

BAB 2 LANDASAN TEORI

Dalam bab ini akan dipaparkan landasan-landasan teori yang telah ada dan menjadi pijakan dalam pelaksanaan penelitian ini.

2.1 Morfologi

Sebelum melihat lebih jauh tentang pengurai morfologi, akan diceritakan lebih dulu tentang morfologi itu sendiri. Morfologi merupakan pengetahuan akan pembentukan suatu kata dari unit-unit lebih kecil yang mengandung makna. Unit-unit terkecil dan mengandung makna itu sendiri disebut juga morfem. Morfem dapat dibagi menjadi kata dasar dan imbuhan (afiks) di mana selanjutnya imbuhan itu dapat dibagi lagi menjadi prefiks, sufiks, infiks, dan konfiks. Penggabungan imbuhan dan kata dasar disebut *concatenative morphology* dikarenakan penggabungan tersebut akan menghasilkan suatu kata yang merupakan hasil penggabungan dua atau lebih morfem yang dikonkatenasikan secara bersama-sama [JURA00]. Ada lagi bentuk lebih kompleks yang disebut *nonconcatenative morphology* yang akan dijelaskan pada subbab 2.4.

Proses morfologi dapat dibagi menjadi dua macam menurut pembentukan kelas kata yang dihasilkan. Dua macam morfologi itu antara lain:

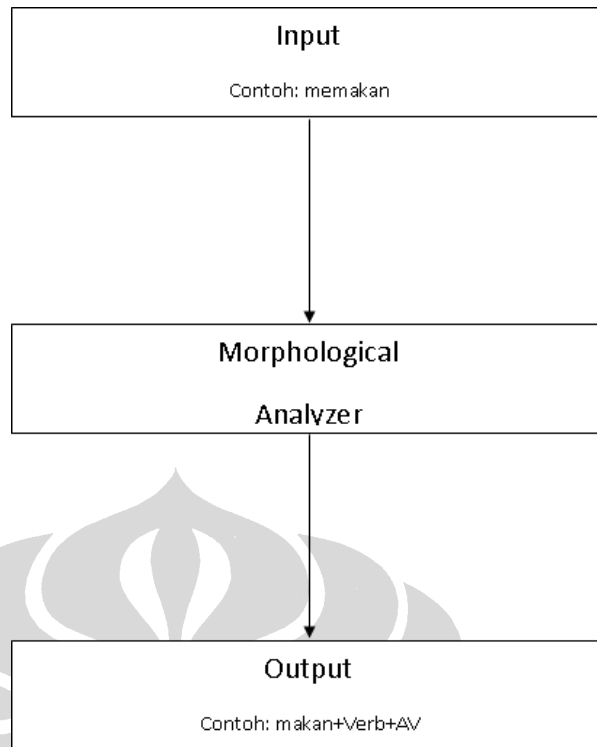
1. Morfologi infleksional: merupakan pembentukan yang menghasilkan kata dengan kelas kata yang sama dengan kelas kata dari kata dasar pembentuknya. Ciri-ciri dari morfologi infleksional yaitu sistematis dalam artian polanya teratur dan memiliki maksud dan hasil yang jelas serta produktif dalam artian dapat diterapkan pada semua kata dengan kelas kata yang sesuai. Contoh: “dibaca” yang merupakan kata kerja dihasilkan dari kata dasar “baca” yang juga merupakan kata kerja.

2. Morfologi derivasional: merupakan pembentukan yang menghasilkan kata dengan kelas kata yang berbeda dengan kelas kata dari kata dasar pembentuknya. Kebalikan dari infleksional, morfologi derivasional ini justru memiliki ciri-ciri tidak sistematis dan tidak produktif. Contoh: “pembaca” yang merupakan kata benda dihasilkan dari kata dasar “baca” yang merupakan kata kerja.

Sebagian besar kata-kata dalam bahasa Indonesia dalam proses pembentukan kata-katanya tergolong dalam morfologi derivasional. Kelas kata dari kata dasar banyak yang berbeda dengan kelas kata setelah diturunkan. Bisa dikatakan hanya imbuhan di-, ber-an, ter-, kata ulang sejati tanpa imbuhan, dan sebagian besar kata ulang sebagian serta berimbuhan saja yang murni tergolong morfologi infleksional. Untuk variasi imbuhan lainnya bisa dikatakan tergolong morfologi derivasional. Sebagai contoh imbuhan meN- untuk kata dasar yang tergolong kata kerja memang tidak akan mengalami perubahan kelas kata saat diturunkan, akan tetapi untuk kata dasar nomina dan adjektiva ia akan mengalami perubahan kelas kata.

2.2 Pengurai Morfologi

Proses yang dilakukan pengurai morfologi sendiri adalah melakukan pengenalan kata yang membagi kata menjadi satu atau lebih morfem [JURA00]. Kata tersebut dikenali dengan melakukan suatu metode untuk menganalisis morfologi pada kata tersebut. Dari *input* suatu kata akan diuraikan menjadi *output* berupa kata dasar pembentuknya beserta fitur-fitur morfologi yang berkesesuaian. Ilustrasinya dapat dilihat pada sebuah contoh pada gambar 2-1.



