

# DOKUMEN BUKTI HAK CIPTA

Rancangan Interaksi antara Mesin Augmented Reality (Sistem AR) dan Manusia, dengan mengkonstruksikan bahasa yang gramatikanya sepadan dengan generalized Finite State Automata (gFSA) dan Nondeterministik gFSA



Pengusul Hak Cipta

**Dr. Aslan Alwi, S.Si., M.Cs**

**Dr. Azhari, M.T**

**Dr. Suprpto, M.Kom**

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2021

# Rancangan Interaksi antara Mesin Augmented Reality (Sistem AR) dan Manusia, dengan mengkonstruksikan bahasa yang gramatiknya sepadan dengan generalized Finite State Automata (gFSA) dan Nondeterministik gFSA

## A. RINCIAN RANCANGAN

Bagian ini merinci rancangan interaksi antara sistem Augmented Reality dengan manusia yang diusulkan sebagai bagian utama dari pengusulan hak cipta. Deskripsi rancangan ini dijelaskan secara rinci pada bagian klaim sebagai berikut.

## KLAIM RANCANGAN

Bagian ini mengemukakan serangkaian argumentasi untuk memberikan bukti dan konstruksi yang mendukung klaim rancangan. Sebelum rancangan dikemukakan secara rinci, terlebih dulu klaim dinyatakan sebagai konjektur-konjektur yang kemudian setiap pembuktian konjektur secara keseluruhan menyatakan rancangan interaksi menggunakan gFSA.

**Konjektur 1. (Konjektur ARS P-NP)** Untuk setiap interaksi pengguna-ARS, dapat dikonstruksikan interaksi yang memiliki kompleksitas polinomial ( $P$ ) dan Nondeterministik polinomial ( $NP$ ).

**Konjektur 2. (Konjektur Regular-ARS)** Untuk setiap interaksi pengguna-ARS, dapat dikonstruksikan sebuah bahasa formal reguler untuk himpunan penanda yang digunakan oleh pengguna untuk berinteraksi dengan ARS.

**Konjektur 3. (Konjektur gFSA-ARS)** Untuk setiap interaksi pengguna-ARS, dapat dikonstruksikan sebuah bahasa formal reguler yang menggunakan generalized Finite State Automata (gFSA) untuk himpunan penanda pengguna.

## Pembuktian dan Analisis Konjektur ARS P-NP

Bagian ini mengemukakan serangkaian argumentasi untuk memberikan bukti dan konstruksi yang mendukung konjektur ARS P-NP (konjektur 1). Argumentasi ini dimulai dengan menyatakan ulang konjektur 1 ke dalam proposisi 1 sebagai berikut.

**Proposisi 1.** Untuk setiap interaksi antara pengguna dan augmented reality system (ARS) dapat memiliki kompleksitas yang polinomial ( $P$ ) dan nondeterministik polinomial ( $NP$ ).

Augmented reality adalah sebuah proses mengkombinasikan atau “menambahkan” (*augmenting*) video atau gambar (2D atau 3D) dengan melakukan overlay terhadap citra dengan data-data

bermanfaat yang dibangkitkan oleh komputer (Hosch, 2017). *Augmented reality* menampilkan dunia nyata dan objek-objeknya dan kemudian memperluasnya. Pada saat ini, objek-objek pada dunia nyata dapat ditampilkan dalam dua metode yang berbeda. Metode-metode ini membuat *augmented reality* dibangkitkan dalam cara *marker based augmented reality* dan *marker-less augmented reality* (Kim dkk. 2014). *Augmented Reality* (AR) menggunakan teknik komputer visi, pengolahan citra dan *computer graphics* untuk menggabungkan konten digital ke dalam dunia nyata. Ini memungkinkan interaksi secara waktu nyata (*real time*) antara pengguna, objek nyata dan objek virtual. AR dapat, sebagai contoh, digunakan untuk menempelkan grafik 3D ke dalam sebuah video sedemikian rupa seakan elemen-elemen virtul adalah bagian dari lingkungan nyata (Siltanen, 2012).

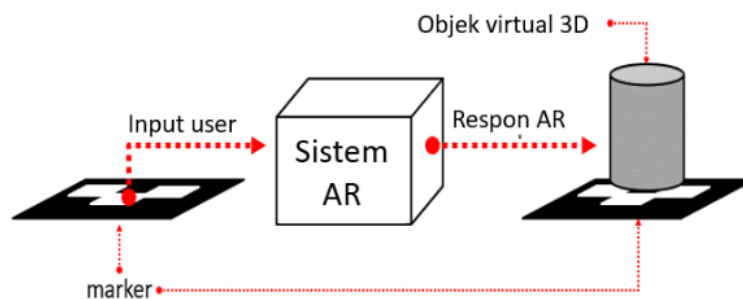
Sebuah penanda adalah sebuah tanda atau citra yang dapat dideteksi oleh sistem komputer melalui video menggunakan teknik pengolahan citra, pengenalan pola dan komputer visi (contoh penanda seperti pada Gambar 5.1). Sekali terdeteksi, penanda kemudian didefinisikan skala dan posisinya secara benar terhadap kamera, sekali terdeteksi, dia kemudian mendefinisikan keduanya yaitu skala yang benar (*correct scale*) dan posisi kamera (Siltanen, 2012). Penanda dapat didefinisikan dalam bermacam fitur. Penanda dapat berupa gambar 2D (Waruwu dkk. 2015) atau berupa fitur titik-titik objek (Pawade-Sakhapara, 2018) atau mengambil keuntungan dari fitur-fitur lokal yang ada (Khalifa, 2015).



