produksi (simbol awal S, selanjutnya di dalam bab ini digunakan sebagai simbol awal untuk tata bahasa bebas konteks adalah S):

$$S \rightarrow AB$$

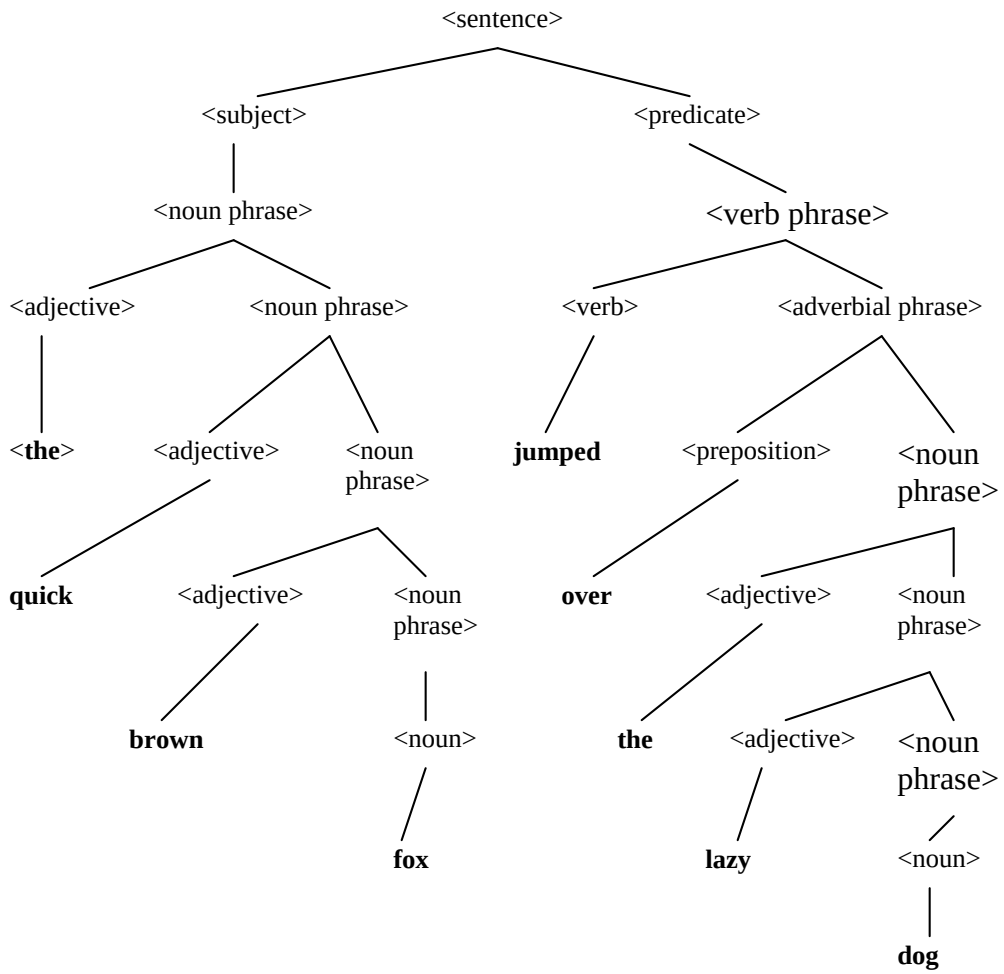
$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

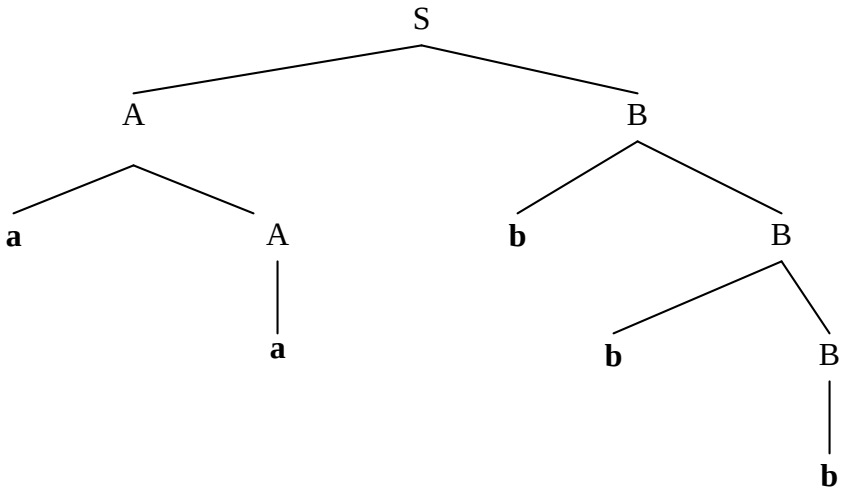
Akan kita gambarkan pohon penurunan untuk memperoleh untai :

aabbbD Pada pohon tersebut simbol awal akan menjadi akar (root). Setiap kali penurunan dipilih aturan produksi yang menuju ke solusi. Simbol-simbol variabel

akan menjadi simpul-simpul yang tidak mempunyai anak akan menjadi simbol terminal. Kalau kita baca simbol terminal yang ada pada gambar 2.2 dari kiri ke kanan akan diperoleh untai IaabbbD



Gambar 2.1 Sebuah pohon sintaks



Gambar 2.2 Pohon penurunan untuk untai aabbbD

Proses penurunan atau parsing bisa dilakukan dengan cara:

A) penurunan terkiri (leftmost derivation): simbol variabel terkiri yang diperluas terlebih dahulu

B) penurunan terkanan (rightmost derivation): simbol variabel terkanan yang diperluas terlebih dahulu

Misal terdapat tata bahasa bebas konteks:

$$S \rightarrow aAS \mid a$$

$$A \rightarrow Sba \mid ba$$

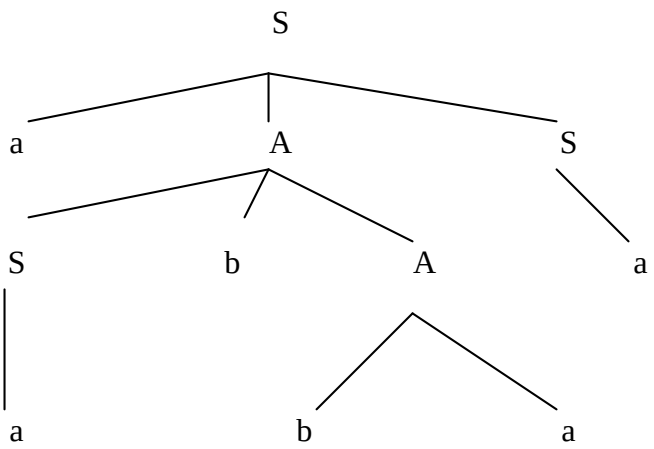
Untuk memperoleh untai aabbaaDdari tata bahasa bebas konteks di atas (➔Î bisa dibaca ¶menurunkanµ):

A) Dengan penurunan terkiri: S ➔ aAS ➔ aSbAS ➔ aabAS ➔ aabbaS ➔ aabbaa

B) Dengan penurunan terkanan : S ➔ aAS ➔ aAa ➔ aSbAa ➔ aSbbaa ➔

aabbaa

Kita lihat pohon penurunnya pada gambar 2.3. Meskipun proses penurunannya berbeda akan tetap memiliki pohon penurunan yang sama.



Gambar 2.3 Pohon penurunan untuk untai Xaabbaa

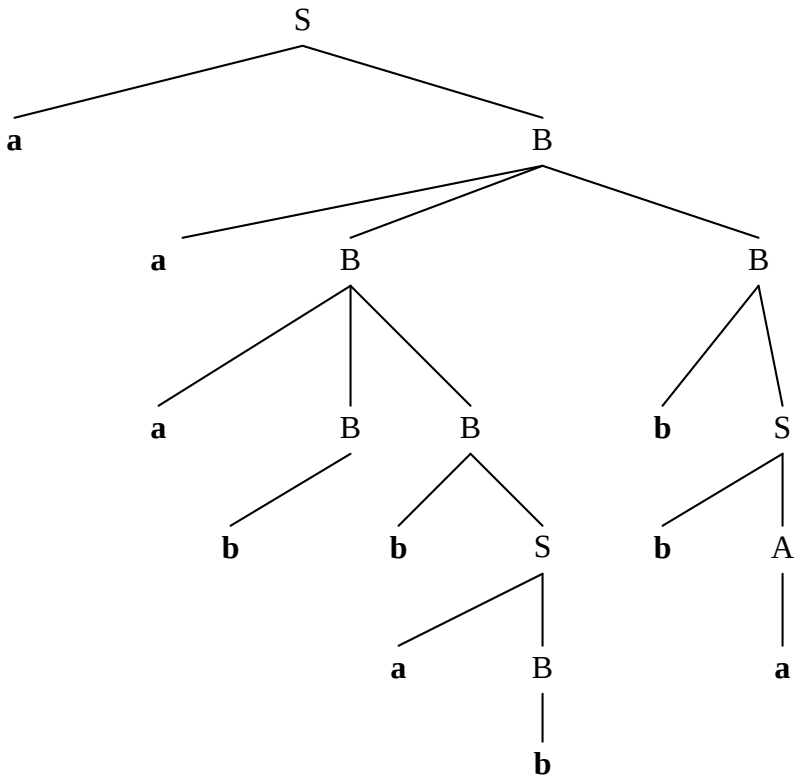
Biasanya persoalan yang diberikan berkaitan dengan pohon penurunan, adalah untuk mencari penurunan yang hasilnya menuju kepada suatu untai yang ditentukan. Dalam hal ini, perlu untuk melakukan percobaan pemilihan aturan produksi yang bisa menuju ke solusi. Misalkan sebuah tata bahasa bebas konteks memiliki aturan produksi:

$$S \rightarrow aB \mid bA$$

A → a | aS | bAA

B → b | bS | aBB

Pohon penurunan untuk memperoleh : 3aaabbabbba¶ bisa dilihat pada gambar 2.4.