Gambar 2-1. Contoh Ilustrasi Skema Pengurai Morfologi

Ilustrasi di atas menunjukkan fungsi suatu pengurai morfologi (*morphological analyzer*). Dari input berupa kata “memakan”, dilakukan proses penguraian morfologi sehingga dapat dikeluarkan hasil berupa kata “makan” yang merupakan kata dasar pembentuk kata “memakan” dan juga *tags* yang sesuai dengan input tersebut. Pada contoh di atas, hasil keluaran berupa kata “makan”, *tag* “+Verb”, dan *tag* “+AV” mengandung penjelasan morfologi bahwa kata “memakan” itu didapatkan dari kata dasar “makan” yang dijadikan bentuk verba dan merupakan kata kerja aktif (*Active Voice*). Penjelasan lebih lanjut tentang pemberian *tags* dapat ditemukan pada Bab 3.

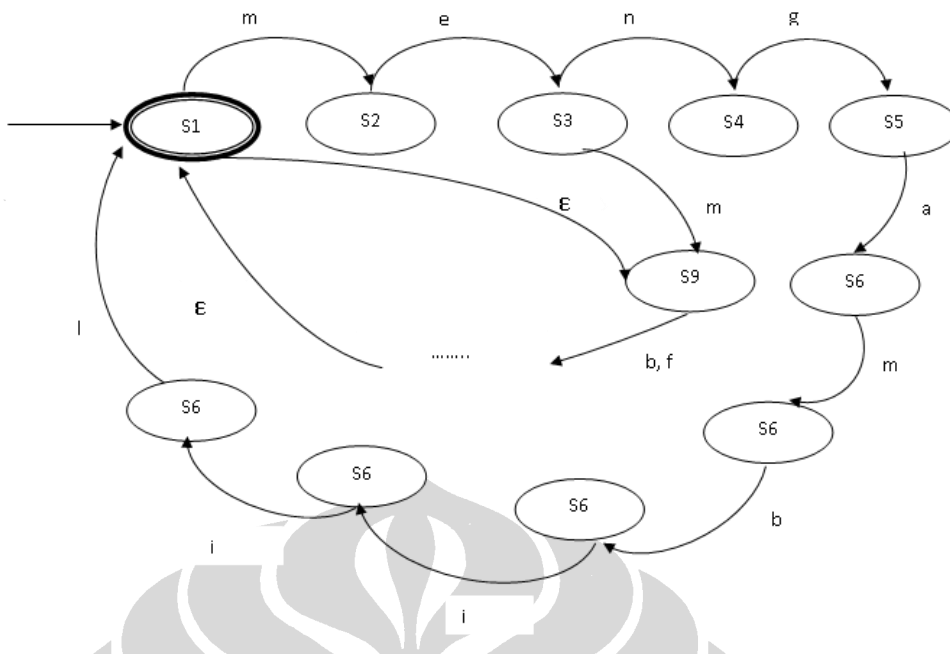
Untuk membangun suatu pengurai morfologi dibutuhkan:

1. *Lexicon*: suatu daftar yang memuat semua kata dasar, imbuhan, dan semua informasi yang dibutuhkan mengenai kata dasar dan imbuhan tersebut.

2. *Morphotactics*: semua aturan yang mengatur urutan penggabungan morfem pada suatu kata. Contoh: imbuhan “meN” jika diletakkan di depan kata dasar “makan” dari kelas kata kerja akan menghasilkan kata “memakan” yang merupakan kata kerja. Aturan ini juga memastikan kata dasar “makan” dengan imbuhan “meN” tidak menghasilkan kata “makanme”.
3. *Orthographics*: semua aturan yang mengatur perubahan yang terjadi sewaktu morfem-morfem tersebut digabungkan. *Orthographics* sering juga disebut *morphophonemics*. Untuk pembahasan pada bab-bab selanjutnya akan digunakan kata morfofonemik (*morphophonemic*) untuk memaparkan aturan-aturan *orthographic* ini. Contoh: imbuhan “meN” digabungkan dengan kata dasar “baca” akan menghasilkan kata “membaca”.

2.3 Finite-State Transducers

Pada awalnya, pemodelan pengurai morfologi menggunakan *Finite-State Automaton* (FSA) dengan *simple-network* biasa [JURA00]. FSA adalah suatu model komputasi yang terdiri dari himpunan *states*, *start state*, *accepting state*, simbol *input*, dan fungsi transisi yang memetakan dari simbol *input* dan *state* saat ini ke *state* selanjutnya. Dalam pengurai morfologi, pada awalnya FSA digunakan untuk menerima kata yang *valid* berdasarkan aturan-aturan yang diterapkan. FSA dibentuk untuk memutuskan bisa atau tidaknya suatu kata di-*recognize* sebagai suatu kata yang *valid*. Contoh penggunaan FSA untuk me-*recognize* sebagian aturan dalam bahasa Indonesia dapat dilihat pada gambar 2-2.