**Gambar 5.1. Sebuah Contoh Penanda**

Interaksi antara pengguna dan ARS dalam penelitian yang dilakukan oleh D, Kumar-Ram (2016), Shetty dkk. (2015), Mota dkk. (2018) dan Sin-Zaman (2009), pada hakikatnya dapat dilihat sebagai sebuah interaksi dengan kompleksitas konstan atau linier. Ini karena untuk sebuah penanda maka di sana ada sebuah respon ARS yang sesuai. Jika dinyatakan dalam bentuk fungsi atau pemetaan,  $f(\text{penanda}) = \text{respon\_ARS}$  atau dijelaskan sebagai  $f(m) = r$  dan  $m$  penanda dan  $r$  respon ARS, jika  $\forall m$  penanda  $r$  tetap maka interaksi ini konstan. Akan tetapi, boleh jadi untuk setiap  $m$  diperoleh respon yang berbeda, yaitu  $y$  maka interaksi ini memperoleh bentuk linier sebagaimana dinyatakan;  $f(m) = y$  dan  $m$  penanda dan  $y$  respon ARS. Interaksi ini dapat menjadi lebih kompleks

dengan menyatakannya dalam bentuk  $f(x)=a.y + r$  dan  $x$  penanda dan  $a.y$  adalah respon  $y$  yang muncul sebanyak  $a$  kali dan  $r$  data tambahan yang di *overlay* pada objek atau respon ARS lain. Respon ARS dapat berupa objek virtual 2D atau 3D atau teks. Gambar 5.2. menunjukkan sebuah ilustrasi sederhana bagaimana interaksi dengan kompleksitas konstan atau linier dibangun. Pada Gambar 5.2 pengguna memberikan input berupa sebuah penanda kemudian ARS melakukan pemindaian terhadap penanda. Ketika penanda dikenali maka ARS lalu merespon dengan menambahkan objek 3D berupa silinder diatas penanda.



**Gambar 5.2. Interaksi Pengguna-ARS dengan Kompleksitas Konstan atau Linier (Sumber gambar: pribadi)**

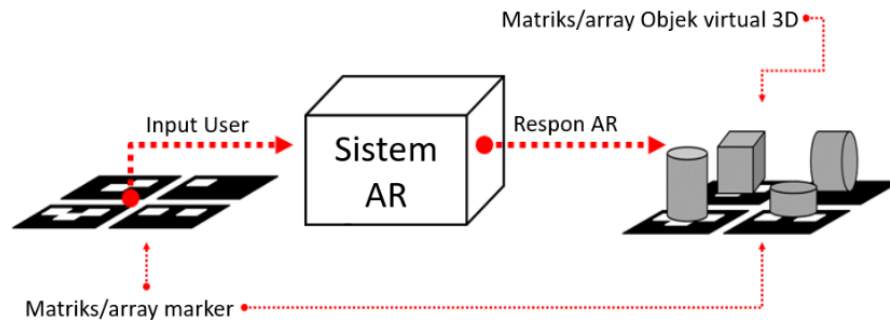
Gambar 5.3. memberikan contoh yang input penanda berupa matriks penanda atau *array* penanda. Ini serupa dengan penelitian yang dilakukan oleh Kim dkk. (2014), Kurpyte-Navakauskas (2014), Tateno dkk. (2017). Walaupun mereka menggunakan sebuah matriks penanda, interaksi juga dapat dilihat sebagai interaksi dengan kompleksitas konstan atau linier. Ini karena untuk sebuah matriks penanda mendapat sebuah respon ARS atau matriks respon ARS. Disana hanya ada pemetaan satu ke satu (*bijective mapping*) antara matrik penanda ke matriks respon ARS. Pemetaan konstan atau linier ini dapat dilihat sebagai berikut:

$f(v)=c$  dan  $v$  vektor atau matriks penanda dan  $c$  respon ARS

$f(v)=w$  dan  $v$  vektor atau matriks penanda dan  $w$  adalah vektor atau matriks respon ARS

$f(v)=a.w + c$  dan  $v$  adalah vektor atau matriks penanda dan  $w$  adalah vektor atau matriks respon ARS,  $a.w$  adalah lebih dari satu respon  $w$  yang jumlahnya sebanyak  $a$  dan  $c$  data tambahan yang di *overlay* pada objek atau respon ARS lain.

Interaksi dengan bentuk kompleksitas ini dapat dilihat dalam contoh ilustrasi pada Gambar 5.3.