Gambar 2.4 Pohon penurunan untuk untai*aabbabbba¶

2.5. Analisis Leksikal

2.5.1. Tugas Scanner

Scanner bertugas melakukan analisis leksikal, yaitu mengidentifikasi semua besaran yang membangun suatu bahasa pada suatu bahasa suatu program sumber. Scanner adalah bagian dari

kompilator yang menerima input berupa stream karakter kemudian memilah program sumber menjadi satuan leksik yang disebut dengan token. Dimana token ini akan menjadi input bagi parser.

2.5.2. Besaran Leksik

Besaran Pembangunan bahasa/leksik meliputi:

1. Identifier

   Identifier ini bisa berupa keywords atau nama. Keywords adalah kata kunci yang sudah didefinisikan oleh suatu bahasa seperti Begin, end, if, else dan sebagainya.

2. Nilai konstanta

   Adalah suatu konstanta yang terdapat pada program. Bisa berupa konstanta integer, real, boolean, character dan sebagainya,

3. Operator dan delimiter

   Operator misalnya operator aritmatika (+,-,*,/), operator logika (<, >, =).

   Delimiter berguna sebagai pemisah/pembatas, misalnya: ( ), ;, :, spasi.

2.6. Metode Parsing

Parsing adalah grup dari subrutin yang mengkonvert *token stream* kedalam parse tree dan parse tree adalah respresentasi structural dari sebuah kalimat yang diparse.

Pengertian parsing secara umum adalah sebuah proses penentuan apakah sebuah string dari token dapat dihasilkan oleh sebuah grammar. Sedangkan parsing pada proses sebuah query adalah merupakan tahapan di mana sintak-sintak dari query akan di-cek untuk menentukan apakah query tersebut sudah dirumuskan sesuai dengan aturan-aturan sintak (aturan-aturan grammar) dari bahasa query. Setelah mengalami proses parsing di dalam parser, maka query tersebut kemudian diproses di dalam optimizer untuk mendapatkan rencana eksekusi.

Proses parsing merupakan tahapan analisis sintaksis yang berguna untuk memeriksa urutan kemunculan *token*. Di dalam mengimplementasikan sebuah metode parsing ke dalam program perlu diperhatikan tiga hal:

1. Rentang waktu eksekusi
2. Penanganan Kesalahan
3. Penanganan kode

Metode parsing bisa digolongkan sebagai berikut:

1. T*op down*

   Kalau dilihat dari terminologi pohon penurunan, metode ini melakukan penelusuran dari *root*/puncak menuju ke *leaf*/daun (simbol awal sampai simbol terminal). Metode *top down* sendiri meliputi:

   a. *Backtrack/backup: Brute Force*

b. *No backtrack: Recursife Descent Parser*

2.6.1. Parsing dengan Recursive Descent Parser

Recursive Descent Parser adalah salah satu cara mengaplikasikan bahasa bebas konteks untuk melakukan analisa sintaksis suatu *source code*. Di sini simbol terminal maupun simbol variabelnya sudah bukan karakter, tetapi berupa besaran leksik sebagai simbol terminalnya dan besaran sintaks sebagai simbol variabelnya. Ciri dari recursive descent parser yang menonjol adalah secara rekursif menurunkan semua variabel dari awal sampai bertemu terminal dan tidak pernah mengambil token secara mundur (no backtrack). Berbeda dengan mesin turing yang dapat maju dan mundur dalam melakukan parsing. Ciri lain dari recursive descent parser adalah sangat bergantung padda algoritma scan dalam mengambil token. Dapat diambil contoh disini sebagai berikut:

Apabila t_ASS (:=) diartikan oleh scnner sebagai t-COL (titik dua) dan t_EQ (sama dengan), maka parser tidak akan dapat mengenali bahwa itu adalah sebenarnya adalah t_ASS (assigment = pemberian nilai). Untuk itu, apabila ingin menggunkan rekursif descent parser, maka disarankan untuk menggunakan scanner yang •gdapat dipercaya•h. Awalnya recursive descen parser ini dapat mengandung back tracking. Contoh, misalkan suatu grammar:

S → c Ad

A → ab | a

Dengan string masukan w = cad untuk membangun sebuah pohon parse bagi string ini secara top-down, mula-mula suatu pohon yang mengandung node

tunggal S diciptakan. Pointer input menunjuk ke c, simbol pertama dari w, lalu dengan menggunkan produksi pertama dari S untuk mengekspansi pohon tersebut.

Dalam banyak hal, dengan menuliskan grammar secara hati-hati, sifat left recursive dari grammar yang dihasilkan, akan dapat diperoleh suatu grammar yang dapat diparse oleh recursif descent parse yang tidak memerlukan back tracking, yaitu predictive parser. Untuk membangun suatu predictive parser, bila diberikan simbol masukan current a dan non terminal A yang akan diekpansi, haruslah dapat diketahui mana diantara alternatif produksi

A 1 | 2 | .. | n

Merupakan alternatif unik yang menurunkan string yang dimulai dari a. dengan perkataan lain, alternatif yang benar harus dapat dideteksi dengan simbol pertama yang diturunkannya. Konstruksi aliran kontrol pada kebanyakan bahasa pemograman, dengan kata kunci yang terbedakan, biasanya dapat dideteksi dengan cara ini.

Di sini yang dimaksudkan dengan rekursif descent parser adalah non back tracking recursive descent parser, yaitu predictive parser,atau bisa juga disebut one pass no back tracking parser. Lebih khusus lagi metode parser yang digunakan adalah LL(1) parser sesuai dengan sifat-sifat proses parsing yang dilakukannya, yaitu top down parsing yang tidak mengandung left recursif.

Untuk mengimplementaskan aturan-aturan produksi memkai teknik recursif descent dengan aturan sebagai berikut:

1. Semua simbol variabel dijadikan prosedur/fungsi.

2. Jika ketemu simbol terminal pada aturan produksi, maka panggil prosedur scan

3. Jika ketemu simbol variabel pada aturan produksi, maka panggil prosedurnya .

4. Penelusuran bersifat top down mengikuti sintaks sesuai pola yang tertera pada diagram sintaks.

5. karena memakai prosedur yang rekursif maka dipakai rekursif stacking yang hampir sama dengan top down parsing metoda brute force dengan limited back up (yaitu dengan implisit stacking, dengan mekanisme rekursif).

6. Metoda recursif descent adalah salah satu metoda parsing tanpa backup. Pada metoda ini urutan produksi direalisasikan dengan menggunakan urutan dari pemanggilan dari fungsi/prosedur.

7. Fungsi/ prosedur ditulis untuk setiap non terminal dari suatu produksi. Setiap fungsi/ prosedur akan melemparkan nilai benar atau salah bergantung pada apakah fungsi tersebut mengenali substring yang diterima sebagai ekspansi dari non terminal.

Sebagai contoh sebuah grammar memiliki aturan produksi (ditulis dalam BNF):

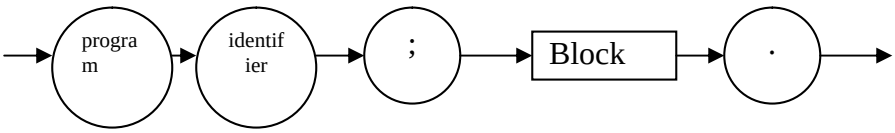
<program> ::= t_PROG t_ID t_SEMICOL <block> t_DOT
<block> ::= t_BEGIN <statement> {t_SIMICOL <statement> } t_END
<statement> ::= t_ID t_ASS <simple_exp> | t_IF <exp> t_THEN
<statement> | t_IF <exp> t+_THEN <statement> t_ELSE <statement>
<exp> ::= <simple_exp> t_EQ <simple_exp> | <simple_exp> t_LT
<simple_exp> | <simple_exp> t_GT <simple_exp>
dan seterusnya•c..(tidak seluruh aturan produksi ditulis

Kita akan membuat prosedur rogram, block, statement, dan exp, berdasarkan notasi BNF di atas.

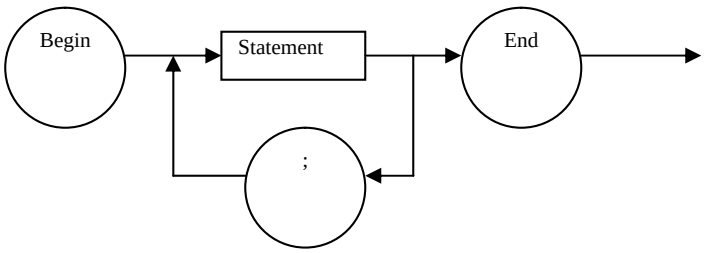
```
Procedure Program
Begin
If token=t_PROG then
Begin
        Scan;
        If token=t_SEMICOL then
        Begin
           Scan;
           Block; {panggil procedure block}
           If token=t_DOT then
           Begin
              Scan;
           End;
        End;
End;
End;
Procedure block
Begin
If token=t_BEGIN then
Begin
   Scan;
   Statement; {pangil prosedur statement}
   While token=t_SEMICOL do
   Begin
      Scan;
      Statement;
   End;
   If token=t_END then
      Scan;
   End;
End;
Procedure Statement
Begin
If token=t_ID then
Begin
   Scan;
   If token=t_ASS then
   Begin
      Scan;
      Simple_exp;
   End;
End;
Else
If token =t_IF then
Begin
   Scan;
   Exp;
   If token=t_THEN then
   Begin
      Scan;
      Statement;
      If token=t_ELSE then
      Begin
         Scan;
         Statement;
      End;
   End;
End;
End;
Procedure Exp;
Begin
Simple_exp;
If (token=t_EQ) or (token=t_LT) or (token=t_GT) then
Begin
   Scan;
   Simple_exp;
End;
End;
```

Diasumsikan sebelum pemanggilan procedure diatas sudah ada token atau procedure scan udah dipanggil. Yang penting diketahui adalah bahwa tiap prosedur yang mewakili simbol variabeldiatas, selesai dipanggil meninggalkan sebuah token jika tidak terjadi kesalahan. Dalam program diatas belum meng-handle penanganan kesalahan.