Gambar 2-2. Contoh Penggunaan FSA

Seperti dapat dilihat pada gambar 2-1, FSA tersebut akan menerima kata-kata yang berawalan fonem “b” ataupun “f” baik dengan imbuhan “mem” maupun tanpa imbuhan “mem”. Dengan FSA ini kata dasar dengan fonem awal “b” dan “p” maupun penggabungannya dengan imbuhan “mem” akan dianggap sebagai suatu kata yang *valid*. Akan tetapi seperti kita lihat di atas, untuk membuat kata “mengambil” perlu dibuat alur sendiri. FSA disini tidak dapat melihat *regularity* antara “meng” dengan “mem”. Ukuran *network* yang dibuat akan meledak dengan cara seperti ini. Setiap penambahan variasi penggabungan imbuhan dengan kata dasar harus didefinisikan dalam FSA, sehingga ukuran *network* akan menjadi sangat besar.

Dengan FSA, kita memasukkan semua kemungkinan pengimbuhan sebagai suatu “*sub-lexicon*” yang masing-masing dimasukkan dalam FSA. Tiap *lexicon* didefinisikan dapat bergabung dengan imbuhan apa saja, kemudian barulah dimasukkan tiap kata seperti yang ditunjukkan pada gambar 2-1.

Akan tetapi dengan cara tersebut ditemukan beberapa masalah di antaranya[JURA00]:

1. Pengurai morfologi seharusnya dapat juga mengenali pola atau sifat pembentukan suatu kata yang tidak dapat ditangani dengan FSA biasa. FSA hanya bisa me-*recognize* kata tersebut *valid* atau tidak. Untuk pengurai morfologi butuh ditampilkan analisis morfologi seperti yang ditunjukkan pada gambar 2.1. Hal seperti itu sulit dilakukan oleh FSA biasa.
2. Pengurai morfologi juga harus bisa menangani variasi ejaan seperti yang terdapat pada *orthographics*. Hal tersebut juga sulit diakomodasikan dengan FSA.

Kemudian pada tahun 1960, Noam Chomsky dan Moris Halle memformulasikan *phonological grammars* yang terdiri dari rangkaian berurutan dari *rewrite rules* [KART01]. *Rewrite rules* ini mengkonversikan representasi abstrak fonologi ke dalam bentuk *surface* melalui beberapa representasi menengah. Bentuk umum dari *rewrite rules* ini berbentuk: $a \rightarrow b/c_d$. Aturan tersebut dapat dibaca sebagai berikut: “a dapat diganti menjadi b jika terletak antara c dan d”.

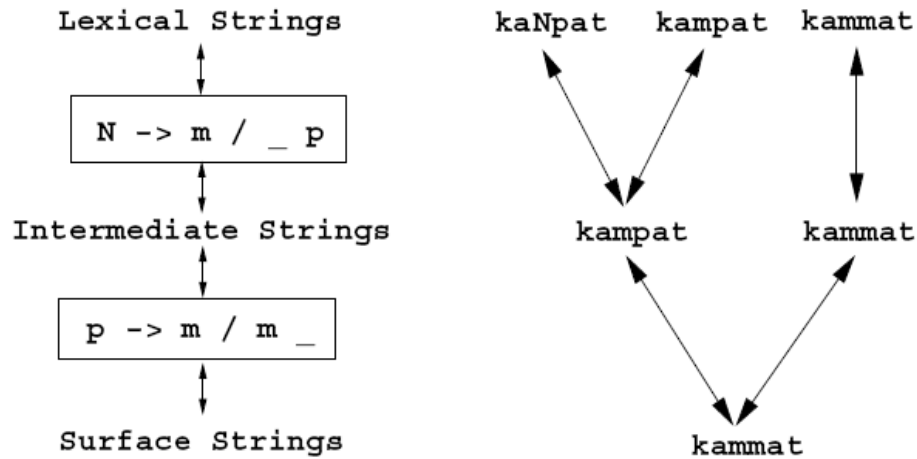
Dari ide awal *rewrite rules* ini akhirnya ditemukan bahwa *rewrite rules* tersebut bisa direpresentasikan ke dalam bentuk *Finite-State Transducer* (FST). FST sendiri adalah sebuah *Finite-State Machine* (FSM) yang memiliki dua pita. Kegunaan dua pita ini nantinya juga akan dimanfaatkan pada konsep morfologi dua tingkat yang akan dijelaskan pada subbab 2.3.

Sebuah *transducer* akan memetakan pita input ke dalam pita *output* dengan menerima isi dari pita *input* kemudian memprosesnya ke dalam pita *output* dan juga sebaliknya. Namun, ditemukan adanya permasalahan yang terjadi pada proses berlawanan arah. Berikut contoh yang terjadi pada morfologi bahasa Finlandia yang telah diamati oleh Lauri Karttunen pada saat mengembangkan pengurai morfologi bahasa Finlandia [KART01]. Pada contoh tersebut diperlihatkan suatu contoh kasus yang terjadi pada dua *rewrite rules* berurutan:

$N \rightarrow m_p$

p->m/m_

Transducer tersebut akan memetakan “kaNpat” menjadi “kammat” dengan “kampat” sebagai representasi menengah. Akan tetapi apabila dilakukan proses sebaliknya dengan memasukkan “kammat” sebagai *input* akan diperoleh tiga hasil *output* seperti pada gambar 2-3.