**Gambar 5.3. Interaksi Pengguna-ARS dengan Kompleksitas Konstan atau Linier (Sumber gambar: pribadi)**

Pada rancangan ini ditunjukkan bahwa interaksi ini dapat diperluas ke dalam kompleksitas yang polinomial dan nondeterministik polinomial. Perluasan ini dilakukan dengan memperluas input pengguna yang semula hanya sebuah unit simbol penanda (unit simbol sebagai sebuah penanda atau sebagai sebuah vektor atau matriks penanda) menjadi sebuah string atau *word* unit simbol.

String atau *word* unit simbol dikonstruksikan dengan mengkombinasikan sejumlah penanda atau sejumlah vektor/matriks penanda yang untuk setiap *word* dipetakan ke sebuah respon ARS atau sebuah *word* respon ARS (kombinasi respon ARS). Ini bukan hanya sekedar sebuah *array* penanda dipetakan ke sebuah *array respon ARS* karena semua *word* yang dapat diberikan oleh pengguna kepada ARS dapat dikumpulkan dan dihimpu menjadi sebuah himpunan yang membangun sebuah bahasa (*language*).

Sebagai contoh, semua kumpulan *word* itu dapat dikonstruksikan sedemikian rupa sehingga membentuk sebuah bahasa reguler dalam hireraki tipe bahasa Chomsky. Bahasa reguler ini dapat dibuat dengan membangun sebuah *generalized Finite State Automata* (gFSA) yang dapat menerima setiap *word* penanda yang diberikan oleh pengguna kepada sistem ARS. Karena dia membangun sebuah bahasa reguler yang diterima oleh sebuah gFSA maka ini menjadikan kompleksitas interaksi antara pengguna-ARS menuruti kompleksitas bahasa reguler atau gFSA, yaitu membangun interaksi pengguna-ARS dengan kompleksitas polinomial, bukan hanya sekedar interaksi dengan kompleksitas konstan atau linier.

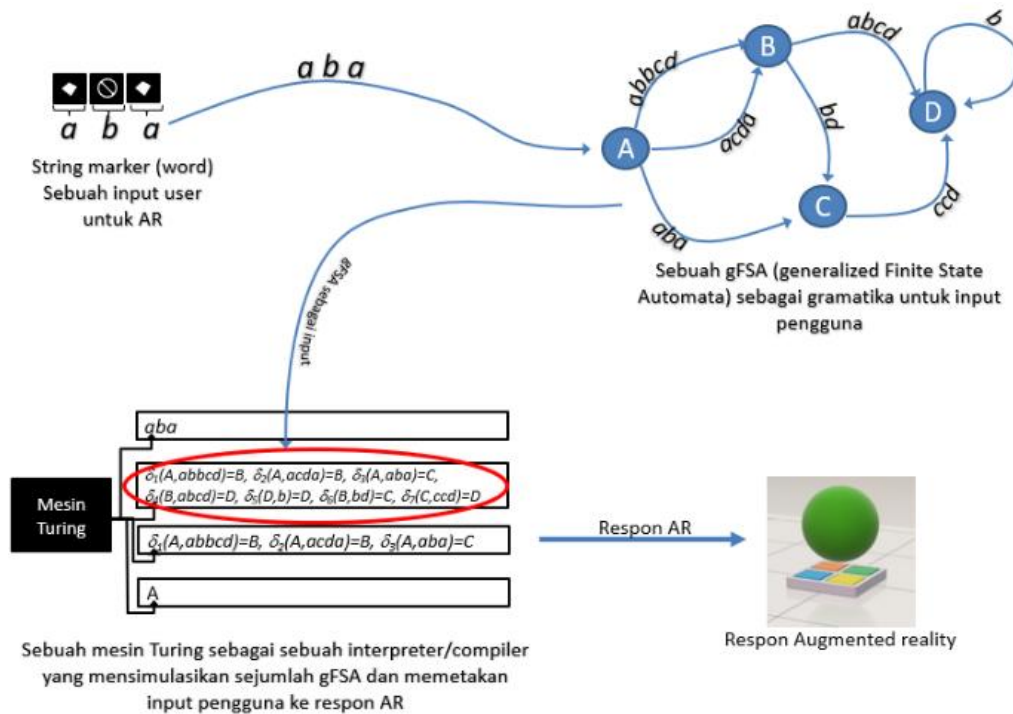
Sebuah mesin Turing universal (UTM) ditempatkan antara keduanya, sebagai penerjemah bahasa reguler antara pengguna dan ARS. UTM bertindak sebagai mesin *interpreter* atau *compiler* antara pengguna dan ARS. ARS memeriksa tata bahasa dari bahasa pengguna lalu memetakannya ke respon ARS yang sesuai atau jika dia diposisikan sebagai *compiler* maka UTM selain memeriksa tata bahasa juga terlebih dahulu memeriksa token-token yang membangun kalimat dari input pengguna.