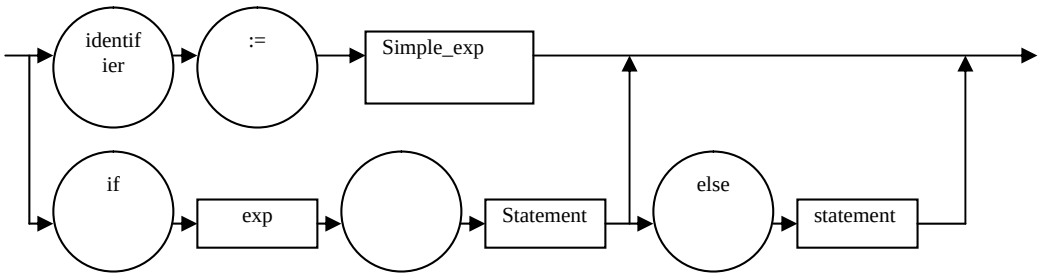
Aturan produksi dalam bentuk notasi BNF pada contoh diatas dapat pula kita buat diagram sintaknya lebih dulu untuk mempermudah.



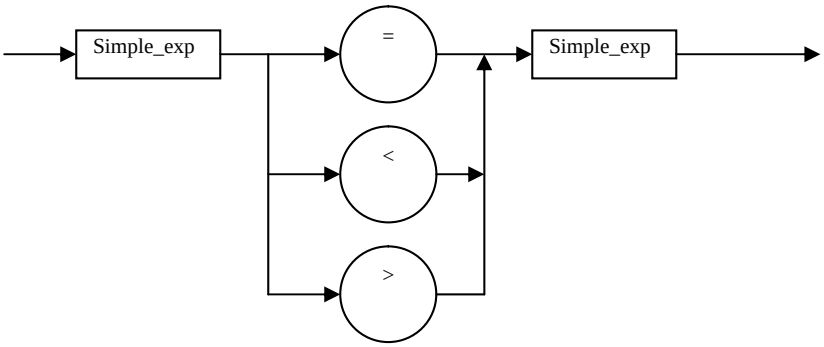
Gambar 2.5. Diagram Keadaan Sintaks Program



Gambar 2.6. Diagram Sintaks Block



Gambar 2.7. Diagram Sintaks Statement



Gambar 2.8. Diagram Sintaks Exp

Pada diagram sintaks di atas langsung digambarkan karaktertokennya misal•f;•f untuk t_SIMICOL, •e.•f untuk t_DOT, •e<•f untuk t_LT, •e:=ÿ t_ASS, dan seterusnya.

Beberapa hal yang perlu diperhatikan pada program di atas:

1. Dalam penanganan error, cukup dilakukan proses halt dengan pemberian pesan keslahan sebelumnya.
2. Beberapa kesalah yang mungkin adalah:

   Jika kesalahan akibat token tidak dikenal.

   Jika kesalahan akibat token tidak pada tempat yang seharusnya.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Kebutuhan Sistem

Untuk merancang sistem ini dibutuhkan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan penulis dalam merancang sistem ini adalah :

1. Komputer AMD XP 2000
2. Memori 128 DDRAM
3. VGA 32 bit

Sedangkan perangkat lunak yang dibutuhkan untuk membuat sistem ini adalah sebagai berikut :

1. Windows sebagai sistem operasi
2. Delphi. 6 untuk membuat program

3.2. Analisis Sistem

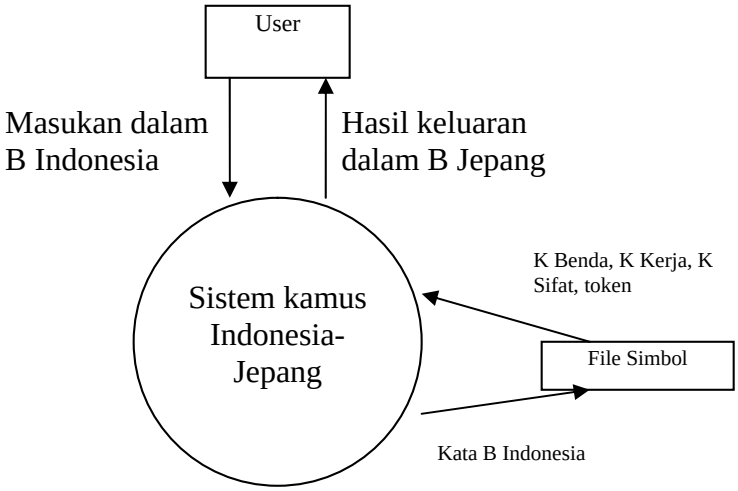
Analisis sistem kamus Indonesia-Jepang, bermaksud menyajikan kata ataupun kalimat dalam bahasa jepang. Sehingga pengguna bisa menggunakan hasilnya sebagai referensi dalam pembelajaran bahasa Jepang.

Untuk dapat menyajikan kebutuhan tersebut maka beberapa beberapa permasalahan yang ada dalam perancangan sistem ini, perlu diperhatikan guna mendukung pelayanan informasi yang dibutuhkan secara cepat.

Oleh karena itu diperlukan perancangan analisis yang dapat menunjang hasil yang lebih interaktif daripada kamus manual.

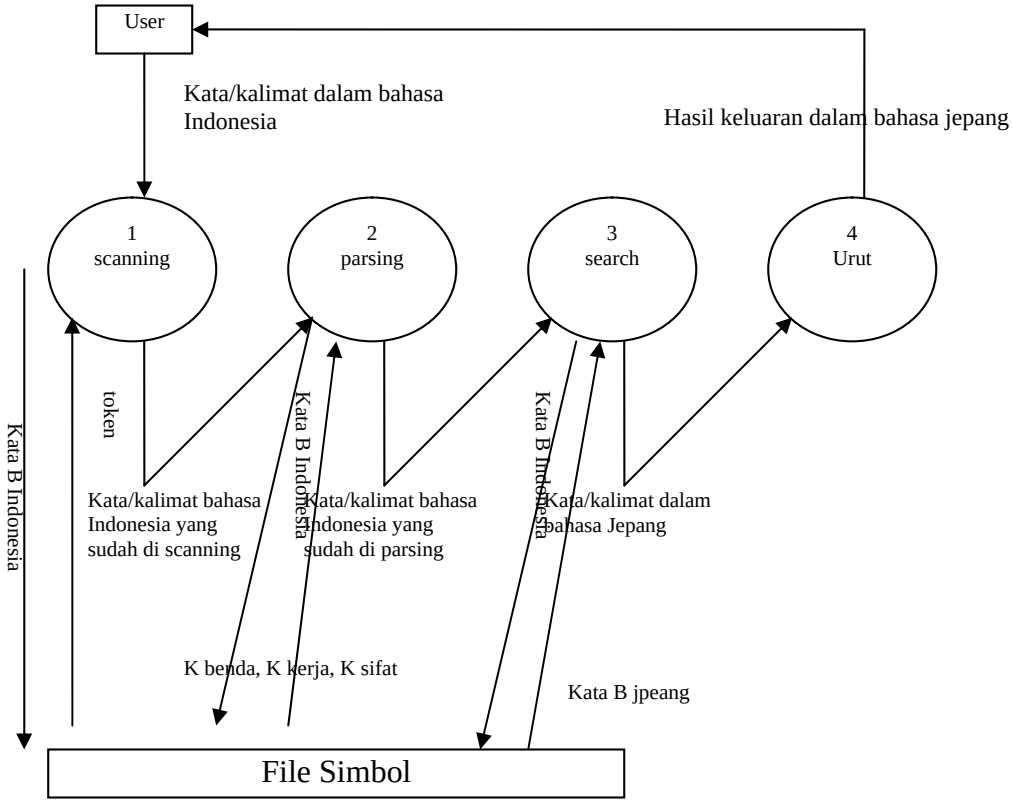
3.3. Data Flow Diagram

3.3.1. Diagram Konteks



Gambar 3.1. Diagram Konteks

3.3.2. DFD Level 0



Gambar 3.2. DFD Level 0

No Proses	Proses	Data Input	Data Output
1	Scanning (kata/kalimat sampai menemukan tanda baca)	Kata/kalimat Bahasa Indonesia	Kata/kalimat Bahasa Indonesia yang sudah di scanning
2	Parsing (melihat aturan masukan)	Kata/kalimat Bahasa Indonesia	Kata/kalimat Bahasa Indonesia yang di parsing
	Search (melakukan pencarian kedalam database)	Kata/kalimat Bahasa Indonesia	Kata/kalimat Bahasa Jepang
	Urut (mengurutkan kata/kalimat sesuai dengan aturan)	Kata/kalimat Bahasa Jepang	Kata/kalimat Bahasa Jepang yang sama dengan aturan produksi

Tabel 3.1. Daftar Proses

3.4. Aturan Produksi

Penyusun mengambil tata bahasa dasar untuk aturan produksi dari proses masukan dalam bahasa Indonesia, dimana masukan tersebut diharapkan agar sesuai dengan aturan yang ada sehingga menghasilkan hasil yang baik. Adapun aturannya sebagai berikut:

1. <Kalimat> ::= <Subjek>spasi<Predikat> spasi <Objek>

2. <Subjek> ::= <Kata Benda>

3. <Predikat> ::= <Kata Kerja>

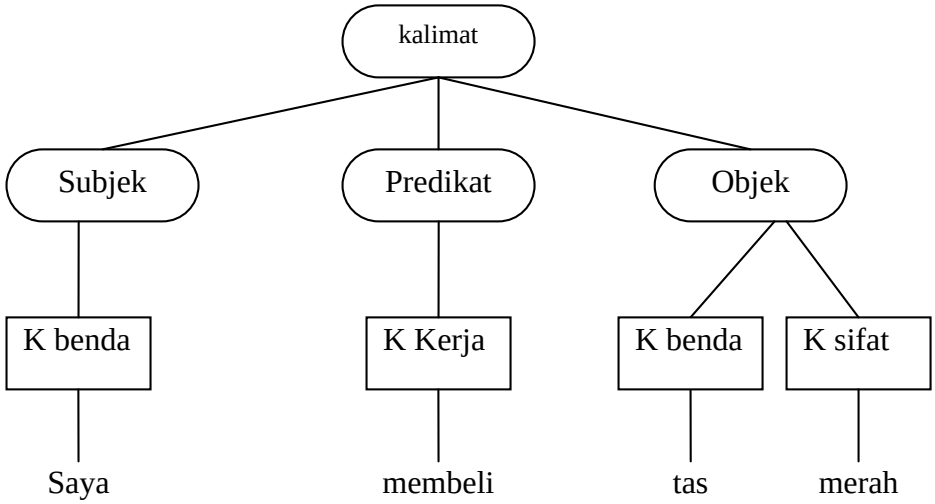
4. <Objek> ::= <Kata Benda> spasi <Kata Sifat>

5. <Kata Benda> ::= artisD awanD arwahD agamaD besokE bibiE bisnis E bisulE asapD asasD bunyiE artileriD dahanG diantaraG artikelD bohongE asramaD basiE bataE batalionE batangE atasD batereE batin E bayiE belakangE asalD agarD adopsiD aspekD berbangga asmara D asliD asistenD akadD asingD asiaD asbakD beranjakE kebangsaanQ kakekN kakiN kantorN keamananN keanehanN kebahagiaan dilarang G kebakaran arenaD kelahiran arcaD adaptorD kepalaN keringatN khawatirN kolamN kudaN kebajikan arsipD asuransiD admiralD abstrakD artiD filmI gambarJ artiD arsitektur hadiahK arsitekD armadaD hujanK arlojiD jambanM jasM ari-ariD bahuE baseballE bagindaE australia aulaD bahanE bahariE bahasaE bahkanE bagianE atomD bajaE bajakE bajiE atmosfirD bajingE bajinganE bajuE bahayaE badaiE awamD ayahD ayanD azabD azimatD babE babakE babiE babuE bacaanE bakalE badakE badanE badaniE badutE bagaiE bagasiE awalD agresiD bangsawanV bangunanE banjirE bankE bankir E banEkuN bantahanE bantalE bakE bantuanE bangsaE agendaD banyolE baptisE baraE barakE barangE atapD agenD laparO bantengE balonE basahE bakatE bakiE bakteriE atletD bakuE bakulE balaE balapanE bangsatE balokE bangsalE bambuE banE banditE ahliD banggaE bangkaiD bangkitE bangkrutE atletikD balasanE rasaU selamatV perbatasan perbuatan akurD perutS perbaikan rapatU alamD rumahU AC$ akuD abukD kursiN rakU pembatalan pamanS alamiD pantaiS pasanganS patahS pedesaanS pembacaS pembakarS perbandingan pencuri S pendapat S pengacara pengangguran pengarahanQ penitipan alamatD pembalutS abangD aktualD akhlakD toiletW TuhanZ akhirD uangX tidakW urusanX akilbalig wcZ akbarD abadD sayaV akarD akalD tasW akademiD nasiQ umpanX tamuW aktris D aktorD aktifD aktaD akibatD suamiV aksiD terbatasW aksaraD akomodasi stasiunV anggotaD analisD angkerD angkasaD angkaD anginD angsuranD anggotaD angus anggaranD andilD andaiD andaD ancamanD anasirD pagi-pagiS anggurD airD arangD lautO adaptasiD arahanD mainanP angkuhD mandiP apotekerD apotekD apiD anugerahD antiD anjingD musikP analisaD alkisahD aljabarD alisD alimD aliasD alhasilD algojoD almariD acarD aluD-alunD alasanD alasD nenekQ alang-alangD arbeiD alatD ambrukD anakD ampunD ampuhD almarhumD ampasD pagiS ambisiD ambasadur ambangD

amatir D amat D amarah D amandemen amandel D amanat D aman D amplop

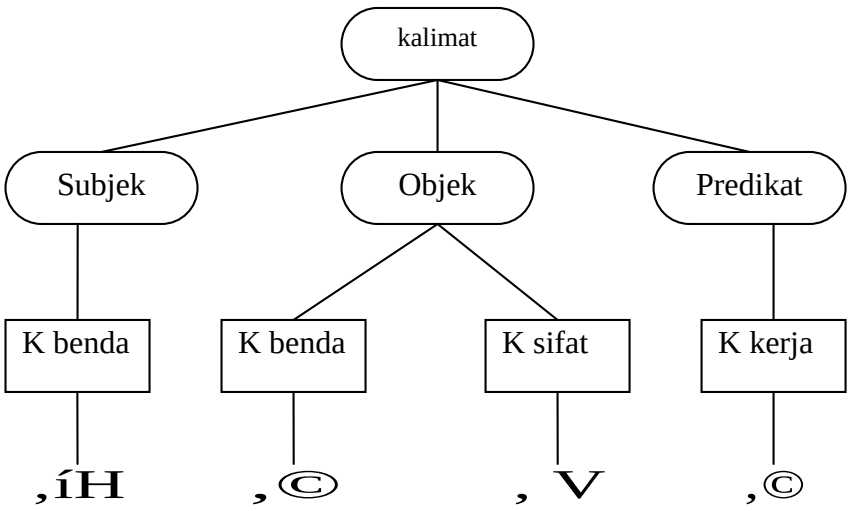
6. <Kata Kerja> ::= erkumpul H mengampuni menganut P mencuci P mendapat P mendukung X menekan P menanam P mengambi Æ E memukul P mengandaikan menganggap mengangkat menganiaya mengantuk membangunkan mengajar P memberikan membantah membantai membantu P membasmi P membasuh P menarik P bergerak E menganyam membuka P memerlukan memilih P memperbaiki memperbanyak membatasi H terlambat mengantup pergi S selesai V abonemen D abnormal D obrolan R teringat W naik Q tuntunan W turun W abadi D memakai P makan P tergesa-gesa J mengingat mengarahkan mengasah P mengatur P mengemudi menghimpun mengikat P membatalkan mengirim P mengunjungi menimbun P menjual P menurunkan X menghimpun jatuh M bertemu E bangun E berbaring hafal K aba-aba D dijual G kehilangan membangunkan jalan-jalan M bersungguh-sungguh V berpikir E berpengalaman bernyanyi berbahagia berbakti E berenang E berbaris E buka E memaafkan membalut P membalik P membalas P membakar P membahas P membagi H baca E membabibuta mematahkan memasukan meletakan mebandingkan mati P melawak
7. <Kata Sifat> ::= amis | mudah | abu | gampang | baik | sedih | malu-malu | hangat | manis | manja | penting | bahagia | bagus | ringan | pedas | kotor | penuh | hebat | suka | bajik | awet | enak | biru | asam | lucu | sibuk | terang | asin | abu-abu | baka | panas | cinta | bakti | tipis | arif | besar | larut | tebal | baru | aneh | merah | lambat | bandel | benci | keras | cerewet | asyik | gembira | sedikit

Dimana aturan ini didapat dari pohon sintaks pada gambar 3.3 sebagai acuan dari aturan produksinya



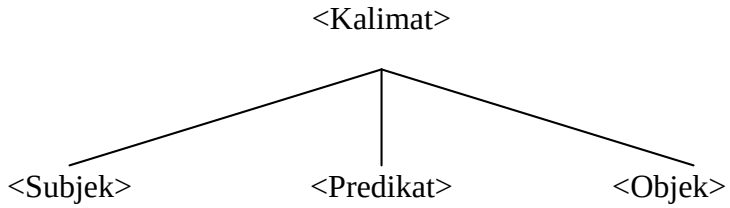
Gambar 3.3. Pohon Sintaks Bahasa Indonesia

dari masukan tersebut kemudian disusun kedalam bahasa Jepang dan pohon sintaksnya adalah sebagai berikut:



Gambar 3.4. Pohon Sintaks Bahasa Jepang

Sub pohon dari gambar 3.3. dapat dilihat sebagai berikut:



Gambar 3.5. Subpohon Bahasa Indonesia

Dan himpunan produksi lainnya adalah sebagai berikut:

1. <Kalimat> ::= <Subjek> spasi <Predikat>

   contoh dalam himpunan produksinya sebagai berikut:

   1. <Kalimat> ::= <Subjek> spasi <Predikat>
   2. <Subjek> ::= <K Benda>
   3. <Predikat> ::= <K Kerja>
   4. <K Benda> ::= Saya
   5. <K Kerja> ::= Makan

2. <Kalimat> ::= <Subjek> spasi <Objek>

   contoh dalam himpunan produksinya sebagai berikut:

   1. <Kalimat> ::= <Subjek> spasi <Objek>
   2. <Subjek> ::= <K Benda>
   3. <Objek> ::= <K Benda>
   4. <K Benda> ::= Saya | bola

3. <Kalimat> ::= <Predikat> spasi <Objek>

   contoh dalam himpunan produksinya sebagai berikut:

   1. <Kalimat> ::= <Predikat> spasi <Objek>

2. <Predikat> ::= <K Kerja>

3. <Objek> ::= <K Benda>

4. <K Kerja> ::= Makan

5. <K Benda> ::= Nasi

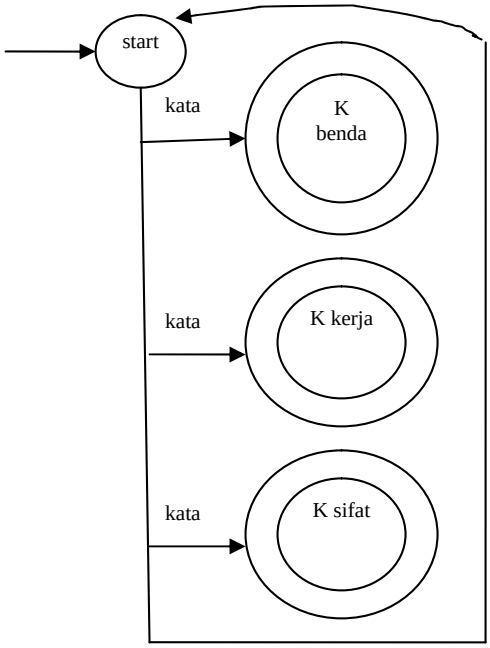
Jika pada proses masukan dimasukan sembarang kata atau kalimat yang tidak memiliki aturan atau pola kalimat yang jelas maka susunan hasil keluarannyapun akan sama seperti pada masukan, untuk itu dapat dilihat dibatasan perancangan sistem pada sub bab 3.7.