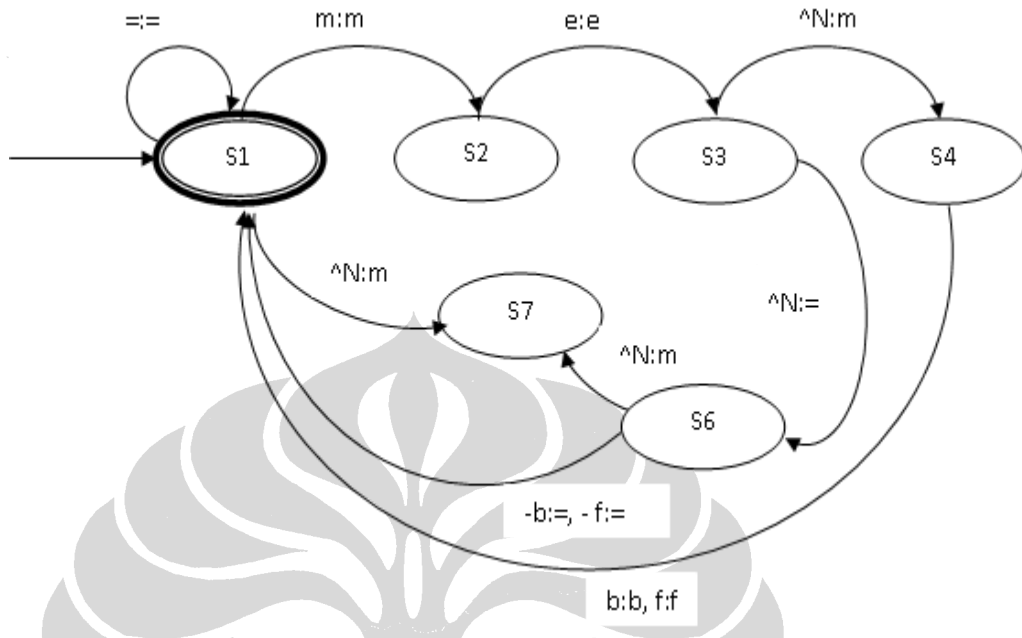


Gambar 2-3. Tiga Hasil Keluaran “kammat”

Jumlah kemungkinan hasil ini akan terus bertambah seiring dengan bertambahnya jumlah aturan yang terlibat. Inilah yang disebut “*overanalysis*” [KART01], di mana hasil-hasil tersebut akan tidak *valid* atau tidak ada dalam bahasa. Mengimplementasikan operasi *compose* untuk semua aturan *transducers* tidak akan mampu menyelesaikan masalah ini. Untuk mengatasi masalah ini haruslah dengan cara meng-*compose lexicon* dengan semua aturan yang ada. Dengan cara seperti ini, hasil yang tidak *valid* tidak akan muncul karena sudah dicegah pada pemeriksaan di *lexicon*-nya.

Pemanfaatan *transducers* untuk pengurai morfologi ini terletak pada aturan-aturan morfofonemik dan morfotaktiknya untuk menerima *input* yang sah, mengenali variasi ejaan, dan menangani pola dan sifat pembentukan suatu kata. Contoh sederhana suatu *transducer* yang dipakai pada pengurai morfologi untuk

menangani salah satu kasus variasi ejaan akibat pemberian imbuhan dapat dilihat pada gambar 2-4.



Gambar 2-4. Contoh Penggunaan *Transducer*

Contoh *transducer* di atas akan menangani salah satu kasus variasi ejaan yang terjadi pada morfologi bahasa Indonesia. Pada transisi S3 menuju S4, ada sebuah *replacement rule* yang akan menuliskan “m” pada pita bawah ketika mendapatkan “^N” pada pita atas. Jalur utama yang diharapkan adalah S1-S2-S3-S4-S1 yang akan menangani kasus perubahan “^N” menjadi “m” bila diikuti oleh fonem “b” ataupun “f”. S6 bertujuan untuk meloloskan kasus-kasus lain yang tidak termasuk kasus yang harus ditangani aturan di atas yaitu kasus-kasus dimana “^N” bertemu dengan fonem selain fonem “b” ataupun “f”. S7 berfungsi sebagai *state* buangan yang akan membuang *replacement rule* yang dipakai pada *state* ini agar kasus-kasus yang tidak diharapkan tidak diterima. Dengan cara seperti ini FST dapat menangkap *regularity* yang sebelumnya tidak dapat dilakukan oleh FSA. Untuk kasus lainnya dapat digunakan *replacement rule* “^N” dengan fonem yang sesuai untuk penggabungan dengan kata dasarnya.