Gambar 5.4. menunjukkan bagaimana interaksi dengan kompleksitas polinomial dibangun. Pada Gambar 5.4. pengguna memberikan input kepada ARS berupa tiga buah penanda yang diwakili secara simbolik oleh huruf  $a$  dan  $b$ . Tiga buah penanda ini membangun sebuah kombinasi penanda “ $aba$ ”. Kombinasi penanda ini dinyatakan sebagai sebuah string penanda atau *word* penanda. Kombinasi  $aba$  adalah sebuah kalimat atau kata dari sebuah bahasa yang seluruh elemen stringnya adalah dapat diterima oleh sebuah automata gFSA seperti pada Gambar 5.4. sehingga kombinasi  $aba$  haruslah juga dapat diterima oleh automata gFSA.

Gagasan bahwa kombinasi  $aba$  harus dapat diterima juga oleh automata gFSA adalah sepadan atau analogi dengan prinsip dari *interpreter* atau *compiler* yang konvensional seperti *compiler* untuk bahasa C atau *interpreter* untuk bahasa PHP atau bahasa pemrograman lainnya, yaitu bahwa setiap kalimat (kode) harus dapat diparsing dan diterima sah sesuai dengan tata bahasa C atau PHP yang menjadi perbedaan adalah alat untuk melakukan parsing. Pada bahasa C atau PHP, alat yang digunakan untuk melakukan parsing adalah gramatika bahasa C atau bahasa PHP. Pada Gambar 5.4., alat yang digunakan untuk melakukan parsing adalah automata gFSA.

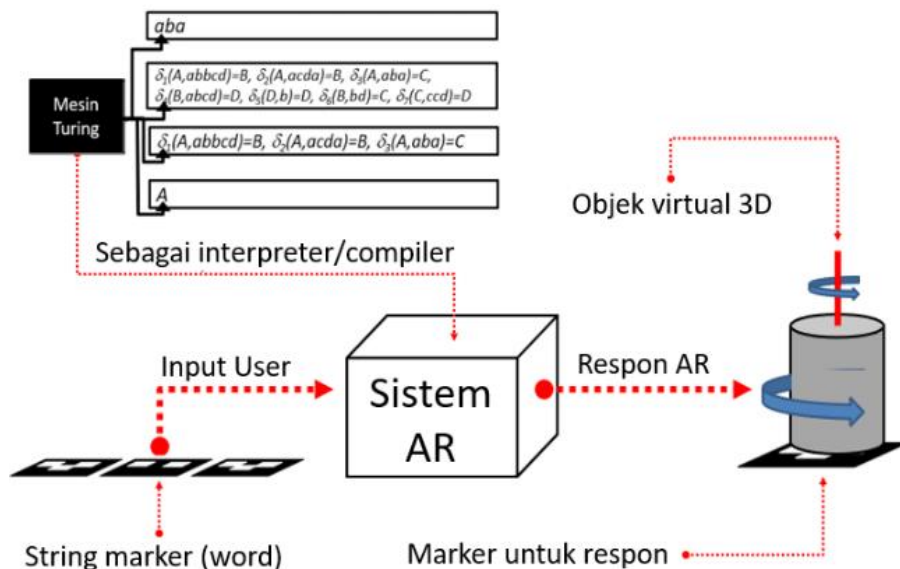
Mesin Turing universal (UTM) memanfaatkan gFSA sebagai alat untuk melakukan parsing dan berkedudukan sebagai *interpreter* atau *compiler* pada Gambar 5.4. Mesin UTM ini menyandikan automata gFSA (enkoding terhadap gFSA) sebagai semata sekumpulan instruksi yang dapat dieksekusi oleh mesin UTM. Enkoding gFSA pada mesin UTM pada Gambar 5.4. adalah  $\{\delta(A, abbcd)=B, \delta(A, acda)=B, \delta(A, aba)=C, \delta(B, abcd)=D, \delta(D, b)=D, \delta(B, bd)=C, \delta(C, ccd)=D\}$  yang A, B, C dan D adalah state-state pada automata gFSA. Enkoding ini disimpan sebagai program pada pita 2 dari mesin UTM. Input pengguna berupa kombinasi penanda  $aba$  ditulis pada pita 1 dan respon ARS ditulis pada pita 4 yang kemudian dipetakan ke aksi *augmented reality*.

Proses parsing oleh mesin UTM berjalan dengan cara *word* pada pita 1 dieksekusi berdasarkan instruksi-instruksi pada pita 2. Jika hasil eksekusi berakhir dengan *state accept* maka kombinasi penanda yang diberikan oleh pengguna adalah sah atau valid, selainnya ditolak.



**Gambar 5.4. Interaksi Menggunakan Sebuah gFSA dan Sebuah UTM**

Sebuah deskripsi yang menyeluruh pada interaksi ini diberikan oleh gambar 5.5 yang UTM yang bertindak sebagai *interpreter/compiler* diletakkan di dalam sistem ARS untuk memeriksa tata bahasa dan menerjemahkannya ke respon ARS yang sesuai.