3.5. Analisis Masalah

Di dalam sub bab ini akan membahas bagaimana kata dalam sistem diproses.

1. Scanning

   Pada proses ini masukan akan di scan, misalkan masukan berupa kalimat: saya makan nasi. Pertama-tama kata-kata akan dilihat perkarakter jika menemukan spasi maka akan dianggap sebuah kata, kemudian kata tersebut akan disimpan dalam token, kemudian melanjutkannya ke kata selanjutnya dan kata yang sudah di scan tadi pun disimpan ditoken kemudian selanjutnya sampai akhir. Apabila masukannya hanya satu kata, misal saya maka pada proses scan kata tersebut bukan kalimat karena tidak berpasangan dengan kata lainnya, setelah proses scan ini maka akan dilanjutkan pada proses parsing. Dari gambar 3.6. dilihat kata akan masuk kedalam jenis kata yang mana jika telah ditemukan akan disimpan dalam token



Gambar 3.6. Diagram Keadaan

2. Parsing

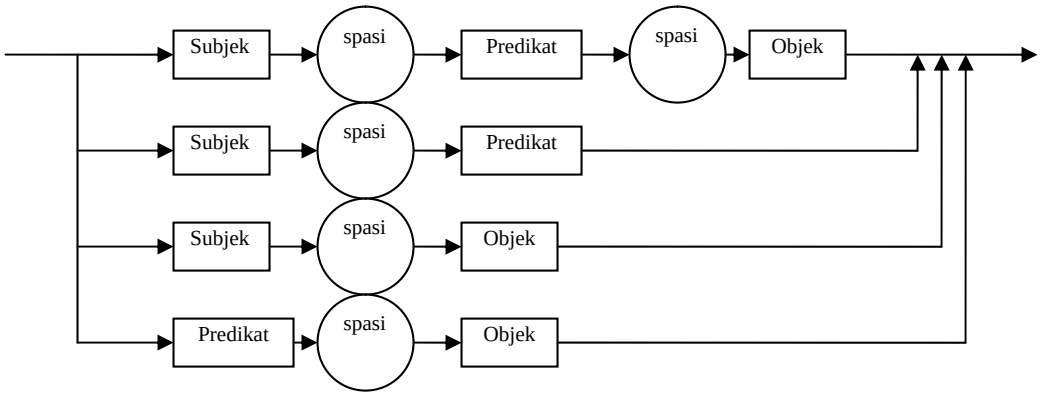
Setelah melakukan proses scanning dan mendapatkan hasil berupa token, maka token yang ada akan di konvert menjadi sebuah pohon parse dan dilihat apakah kata-kata tersebut termasuk kedalam aturan produksi yang ada. Pada proses ini kata yang berada di token tadi diambil untuk disesuaikan dengan aturannya dimana kata yang sudah mempunyai token tadi termasuk kedalam jenis kata yang mana apakah subjek, predikat, atau objek. Jika telah memenuhi tempatnya sebagai subjek atau prediakat atau objek maka akan diteruskan pada proses pencarian pada database.

<Kalimat> ::= <Subjek> spasi <Predikat> spasi <Objek>

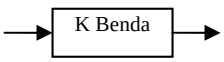
<Kalimat> ::= <Subjek> spasi <Predikat>

<Kalimat> ::= <Subjek> spasi <Objek>

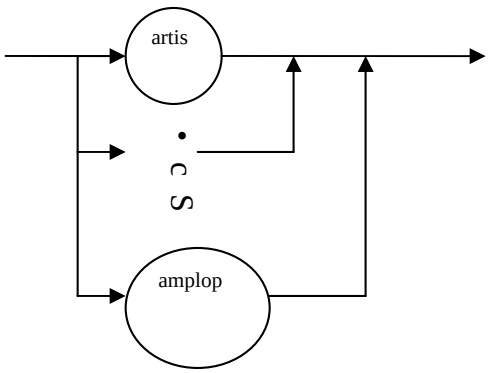
<Kalimat> ::= <Predikat> spasi <Objek>



Gambar 3.7. Diagram Sintaks Kalimat



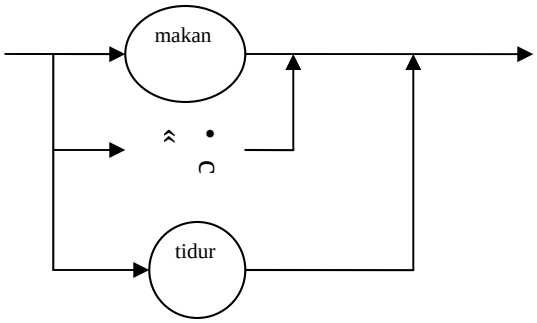
Gambar 3.8. Diagram Sintaks Subjek



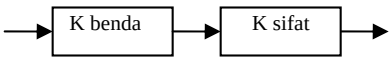
Gambar 3.9. Diagram Sintaks Kata Benda



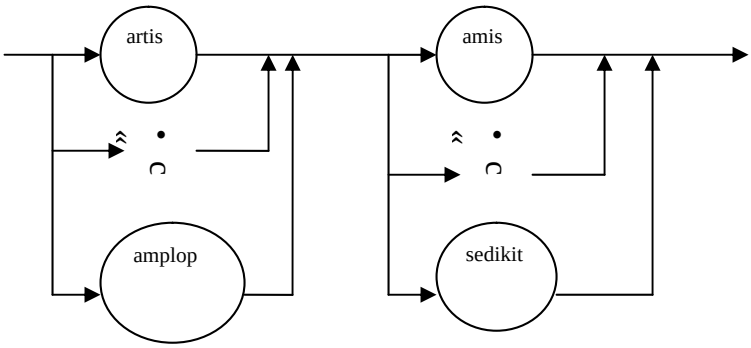
Gambar 3.10. Diagram Sintaks Predikat



Gambar 3.11. Diagram Sintaks Kata Kerja



Gambar 3.12. Diagram Sintaks Objek



Gambar 3.13. Diagram Sintaks K Benda dan K Sifat

3. Pencarian

   Didalam proses pencarian ini kata-kata tersebut akan dicari pada database yang ada, jika kata tidak ada dalam database maka kata masukannya yang akan ditampilkan pada keluaran.

4. Pengurutan

Didalam proses ini kalimat yang sudah ditentukan aturan produksinya akan di urutkan sesuai aturannya dalam hal ini urutannya adalah SOP (Subjek Predikat Objek).

3.6. Algoritma

3.5.1. Algoritma Scanning

```
if rs.eof do
        result:= false;
        ket:= NB
endif
else
        if rs.recordset.field[0].value =  1  then
                kalimat:= true;
        endif
        else
                kalimat:= false;
        result:= true;
        ket:=rs.recordset.fields[1].value;
        end
rs.close
endif
```

3.5.2. Algoritma Parsing

```
 if banyak_token>0 then
      begin
           banyak_kalimat:=token[banyak_token].KalimatKe;
           tokenke ← 0;
           cek_id   ← 1;
           for i←1 to banyak_kalimat do
           begin
                Ganti_Kalimat ←false;
                while ganti_kalimat=false do
                begin
                     tokenke←tokenke+1;
                     with token[tokenke] do
                     begin
                          if KalimatKe←i then
                          begin
                               if JenisKata =2 then
                               begin
                                    tspo←'Predikat';
                                    cek_id←tokenke;
                               end
                               else if JenisKata = 6 then
                                    tspo←''
                               else
                               begin
                                    if token[cek_id].TSPO ='Predikat' then
                                    begin
                                         tspo←'Objek';
                                         cek_id←tokenke;
                                    end
                                    else
                                    begin
                                         tspo←'Subjek';
                                         cek_id←tokenke;
                                    end;
                               end;
                          end
                          else
                               ganti_kalimat←true;
                     end;
```

```
            end;
        end;
    end;
```