2.4 Morfologi Dua Tingkat (*Two-Level Morphology*)

Pada tahun 1983, Koskenniemi membentuk implementasi dari model *constraint-based* yang tidak tergantung pada aturan *compiler*, penggabungan atau algoritma *finite state* di mana setiap aturan memiliki dua bentuk yaitu *lexical* dan *surface* yang kemudian aturan-aturan tersebut dijalankan secara paralel [KART01]. Hal ini kemudian disebut dengan Morfologi Dua Tingkat (*Two-Level Morphology*) [KART01]. Setiap aturan mendefinisikan hubungan antara *lexical* dan *surface* dengan lingkungan di mana bentuk hubungan tersebut diperbolehkan, dibutuhkan, atau dilarang. Morfologi dua tingkat ini didasarkan atas tiga prinsip, antara lain:

1. Aturan adalah pemetaan dari *constraints symbol* ke *symbol* dilakukan secara paralel, bukan secara berurutan.
2. *Constraints* bisa merujuk pada *lexical context*, *surface context*, atau keduanya secara bersamaan.
3. *Lexical lookup* dan *morphological analysis* dilakukan secara berurutan dan bersamaan.

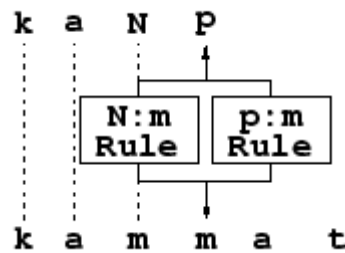
Untuk ilustrasi dua aturan pertama, lihat lagi contoh kasus “kaNpat”. Gambaran aturan dua tingkat untuk contoh tersebut pada hubungan *lexical-surface* dapat dilihat pada gambar 2-5.



Gambar 2-5. Ilustrasi Contoh Hubungan *Lexical-Surface* pada Aturan Dua Tingkat

Pada contoh di atas ditunjukkan pemetaan setiap simbol pada pita *lexical* dengan kata “kaNpat” dengan setiap simbol pada pita *surface* dengan kata “kammata”. Pasangan “N:m” dibatasi *constraint* pada lingkungan yang diikuti

symbol “p” pada sisi *lexical*. Pasangan “p:m” juga dibutuhkan setelah *symbol* “m” pada sisi *surface*. Kedua *constraints* berdiri bebas satu sama lain dan perilaku paralel dari morfologi dua tingkat menyebabkan hasil yang sama dengan dua aturan yang diganti berurutan. Aplikasi paralel dari morfologi dua tingkat ini dapat dilihat sebagai berikut pada Gambar 2-6.



Gambar 2-6. Ilustrasi Contoh Aplikasi Paralel Morfologi Dua Tingkat

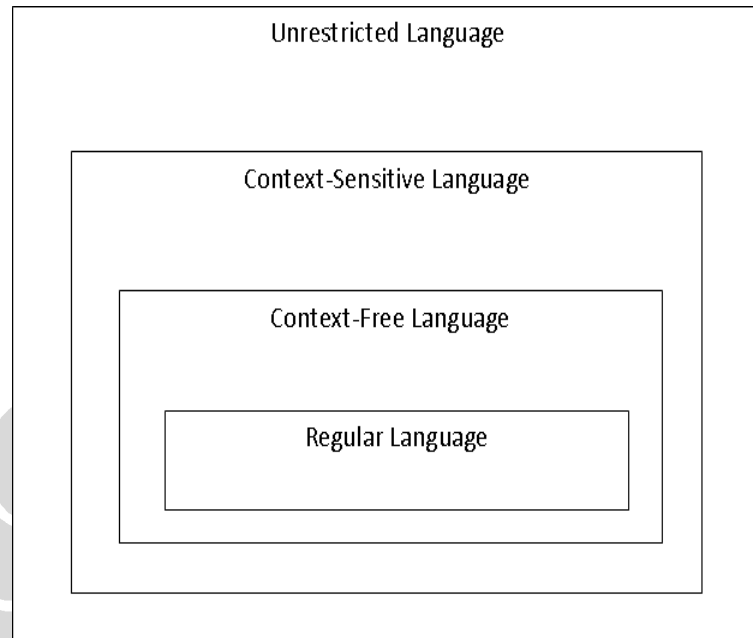
Penggunaan penggabungan aturan secara paralel ini tidak menyelesaikan masalah *overanalysis*, akan tetapi dengan morfologi dua tingkat ini masalah *overanalysis* menjadi lebih mudah diatur. *Lexicon* dapat diatur sebagai penyaring *lexical* sehingga *overanalysis* dapat diatasi. Proses analisa hanya melakukan fungsinya saja yaitu pemetaan simbol-simbol tanpa memikirkan masalah *overanalysis*.

2.5 Nonconcatenative Morphology

Berbeda dengan *concatenative morphology* yang memiliki definisi yang lebih jelas, *nonconcatenative morphology* lebih merupakan properti yang menggunakan operasi morfologi yang mengakibatkan tidak dapat dianalisa dengan metode parsial rekurensi [MCCA81]. Untuk mendapatkan analisa morfologi suatu kata yang tergolong dalam *nonconcatenative morphology*, tidak bisa dilakukan dengan konkatenasi secara berurutan dan bersama-sama. Contoh perilaku *nonconcatenative morphology* dalam bahasa Indonesia adalah kata ulang dan sisipan. Kedua perilaku ini tidak dapat secara langsung diimplementasikan ke dalam FST dikarenakan bentuknya yang bukan merupakan *regular language*.