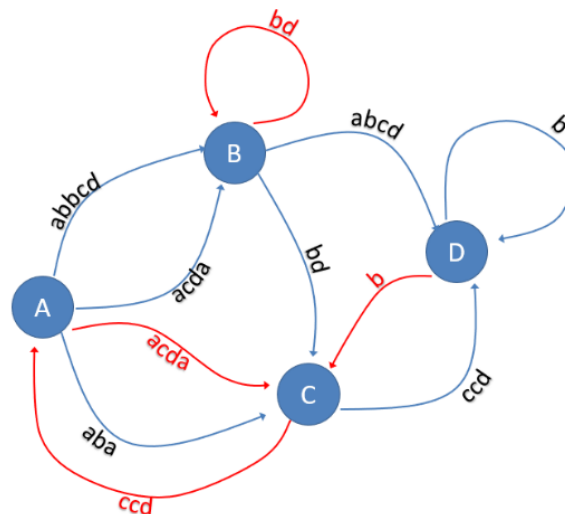


**Gambar 5.5. Interaksi Pengguna-ARS dengan Kompleksitas Polinomial**

Pada Gambar 5.5, mesin UTM ditempatkan di dalam sistem ARS dan berkedudukan sebagai *interpreter* atau *compiler*. Pada contoh Gambar 5.5, inputan pengguna yang berupa kombinasi

penanda *aba* diterima sebagai valid oleh mesin UTM kemudian mesin UTM menerjemahkannya kepada respon ARS yang berupa aksi penambahan benda maya 3D. Benda berupa silinder yang sedang berputar di atas penanda yang disediakan untuk *augmenting*.

Selanjutnya, sebuah interaksi yang memiliki kompleksitas nondeterministik polinomial (NP) dapat dibuat dengan cara membangun bahasa pengguna sebagai bahasa yang diterima oleh sebuah nondeterministik gFSA (NgFSA). Gambar 5.6. memberikan sebuah contoh bahasa penanda pengguna untuk interaksi dengan kompleksitas NP.