3.5.3. Algoritma Pertukaran Objek Predikat

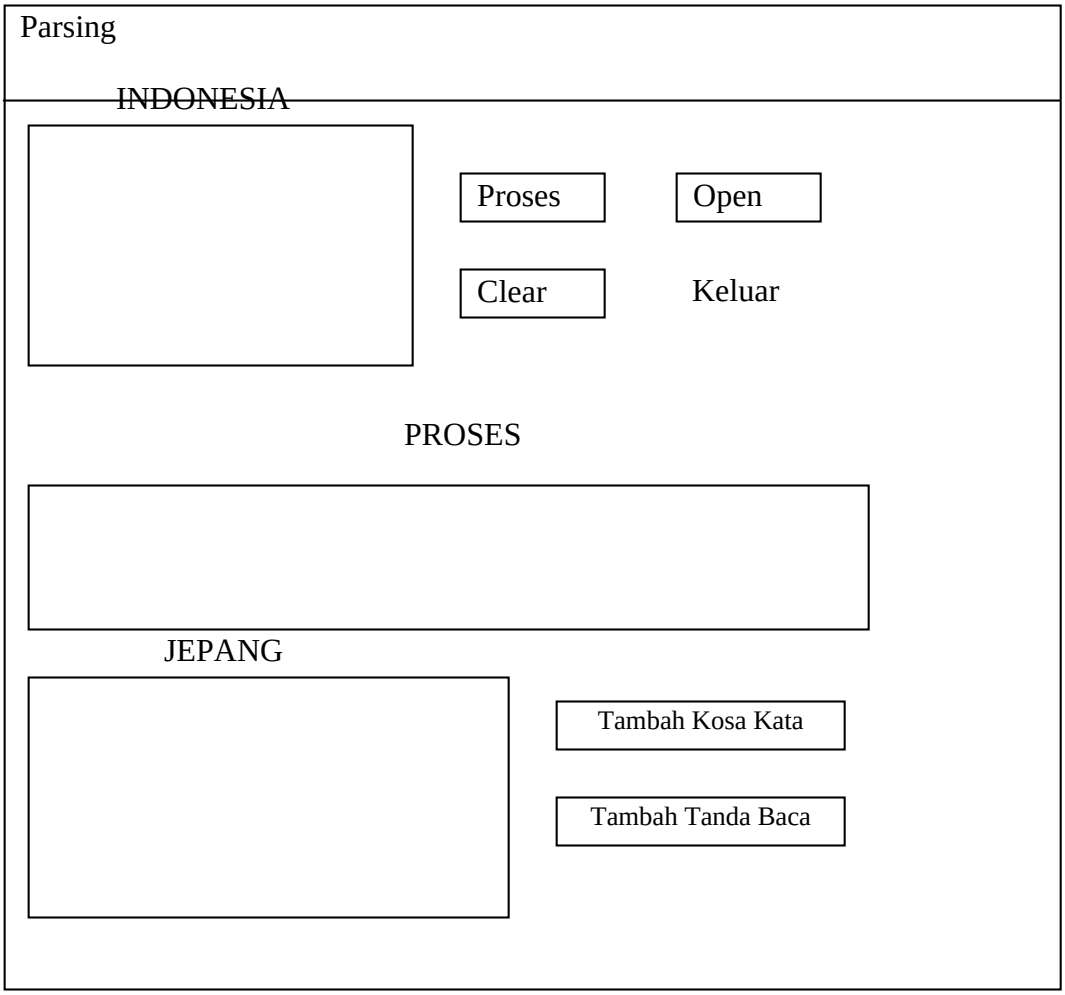
```
if    (token > 0 ) then
for i ← banyak_kalimat do { untuk kata lebih dari satu}
            Gantikal ← false { hanya satu kata}
            While gantikal = false do
                        Tokenke ← tokenke+1;
                        If (kalke = i ) then
                                    If (tSPO = Subjek) then
                                                SubjekJ ← SubjekJ + TJepang;
                                    endif
                                    Else if (tSPO = Predikat) then
                                                PredikatJ ← PredikatJ + Tjepang;
                                    Endif
                                    Else if (tSPO = Objek) then
                                                Objek← ObjekJ + Tjepang;
                                    Endif
                                    Else
                                    Tanda_baca ← tanda_baca + Tjepang
                                    End
                        Endif
                        Else
                        Gantikal ← true;                    {ganti kalimat}
                        Tokenke ← tokenke ? 1;
                        End
            Endwhile
Endfor
endif
```

3.7. Batasan Perancangan Sistem

Perancangan sistem kamus Indonesia-Jepang ini ada pada proses input dan keluarannya, dimana pada proses inpu kalimat yang dimasukan diusahakan menggunakan pola kalimat dasar yaitu SPO (Subjek Predikat Objek), jika tidak kalimat yang dihasilkan akan diurutkan sesuai dengan urutan pada proses masukan. Sedangkan pada proses keluaran kata-katanya masih dalam bentuk kamus dan tidak ada partikel yang mengikuti kata-kata tersebut.

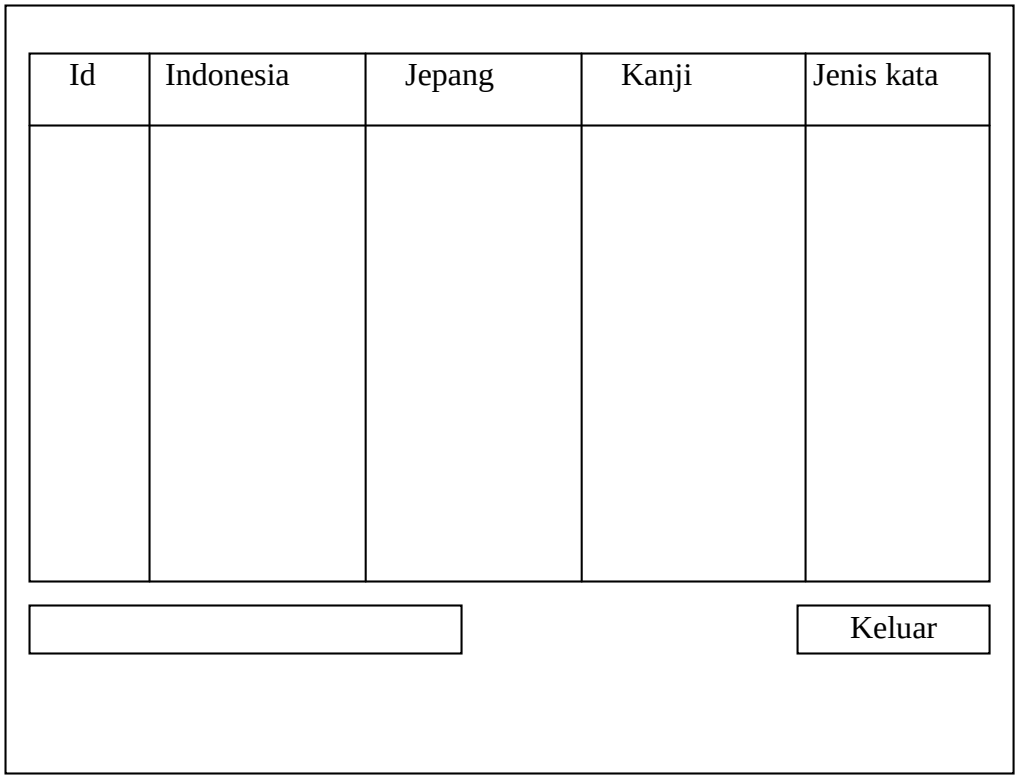
3.8. Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak ini adalah perancangan antar muka dengan pengguna. Antar muka tersebut antara lain adalah tampilan awal sistem dan tampilan detail hasil proses, tampilan edit dan pemasukan data ke dalam database.



Gambar 3.14. Tampilan Menu Utama

Pada gambar 3.14. diperlihatkan sebagai form utama dimana form ini akan menampilkan tampilan untuk proses masukan dimana kata dimasukan ke dalam form Indonesia kemudian dengan mengklik tombol proses maka hasil akan dikeluarkan pada form Jepang dan form Kanji masukan bisa juga dari file .txt dengan cara mengklik tombol open pada menu. Proses dapat dilihat dengan mengklik tombol Lihat proses. Pada gambar 3.15. ditampilkan menu edit database dan juga untuk penambahan database itu sendiri.



Gambar 3.15. Tampilan Menu Edit Database

BAB IV

IMPLEMENTASI SISTEM

4.1. Implementasi

4.1.1. Proses Masukan File

Tampilan awal dari sistem ini adalah seperti pada gambar 4.1.



Gambar 4.1. Tampilan Awal Sistem

Pada gambar 4.2. ditampilkan proses masukan dalam bahasa Indonesia.



Gambar 4.2. Proses Masukan File

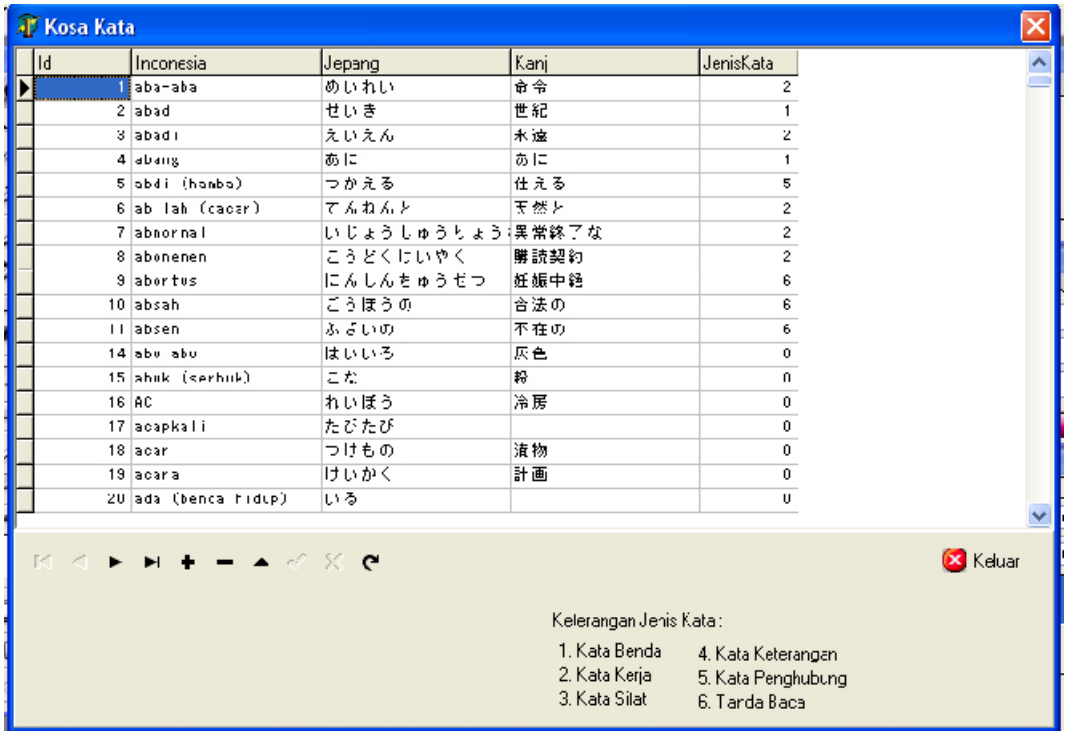
4.1.2 Proses Keluaran File

Pada proses ini ditampilkan hasil output dari file yang telah diproses setelah poses masukan file. Gambar 4.3. merupakan tampilan database dan kita dapat langsung menambahkan kata-kata yang belum ada dalam database tersebut.