Berikut ini merupakan penjelasan singkat tentang *regular language*. *Regular language* adalah bahasa yang dapat dibangkitkan dari bahasa-bahasa

berunsur tunggal dengan hanya operasi-operasi *union*, konkatenasi, dan *Kleene's Star*. *Regular language* dapat diterima oleh FSA. Dalam Chomsky hierarchy, posisi regular grammar dapat dilihat seperti pada Gambar 2-7.

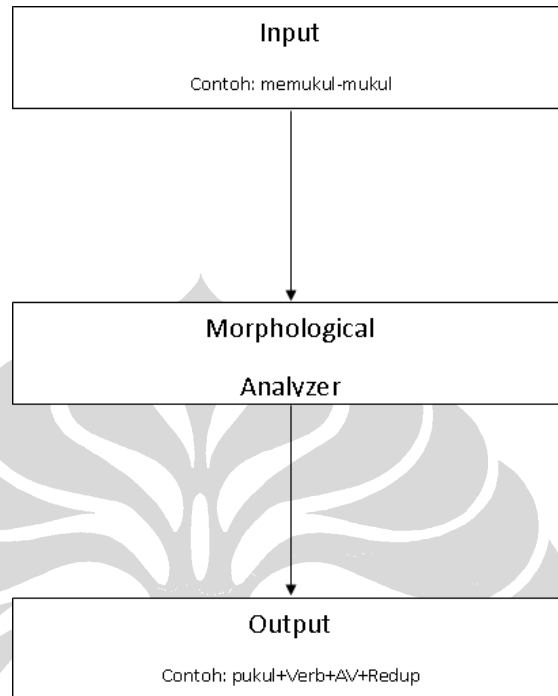


Gambar 2-7. Chomsky Hierarchy

Regular language berada di bagian paling dalam pada Chomsky Hierarchy, sedangkan perilaku *nonconcatenative morphology* berada di luar cakupan bagian *regular language*. [JURA00]. Dari sifat-sifat *regular language* akan terlihat bahwa *nonconcatenative morphology* bukanlah berada di tingkatan *regular language* akan tetapi sudah berada di cakupan *context-free language*.

Oleh sebab itu, untuk membuat perilaku kasus yang termasuk *nonconcatenative morphology* dapat diimplementasikan sebagai FST, ia harus dibuat seolah-olah seperti *regular language*. Kasus *nonconcatenative morphology* tersebut mendapat perlakuan khusus yang membuatnya dapat diimplementasikan sebagai FST. Bentuk perlakuan khusus tersebut dapat dilihat pada subbab ini setelah pembahasan skema *nonconcatenative morphology*.

Berikut ini contoh bentuk kata ulang dalam bahasa Indonesia yang merupakan salah satu contoh bentuk *nonconcatenative morphology* pada Gambar 2-8.

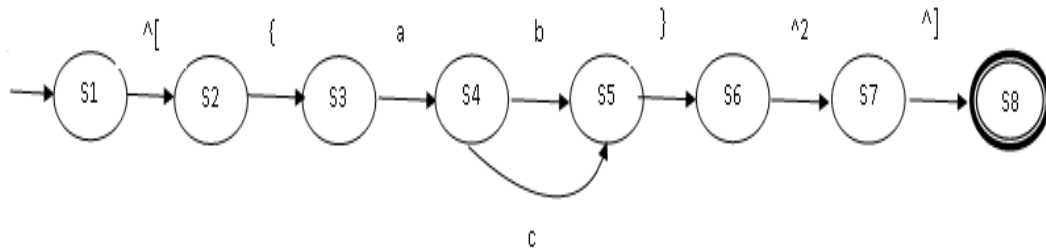


Gambar 2-8. Contoh Skema Pengurai Morfologi untuk *Nonconcatenative Morphology*

Perulangan kata dasar “pukul” menjadi “memukul-mukul” harus dapat dimodelkan dalam suatu FST. Untungnya hal ini dapat dilakukan dengan menggunakan metode *compile-replace* yang memodifikasi *network* hasil *regular expressions* agar dapat menangani *nonconcatenative morphology* dengan *finite-state network*.

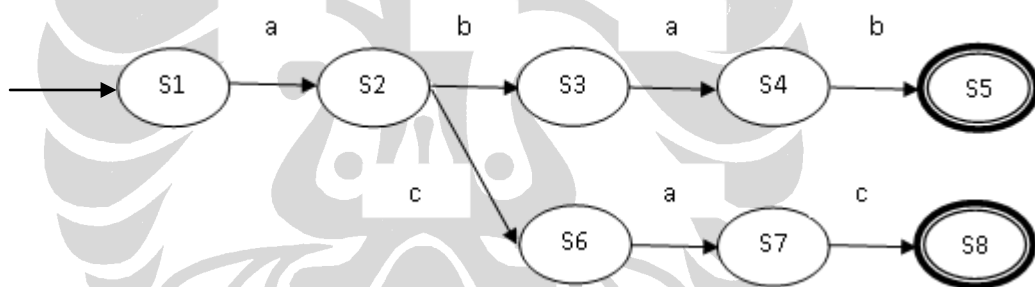
Proses yang sebenarnya dilakukan pada *compile-replace* adalah menjalankan sebuah *regex compiler* lanjutan untuk output dari *regex compiler* sebelumnya. Hasilnya adalah sebuah *finite-state* yang sudah dimodifikasi [BEES03]. Agar hasil dari *regex compiler* sebelumnya dapat diproses oleh *compile-replace* pada *regex compiler* lanjutan, sebuah *regex* harus diberi *delimiter* yaitu simbol “^[]” dan “^]”. Dengan adanya *delimiter* tersebut, *compile-replace*

akan membacanya dan melakukan kompilasi pada regex di dalamnya. Sebagai contoh bisa kita lihat suatu *network input* seperti pada Gambar 2-9.