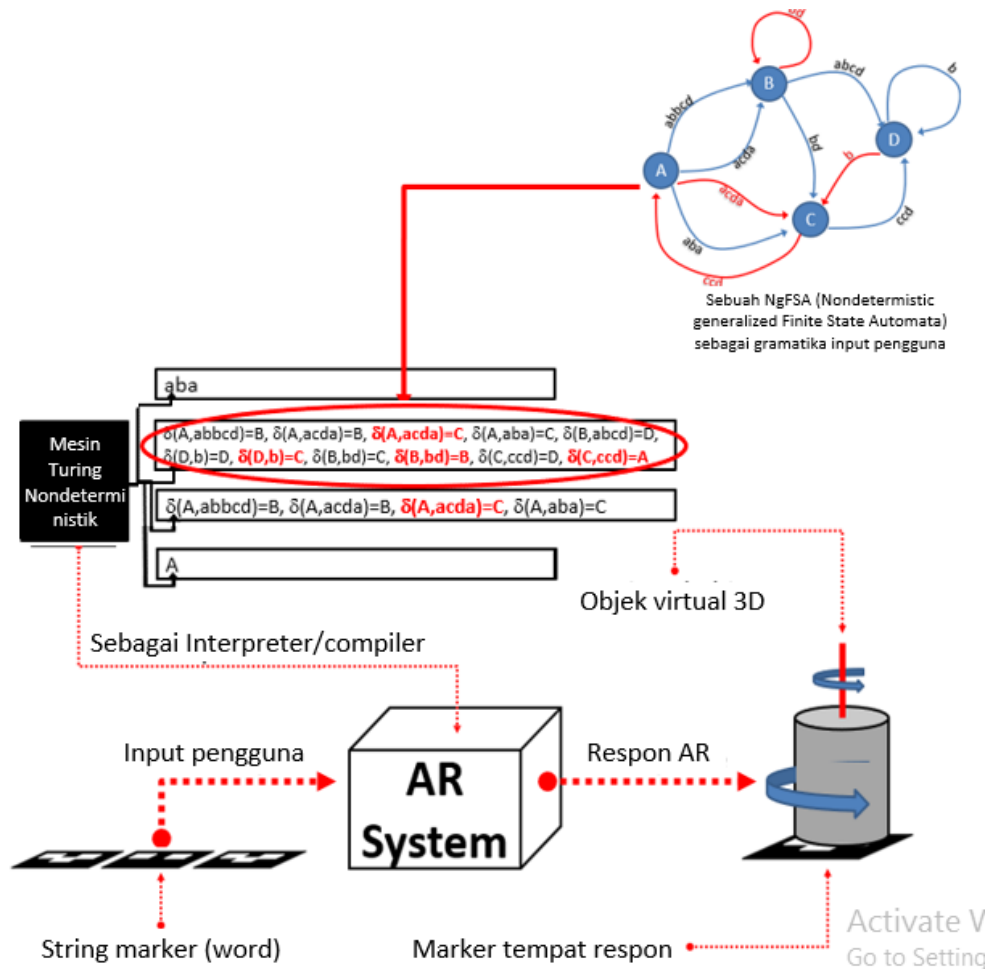


**Gambar 5.6. Contoh Sebuah NgFSA untuk Bahasa Penanda Pengguna**

Gambar 5.6. menyajikan sebuah contoh NgFSA. Pada contoh ini, input “*acda*” dapat berpindah ke *state* C dan juga dapat berpindah ke *state* B sehingga automata pada Gambar 5.6. bersifat tidak tentu atau nondeterministik demikian pula input “*b*” yang *state* berikut (*next state*) jika input “*b*” terjadi terdapat 2 kemungkinan yaitu *state* C atau *state* D juga input “*ccd*” memiliki dua *next state* yaitu *state* A dan D.

Sebuah *interpreter* berupa mesin Turing nondeterministik dapat dibuat untuk mengesekusi NgFSA, memeriksa tata bahasa dan memetakannya kepada respon ARS yang sesuai. Gambar 5.7. menyajikan sebuah contoh *interpreter/compiler* yang menyimpan enkoding NgFSA ke dalam pita 2 mesin UTM. UTM lalu menggunakan enkoding NgFSA untuk memeriksa keabsahan bahasa yang diinputkan oleh pengguna. Enkoding NgFSA dieksekusi secara nondeterministik sedemikian rupa respon ARS menjadi nondeterministik sebagai akibat input pengguna yang menggunakan bahasa nondeterministik.





**Gambar 5.7. Interaksi Pengguna-ARS dengan Kompleksitas Nondeterministik Polinomial (Sumber gambar: pribadi)**

Dari sejumlah penjelasan di atas diperoleh sebuah argumentasi yang menunjukkan bahwa interaksi antara pengguna dan ARS dengan kompleksitas polinomial atau nondeterministik polinomial yang dinyatakan oleh konjektur 1 adalah ada (eksis) dan dapat dikonstruksikan yang memiliki arti bahwa dia dapat ada dan dikonstruksikan menggunakan mesin Turing.

Pengertian “ada” atau “eksis” secara formal matematika adalah berarti pada setiap interaksi antara pengguna-ARS, **dapat** dikonstruksikan sebuah bahasa yang sepadan dengan automata gFSA atau NgFSA yang memiliki kompleksitas *polynomial* atau *nondeterministic polynomial* dan **dapat** dikonstruksikan sebuah mesin Turing yang berkedudukan sebagai *interpreter* atau *compiler* ditengahnya yang dapat memetakan bahasa interaksi tersebut ke sebuah himpunan respon ARS. Pada penjelasan di atas, konstruksi-konstruksi tersebut telah ditunjukkan. Untuk penjelasan dan konstruksi yang lebih rinci dikemukakan pada bab-bab berikutnya.

Pembuktian konjektur selanjutnya adalah menunjukkan bahwa mesin Turing universal yang berposisi sebagai *interpreter* atau *compiler* adalah dapat dibangun. Secara umum, interaksi ini dapat diperluas untuk sembarang interaksi antara pengguna kepada sembarang jenis mesin dan interaksi mesin kepada mesin, bahkan dengan mengganti gFSA dengan fuzzy gFSA, *probabilistic* gFSA atau automata kuantum, sembarang bahasa interaksi dapat dibangun.

### Kesimpulan Proposisi 1

Seluruh rangkaian argumentasi yang telah diberikan adalah untuk menunjukkan bagaimana pernyataan konjektur 1 adalah dapat dibangun, yaitu yang konstruksi itu dilakukan menggunakan gFSA atau NgFSA serta interpreternya atau kompilernya menggunakan UTM. Argumentasi untuk proposisi 1 juga sebenarnya menunjukkan bagaimana konjektur 2 dan 3 dijelaskan dan dapat dibangun, yaitu bahwa sebuah bahasa reguler untuk pengguna untuk berinteraksi dengan ARS seperti yang dinyatakan oleh konjektur 2 adalah dapat dibuat. Bahasa reguler itu adalah sekumpulan *word* penanda dari pengguna yang mematuhi sebuah *finite state automata* (FSA).

Argumen-argumen itu sekaligus ini juga memberi jalan bahwa konjektur 3 adalah dapat dibuktikan dengan mengeneralisir FSA menjadi sebuah gFSA. Karena gFSA adalah juga sebuah FSA maka dia juga menerima bahasa reguler. Akan tetapi, untuk sebuah argumentasi yang lebih rinci, konjektur 2 dan konjektur 3 diwakilkan kepada proposisi 2 dan untuk proposisi ini diberikan sebuah argumentasi tentang bagaimana FSA atau gFSA dapat digunakan untuk membangun sebuah bahasa reguler untuk himpunan *word* penanda.

### Pembuktian dan Analisis Konjektur Regular-ARS dan Konjektur gFSA-ARS

Bagian ini menunjukkan sejumlah argumentasi untuk memberikan serangkaian bukti dan konstruksi yang mendukung konjektur Regular-ARS dan konjektur gFSA-ARS (konjektur 2 & 3).