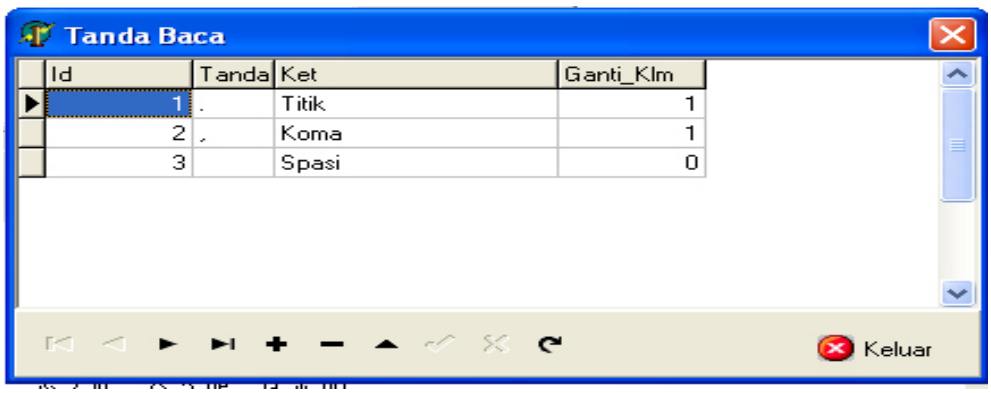
Gambar 4.4. merupakan tampilan dari tanda baca dari kalimat dimana tanda baca ini digunakan sebagai pemisah dari satu kata ke kata berikutnya.

Kalimat masukan akan diproses dan ditampilkan dalam proses keluaran sesuai dengan aturan yang ada seperti pada gambar 4.5..

Kita dapat membuka file dalam format .txt seperti pada gambar 4.6.



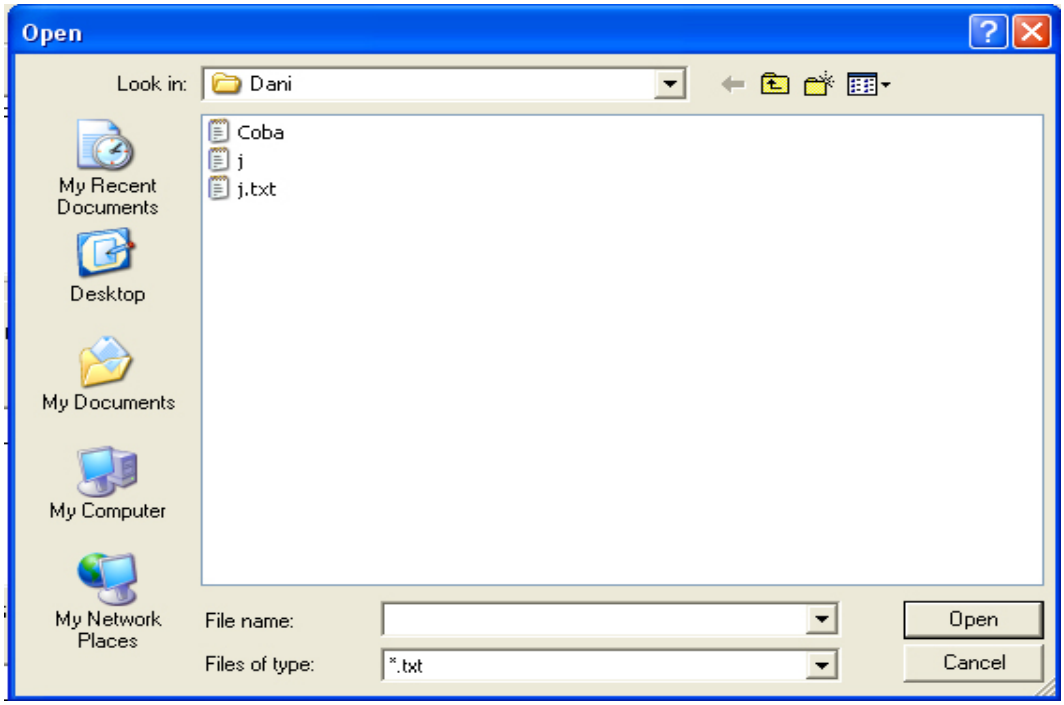
Gambar 4.3. Tampilan Database



Gambar 4.4. Tampilan Tanda Baca



Gambar 4.5. Hasil Proses



Gambar 4.6. Tampilan Open

4.2. Analisa Hasil

Didalam software yang dibuat ini pada proses pengurutan kata menggunakan aturan pola kalimat bahasa Jepang yaitu SOP (Subjek Objek Predikat) yang berlainan dengan aturan kalimat bahasa Indonesia yaitu SPO (Subjek Predikat Objek), aturan ini dimasukan pada saat hasil akan ditampilkan, dimana kata yang telah dipecah dimasukan kedalam token dan didalam token tersebut disusun berdasarkan jenis kalimatnya. Apabila kalimat tersebut mempunyai pola SPO maka secara otomatis akan dikeluarkan dalam bentuk SOP pada layar keluaran bahasa Jepang.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah mengamati kinerja dari perangkat lunak kamus ini, maka penulis dapat mengambil kesimpulan bahwa:

1. Kata-kata yang dihasilkan berupa kata kamus (jishoukei)
2. perangkat lunak ini dapat memudahkan dalam pencarian arti kata dalam bahasa jepang.

5.2. Saran

Pada hasil output kata yang dihasilkan tidak menggunakan partikel sehingga kata satu dengan kata yang lain dirasa berdiri sendiri oleh karena itu untuk kedepannya diharapkan ada partikel penghubung kata agar kata menjadi tidak terlalu rancu untuk dibaca.

Pada kata indonesia yang mempunyai dua kata dan dalam bahasa jepangnya satu kata akan ditampilkan perkata sehingga kata tersebut menjadi rancu, diharapkan nantinya kata-kata tersebut bisa memakai aturan-aturan yang baru sehingga bisa ditampilkan keluaran yang sebenarnya.

Daftar Pustaka

1. Hariyanto Bambang, 2000, Struktur Data, Informatika, Bandung.
2. Team Kashiko, 1999, Kamus lengkap Jepang-Indonesia Indonesia-Jepang, Kashiko, Surabaya.
3. Silberschatz, Korth and Sudarshan, 1997, Database System Consepts, McGraw-Hill Book Co, Singapore.
4. Ir. Inge Martina, 2002, 36 Jam Belajar Komputer Database Client/ Server Menggunakan Delphi, Elex Media Komputindo, Jakarta.
5. Firrar Utdirartatmo, 2001, Teknik Kompilasi, J & J Learning Yogyakarta, Yogyakarta.
6. Antony Pranata, 2002, Pemograman Borland Delphi 6 edisi 4, Andi Yogyakarta, Yogyakarta.
7. Abdul Kadir, 2001, Konsep dan Tuntunan Praktis Basis Data, Andi Yogyakarta, Yogyakarta.

LAMPIRAN

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, DB, ADODB,
ComCtrls, OleServer,
  Excel2000, Grids, jpeg;

type
  TFrmMain = class(TForm)
    Indonesia: TMemo;
    Jepang: TMemo;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    Proses: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    SpeedButton5: TSpeedButton;
    Label4: TLabel;
    SpeedButton7: TSpeedButton;
    Bevel1: TBevel;
    SpeedButton8: TSpeedButton;
    RS: TADOQuery;
    OpenDialog1: TOpenDialog;
    Bar: TProgressBar;
    SaveDialog1: TSaveDialog;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton6: TSpeedButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure ProsesClick(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
  private
    { Private declarations }
    Procedure Baca_Data(NamaFile : string);
    Procedure Simpan_Data(NamaFile :
string;Memo:TMemo);
    Procedure Init_Token;
    Function Cek_Tanda_Baca(Tanda : Char; var
GantiKalimat:boolean;var ket:string):boolean;
    Procedure Lihat_Kamus(Indonesia:String; var
Jepang,Kanji,ket:String; var Jenis_Kata:Integer);
    Procedure Tabel_Token(Kalimat : string);
  public
    { Public declarations }
  end;

  TokenX = record
    TIndonesia,
    TJepang,
    TKanji        : string;
    JenisKata   : integer;
    KalimatKe   : integer;
    Urutan        : integer;
    TKet           : string;
    TSPO           : string;
  end;


var
  FrmMain: TFrmMain;
  Token : Array [1..1000] of TokenX;
  banyak_Token,KalimatKeX,UrutanX:integer;
  SQL,Koneksi:String;
  FileX : Text;


implementation

uses Kata, TandaBaca, Proses ;

{$R *.dfm}

Procedure TFrmMain.Baca_Data(NamaFile : string);
var
    f:textfile;
    isidata:string;
begin
    Indonesia.Clear;
    assignfile(f,NamaFile);
    reset(f);
    while not eof(f) do
    begin
        readln(f,isidata);
        indonesia.Lines.Add(isidata);
    end;
    closefile(f);
end;

Procedure TFrmMain.Simpan_Data(NamaFile :
string;Memo:TMemo);
var
    f:textfile;
    i:integer;
begin
    assignfile(f,Namafile);
    rewrite(f);
    for i:= 0 to memo.Lines.Count do
        writeln(f,memo.Lines[i]);
    closefile(f);
end;

Procedure TFrmMain.Init_Token;
var
    i:integer;
begin
    banyak_Token:=0;
    for i := 1 to 1000 do
    begin
        with token[i] do
        begin
            TIndonesia   :='';
            TJepang       :='';
            Tkanji         :='';
            JenisKata     :=0;
            KalimatKe     :=0;
            urutan         :=0;
            TKet            :='';
            {TSPO           :='';}
        end;
    end;
end;
```

```
Function TFrmMain.Cek_Tanda_Baca(Tanda : Char; var
GantiKalimat:boolean;var ket:string):boolean;
begin
      sql:='select Ganti_Klm,ket from t_tandabaca where
tanda='" + tanda + "";
      rs.ConnectionString :=Koneksi;
      rs.SQL.Clear;
      rs.SQL.Add(sql);
      rs.Prepared;
      rs.Open;

      if rs.Eof then
      begin
            result:=false;
            ket:='-'
      end
      else
      begin
            if rs.Recordset.Fields[0].Value = '1' then
                  GantiKalimat:=true
            else
                  GantiKalimat:=false;
            result:=true;
            ket:=rs.Recordset.Fields[1].Value;
      end;
      rs.Close;
end;