Gambar 2-9. Contoh *Network Input Compile-Replace*

Ekspresi “{ab}²” yang diletakkan di dalam *delimiter* akan dipetakan menjadi perulangan yang menghasilkan bentuk “abab”. Begitu juga untuk ekspresi “{ac}²” akan dipetakan menjadi bentuk perulangan “acac”. Contoh dapat dilihat pada Gambar 2-10.



Gambar 2-10. Contoh *Network Output Compile-Replace*

Proses ini menghasilkan *network* yang menerima “abab” dan “acac” sesuai yang diharapkan untuk reduplikasi. Penjelasan lebih lanjut tentang metode *compile-replace* akan dijelaskan sebagai salah satu fitur dalam subbab 4.1.1.

2.6 Morfologi Bahasa Indonesia

Dalam bahasa Indonesia suatu kata dapat dibentuk dari kata lain [ALWI03]. Suatu kata dapat dipotong-potong menjadi bagian lain yang lebih kecil di mana tiap pemotongan itu mengandung makna dan informasi linguistik. Sebagai contoh kata penulisan-penulisan dapat dipotong sebagai berikut:

penulisan-penulisan → tulisan-tulisan → tulis-tulis → tulis

Kata “tulis” sudah tidak bisa dibagi lagi menjadi suatu bentuk yang lebih kecil yang mengandung makna. Kata “penulisan-penulisan” dapat dibentuk dari morfem “tulis”, “peN”, “-an” lalu dilakukan pengulangan.

Kata-kata dalam bahasa Indonesia berdasarkan bentuknya bisa kita bagi menjadi empat bagian yaitu kata dasar, kata turunan (dibentuk dari kemunculan imbuhan pada suatu kata), kata ulang, dan kata majemuk. Berhubung kata majemuk tidak di dalam ruang lingkup riset ini, maka berikut akan dijelaskan tentang kata dasar, imbuhan, dan kata ulang.

2.6.1 Kata Dasar

Kata dasar adalah bentuk kata paling sederhana yang memiliki makna dan belum mengalami proses afiksasi ataupun reduplikasi. Beberapa contoh kata dasar di antaranya: “pukul”, “makan”, “keras”, “dingin”, “batu”, “gedung”, “satu”, “bawah”. Ada beberapa macam pembagian kata dasar, akan tetapi dalam pembangunan pengurai morfologi kali ini cukup dibagi menjadi kelas kata dasar verba, adjektiva, nomina, dan kelas kata dll. Penjelasan lebih jauh tentang kelas kata dasar akan dijelaskan pada BAB 3.

2.6.2 Imbuhan (Afiks)

Imbuhan (afiks) adalah suatu morfem terikat yang dipakai untuk menurunkan suatu kata [ALWI03]. Suatu kata dasar ataupun kata turunan dapat kembali diturunkan melalui proses afiksasi atau pemberian imbuhan. Imbuhan dapat dibagi menjadi empat macam yaitu awalan (prefiks), sisipan (infiks), akhiran (sufiks), dan gabungan imbuhan (konfiks).

2.6.2.1 Awalan (Prefiks)

Awalan (Prefiks) adalah imbuhan yang diletakkan pada bagian muka suatu kata [ALWI03]. Peletakan prefiks di depan suatu kata akan menghasilkan suatu kata baru yang mungkin akan mengubah kelas katanya. Awalan (prefiks) yang terdapat dalam bahasa Indonesia di antaranya: “di-”, “meN-”, “peN-”, “ber-”, “per-”, “ter-”, dan “ke-”. Penjelasan perilaku yang terjadi pada awalan-awalan tersebut akan dijelaskan pada BAB 3.

2.6.2.2 Sisipan (Infiks)

Sisipan (Infiks) adalah imbuhan yang diselipkan di tengah suatu kata [ALWI03]. Menyelipkan infiks di tengah kata juga akan menghasilkan suatu kata yang mungkin akan mengubah kelas katanya. Sisipan (infiks) dalam bahasa Indonesia di antaranya: “el”, “er”, dan “em”. Sisipan berada di luar ruang lingkup riset ini.

2.6.2.3 Akhiran (Sufiks)

Akhiran (sufiks) adalah imbuhan yang digunakan di bagian belakang suatu kata [ALWI03]. Akhiran (sufiks) yang diletakkan pada bagian belakang suatu kata juga akan menghasilkan suatu kata baru yang mungkin akan mengubah kelas katanya. Dalam riset ini, tercakup akhiran (sufiks) dalam bahasa Indonesia sebagai berikut: “-an”, “-kan”, dan “-i”. Ada beberapa bentuk akhiran lainnya akan tetapi tidak dicakup dalam riset ini. Untuk penjelasan perilaku yang terjadi pada akhiran-akhirian yang tercakup dalam riset ini akan dijelaskan pada BAB 3.