***Proposisi 2.*** Untuk setiap interaksi pengguna-ARS dapat dikonstruksikan sebuah bahasa formal reguler yang menggunakan generalized Finite State Automata (gFSA) untuk himpunan penanda pengguna.

Pembuktian dilakukan dengan menunjukkan bahwa proposisi 2 adalah benar sehingga konjektur 3 adalah benar maka secara otomatis itu juga membuktikan konjektur 2 sehingga pembuktian proposisi 2 cukup untuk kedua konjektur.

Motivasi utama daripada teori yang dikemukakan dalam rancangan ini adalah merumuskan secara formal interaksi berbasis penanda antara pengguna dan ARS. Argumentasi ini melihat bahwa penyajian penanda-penanda kepada ARS adalah sebuah sajian string atau *word* dari sejumlah

kombinasi penanda. Dalam sejarah ide ARS sendiri, konsep tentang penanda oleh Siltanen (2012), Kim dkk. (2014) dan beberapa penelitian sebelumnya yang dilakukan oleh D, Kumar-Ram (2016) dan Shetty dkk. (2015) adalah terutama menggunakan penanda sebagai input untuk interaksi antara pengguna dan ARS. Akan tetapi, rancangan ini, mencari sebuah bentuk perluasan dari kompleksitas interaksi dari penelitian yang mereka lakukan. Perluasan ini dimulai dengan mengubah cara pandang penyajian penanda pada penelitian-penelitian mereka, yaitu dari penyajian sebuah penanda atau sebuah vektor atau matriks penanda untuk sebuah respon ARS atau sebuah vektor/matriks respon ARS menjadi penyajian penanda sebagai string penanda atau *word* penanda dan kumpulan *word* penanda itu membangun suatu bahasa formal. Pada proposisi 2 ini akan ditunjukkan bahwa bahasa formal yang dibangun adalah bahasa reguler.

Beberapa studi yang lain juga menggunakan kombinasi penanda atau sekumpulan penanda di dalam bentuk sebagai sebuah matriks penanda, yaitu seperti pada penelitian Kim dkk. (2014), Kurpyte-Navakauskas (2014) atau Tateno dkk. (2017). Akan tetapi, tidak dimaksudkan untuk membangun sebuah string penanda atau *word* penanda dan lebih penting lagi, mereka tidak memaksudkan untuk membangun suatu bahasa formal sebagai bahasa interaksi antara pengguna dan ARS. Mereka hanya terbatas membangun suatu pemetaan *bijective* antara penanda pengguna ke respon ARS atau antara matriks penanda ke matriks respon ARS. Sebatas membangun kompleksitas yang konstan atau linier. Pada kebanyakan penelitian tentang ARS, tidak dimaksudkan untuk meninjau kompleksitas interaksi pengguna-ARS. Seperti misalnya penelitian yang dilakukan oleh Sin-Zaman (2009), Yao dkk. (2008) dan Ilievski-Trajkovik (2013) penelitiannya dimaksudkan untuk menemukan dan mengusulkan fungsi-fungsi baru dalam ARS juga penelitian yang dilakukan oleh Zhang-Lew (2012), Vera dkk. (2011) dan Rafał Wojciechowski (2012) mengusulkan sebuah *framework* ARS pada level atas serta penelitian yang dilakukan oleh Pagani (2014) menawarkan sebuah cara *registering images* pada level bawah ARS atau sebuah konsep citra (*adaptive thresholding*) terhadap citra penanda (Bradley, 2013). Rancangan ini mengusulkan sebuah sudut pandang yang berbeda, yaitu pada sisi pandang kompleksitas interaksi dan mengusulkan sebuah bentuk formal interaksi antara pengguna-ARS dengan mengemukakan gagasan bahasa formal bagi interaksi pengguna-ARS.

Pada proposisi-proposisi berikutnya akan ditunjukkan juga bahwa bahasa formal yang dapat dibangun adalah tak terbatas pada bahas reguler. Akan tetapi, mencakup seluruh tipe bahasa formal

dalam hierarki Chomsky, bahkan mencakup bahasa-bahasa bersifat nondeterministik, probabilistik atau fuzzy, bahkan dapat diperluas lebih dari itu.