Procedure TFrmMain.Lihat_Kamus(Indonesia:String; var
Jepang,Kanji,ket:String; var Jenis_Kata:Integer);
begin
      sql:='select Jepang,Kanji,JenisKata from t_Kosakata
where Indonesia='" + Indonesia + "";
      rs.ConnectionString :=Koneksi;
      rs.SQL.Clear;
      rs.SQL.Add(sql);
      rs.Prepared;
      rs.Open;
      if rs.Eof then
      begin
            Jepang:=Indonesia;
            Kanji:=Indonesia;
            Jenis_Kata:=0;
            Ket:='Belum ada dikamus';
      end
      else
      begin
            Jepang:=rs.Recordset.Fields[0].Value;
            Kanji:=rs.Recordset.Fields[1].Value;
            Jenis_Kata:=rs.Recordset.Fields[2].Value;
            Ket:='-';
      end;
      rs.Close;
end;

Procedure TFrmMain.Tabel_Token(Kalimat : string);
var

Pjg_Kalimat,i,OldKalimatKe,banyak_kalimat,tokenke:Integer;
      Kata,ket,OldSPO    : string;
      Ganti_Kalimat :boolean;
      cek_id : integer;
begin
      Pjg_Kalimat:=length(kalimat);
      kata:='';
      for i:= 1 to Pjg_kalimat do
      begin
            Ganti_Kalimat:=false;
            if cek_Tanda_Baca(kalimat[i],Ganti_Kalimat,ket)
then //true berarti itu adalah tanda baca
            begin
                  // urutanke dalam kalimat
                  // Masukin kata ke dalam tabel token
                  if kata<>'' then
                  begin
                        banyak_token:=banyak_token+1;
                        UrutanX:=UrutanX+1;
                        with token[banyak_token] do
                        begin
                              Tindonesia := kata;
                              kata:='';

Lihat_Kamus(TIndonesia,TJepang,TKanji,TKet,JenisKata);
                              Urutan:=urutanX;
                              KalimatKe:=KalimatKeX;
                        end;
                  end;

                  // Masukin tanda baca ke dalam tabel token
                  banyak_token:=banyak_token+1;
                  UrutanX:=UrutanX+1;
                  with token[banyak_token] do
                  begin
                        Tindonesia := kalimat[i];

Lihat_Kamus(TIndonesia,TJepang,TKanji,TKet,JenisKata);
                        TKet:=ket;
                        Urutan:=urutanX;
                        KalimatKe:=KalimatKeX;
                  end;


                  if Ganti_Kalimat = true then
                  begin
                        KalimatKeX:=KalimatKeX+1; // tentuin
kalimat ke
                        UrutanX:=0;
                  end;

            end
            else
            begin
                  kata:=kata+kalimat[i];
                  if i=pjg_kalimat then
                  begin
                        banyak_token:=banyak_token+1;
                        UrutanX:=UrutanX+1;
                        with token[banyak_token] do
                        begin
                              Tindonesia := kata;

Lihat_Kamus(TIndonesia,TJepang,TKanji,TKet,JenisKata);
                              Urutan:=urutanX;
                              KalimatKe:=KalimatKeX;
                        end;
                  end;
            end;
      end;

end;

procedure TFrmMain.FormCreate(Sender: TObject);
begin
      koneksi:='Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=db.mdb';
      Indonesia.Clear;
      Jepang.Clear;
      {Kanji.Clear; }
end;

procedure TFrmMain.SpeedButton5Click(Sender: TObject);
begin
      FrmKosaKata:=TFrmKosaKata.Create(self);
      FrmKosaKata.ShowModal;
      FrmKosaKata.Free;
end;
```

```
procedure TFrmMain.SpeedButton8Click(Sender: TObject);
begin
      FrmTandaBaca:=TFrmTandaBaca.Create(self);
      FrmTandaBaca.ShowModal;
      FrmTandaBaca.Free;
end;

procedure TFrmMain.ProsesClick(Sender: TObject);
var
      banyak_kalimat,tokenke,cek_id,i:integer;

SubjekJ,PredikatJ,ObjekJ,SubjekK,PredikatK,ObjekK,OldSP
O :string;
      Ganti_Kalimat :boolean;
      Tanda_BacaJ, Tanda_BacaK:string;
begin
      screen.Cursor := crHourGlass;
      init_Token;
      KalimatKeX:=1;
      UrutanX:=0;
      bar.Min := 0;
      bar.max := indonesia.Lines.Count;
      bar.Position :=0;

      for i:= 0 to indonesia.Lines.Count -1 do
      begin
            tabel_Token(indonesia.Lines.Strings[i]);
            bar.Position := bar.Position + 1;
      end;

      jepang.Lines.Clear;
      {Kanji.Lines.Clear; }

      if banyak_token>0 then
      begin
            banyak_kalimat:=token[banyak_token].KalimatKe;
            tokenke        :=0;
            cek_id         :=1; // index untuk token SPO
sebelumya
            for i:= 1 to banyak_kalimat do
            begin
                  Ganti_Kalimat :=false;
                  while ganti_kalimat=false do
                  begin
                        tokenke:=tokenke+1;
                        with token[tokenke] do
                        begin
                              //if TIndonesia <> '13'
                              if KalimatKe = i then
                              begin
                                    if JenisKata = 2 then
                                    begin
                                          tspo:='Predikat';
                                          cek_id:=tokenke;
                                    end
                                    else if JenisKata = 6 then
                                          tspo:=''
                                    else
                                    begin
                                          if token[cek_id].TSPO
='Predikat' then
                                          begin
                                                tspo:='Objek';
                                                cek_id:=tokenke;
                                          end
                                          else
                                          begin
                                                tspo:='Subjek';
                                                cek_id:=tokenke;
                                          end;
                                    end
                              end
                              else
                                    ganti_kalimat:=true;
                        end;
                  end;
            end;

      // ubah struktur SPO menjadi SOP

      if banyak_token>0 then
      begin
            banyak_kalimat:=token[banyak_token].KalimatKe;
            tokenke        :=0;
            Tanda_BacaJ:='';
            Tanda_BacaK:='';
            for i:= 1 to banyak_kalimat do
            begin
                  SubjekJ        :='';
                  PredikatJ      :='';
                  ObjekJ         :='';
                  SubjekK        :='';
                  PredikatK      :='';
                  ObjekK         :='';

                  Ganti_Kalimat :=false;
                  while ganti_kalimat=false do
                  begin
                        tokenke:=tokenke+1;
                        with token[tokenke] do
                        begin
                              if KalimatKe = i then
                              begin
                                    if TSPO = 'Subjek' then
                                    begin
                                          SubjekJ:=Tanda_BacaJ +
SubjekJ + TJepang + ' ';
                                          SubjekK:=Tanda_BacaK +
SubjekK + TJepang + ' ';
                                          Tanda_BacaJ:='';
                                          Tanda_BacaK:='';
                                    end
                                    else if TSPO = 'Predikat' then
                                    begin
                                          PredikatJ:=Tanda_BacaJ +
PredikatJ + TJepang + ' ';
                                          PredikatK:=Tanda_BacaK
+ PredikatK + TJepang + ' ';
                                          Tanda_BacaJ:='';
                                          Tanda_BacaK:='';
                                    end
                                    else if TSPO = 'Objek' then
                                    begin
                                          ObjekJ:=Tanda_BacaJ +
ObjekJ + TJepang + ' ';
                                          ObjekK:=Tanda_BacaK +
ObjekK + TJepang + ' ';
                                          Tanda_BacaJ:='';
                                          Tanda_BacaK:='';
                                    end
                                    else
                                    begin

Tanda_BacaJ:=Tanda_BacaJ + TJepang;

Tanda_BacaK:=Tanda_BacaK + TKanji;
                                    end;
                              end
                              else
                              begin
                                    ganti_kalimat:=true;
                                    tokenke:=tokenke-1;
                              end;
                        end;
```

```
                end;
                jepang.Lines.Strings[0] :=
jepang.Lines.Strings[0]+ SubjekJ + ObjekJ + PredikatJ;
                {kanji.Lines.Strings[0]    :=
kanji.Lines.Strings[0] + SubjekK + ObjekK + PredikatK;}
            end;
        end;
        screen.Cursor := crDefault;
end;

procedure TFrmMain.SpeedButton6Click(Sender: TObject);
var
        i:integer;
begin
        frmProses:=TfrmProses.Create(self);

        with frmProses.Grid do
        begin
            RowCount :=banyak_Token+1;
            for i := 1 to banyak_Token do
            begin
                Cells[0,i]:=inttostr(i);
                Cells[1,i]:=Token[i].TIndonesia;
                Cells[2,i]:=Token[i].TJepang;
                Cells[3,i]:=Token[i].TKanji;
                Cells[4,i]:=inttostr(Token[i].JenisKata);
                Cells[5,i]:=inttostr(Token[i].KalimatKe);
                Cells[6,i]:=inttostr(Token[i].Urutan);
                Cells[7,i]:=Token[i].TKet;
                {Cells[8,i]:=Token[i].TSPO;}
            end;
        end;
        frmProses.ShowModal;
        frmProses.Free;
end;

procedure TFrmMain.SpeedButton1Click(Sender: TObject);
var
        lokasi:string;
begin
        if opendialog1.Execute then
        begin
            lokasi:=opendialog1.FileName;
            baca_data(lokasi);
        end;
        bar.Position :=0;
end;

procedure TFrmMain.SpeedButton3Click(Sender: TObject);
begin
        indonesia.Clear;
        jepang.Clear;
        {kanji.Clear;}
        bar.Position := 0;
end;

procedure TFrmMain.SpeedButton7Click(Sender: TObject);
begin
        application.Terminate;
end;
end.
uses Kata;
```