2.6.2.4 Gabungan Imbuhan (Konfiks)

Gabungan imbuhan (konfiks) merupakan gabungan dari prefiks dan sufiks yang membentuk satu kesatuan [ALWI03]. Penggabungan pada konfiks dilakukan secara serentak, sehingga jika penggabungan prefiks dan sufiks dilakukan tidak secara serentak maka ia bukanlah konfiks. Meskipun begitu, contoh penggabungan seperti itu juga tercakup dalam ruang lingkup riset ini. Selanjutnya untuk konfiks akan ditunjukkan perilakunya pada BAB 3.

2.6.3 Kata Ulang (Reduplikasi)

Kata ulang adalah kata dasar atau kata turunan yang mengalami pengulangan. Bentuk pengulangan tersebut dapat dilakukan pada seluruh kata ataupun sebagian kata tersebut. Kata ulang juga akan menghasilkan suatu kata baru yang mungkin memiliki kelas kata yang berbeda dari kata dasar ataupun kata turunan yang diulang.

Kata ulang dapat dibedakan menjadi kata ulang sejati, kata ulang sebagian, dan kata ulang berimbuhan. Pada kata ulang sejati, seluruh kata dasar ataupun kata turunannya diulang, sedangkan pada kata ulang sebagian atau berimbuhan

tidak mengulang keseluruhan kata dasar atau kata tutunan tersebut. Perilaku dari kata ulang akan dijelaskan pada BAB 3.

2.7 Pengurai Morfologi Bahasa Indonesia

Pengurai morfologi (*morphological analyzer*) untuk bahasa Indonesia sebelumnya juga sudah pernah dilakukan oleh Hendra Hartono pada tahun 2002 sesuai dengan Tugas Akhir (TA) yang ia lakukan pada saat itu. Judul dari TA tersebut yaitu “Pengembangan Pengurai Morfologi untuk bahasa Indonesia dengan Model Morfologi Dua Tingkat Berbasis PC-KIMMO” [HART02].

Pada TA tersebut Hendra Hartono juga menggunakan aturan-aturan morfofonemik dan morfotaktik yang merujuk kepada buku “Tata bahasa Baku bahasa Indonesia” karangan Hasan Alwi dkk [ALWI03]. Oleh karena itu dari segi rancangan morfofonemik dan morfotaktik ada beberapa kesamaan dengan TA ini.

Dalam konsep teori, TA Hendra Hartono juga menggunakan prinsip morfologi dua tingkat dengan memanfaatkan *finite-state transducers*. Perbedaannya adalah pada bagian implementasi Hendra Hartono menggunakan PC-KIMMO sebagai *tools* untuk membangun pengurai morfologi tersebut. Implementasi dalam PC-KIMMO tersebut berupa kumpulan-kumpulan aturan morfotaktik maupun morfofonemik yang digabungkan. Salah satu contoh implementasi aturan dalam PC-KIMMO dapat kita lihat sebagai berikut:

RULE t:n ⇔ g:@+:@ ___

Aturan di atas merupakan aturan *t replacement with m* yang akan memetakan *t* yang terdapat pada pita atas menjadi *n* pada pita bawah dengan syarat terdapat karakter “g+” sebelumnya. Aturan di atas hanyalah satu dari sekian banyak aturan yang pada akhirnya nanti akan menghasilkan pengurai morfologi yang telah diimplementasikan oleh Hendra Hartono.

Oleh karena banyaknya kesamaan, maka pada bagian awal pengembangan TA ini juga banyak mengacu kepada TA Hendra Hartono. Dapat dikatakan TA ini

juga mengembangkan lebih lanjut apa yang dilakukan oleh Hendra Hartono pada Tugas Akhirnya tersebut.

Beberapa bagian yang dikembangkan dari TA Hendra Hartono selain pemakaian *tools* baru terletak pada informasi linguistik yang terkandung didalamnya. Pengurai morfologi yang dikembangkan pada TA ini menggunakan banyak *tags* yang akan memberikan informasi-informasi linguistik yang diperlukan dalam mempelajari bentuk kata tersebut. Selain informasi-informasi linguistik tersebut, pengurai morfologi ini juga mencakup kata ulang yang pada TA Hendra Hartono tidak dikembangkan. Pada TA Hendra Hartono, pengurai morfologi yang dikembangkan hanya untuk kata dasar, prefiks, sufiks, dan konfiks. Berbagai tipe kata ulang tidak tercakup di dalam TA tersebut. Dengan *tools* yang berbeda, TA kali ini mengantisipasi perilaku kata ulang. Pada TA Hendra Hartono kata ulang tidak diimplementasikan karena keterbatasan *tool* yang ia gunakan. Selain itu, juga dilakukan pengkajian ulang dari aturan-aturan morfonemik dan morfotaktik untuk pengembangan pengurai morfologi kali ini.