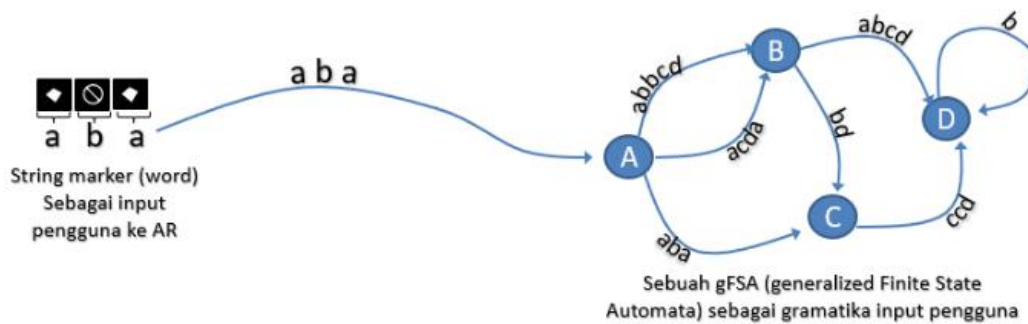
Interaksi antara pengguna dan ARS dalam bentuk *word* penanda memerlukan gagasan bahasa yang dibentuk dari himpunan *word* penanda dan memenuhi suatu tata bahasa atau automata. Karena itu, sebagai usaha pertama, untuk membangun bahasa interaksi dari himpunan *word* penanda, dibangun sebuah automata, dalam hal ini dipilih sebuah *generalized finite state automata* (gFSA) lalu akan ditunjukkan bahwa sebuah bahasa dari himpunan *word* penanda dapat dibangun dengan menggunakan gFSA.

*Generalized finite state automata* (gFSA) adalah sebuah *finite state automata* (FSA) yang inputnya diperumum menjadi sebuah *word*, bukan hanya sebuah karakter. Definisi formal gFSA dinyatakan dalam definisi 5.1. berdasarkan definisi yang dikemukakan oleh (Sánchez and Shapiro, 2016) tetapi menghilangkan variabel M pada pernyataan “tripel  $M = (G, S, Y)$ ” sebagai penyederhanaan.

**Definisi 5.1** Sebuah *generalized finite state automata* (gFSA) adalah sebuah *tripel*  $(G, S, Y)$ . Di sini  $G = (V, E, \phi)$  adalah sebuah graf berarah berlabel yang pelabelan  $\phi$  menandai setiap  $e \in E$ , sebuah bahasa  $L_e$ .  $S \in V$  adalah *start state*.  $Y \subset V$  adalah himpunan *state accept*.

Di dalam konteks interaksi antara pengguna dan ARS,  $L_e$  adalah sebuah bahasa yang mengumpulkan semua *word* penanda yang mungkin disajikan atau diinputkan oleh pengguna. Pada perumuman ini, cara pengguna memasukkan inputan (penanda) adalah diperluas ke dalam bahasa reguler. Pada implementasinya, di sana mungkin ada lebih dari satu gFSA yang dibutuhkan untuk membangun interaksi antara pengguna dan ARS. Ini sesuai dengan variasi konteks dari interaksi atau bahwa bahasa reguler yang dibangun oleh pengguna memiliki gramatika yang sepadan dengan beberapa buah gFSA. Karena itu, ARS memerlukan sebuah *interpreter/compiler* yang mampu menerjemahkan bahasa pengguna menggunakan beberapa gFSA berbeda.

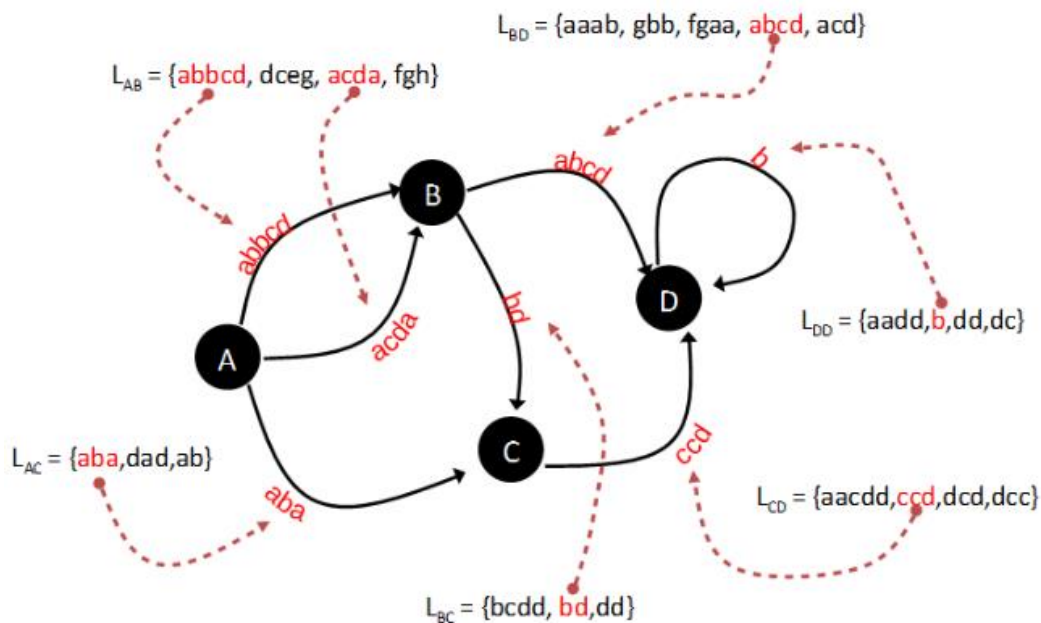
Gambar 5.8. mengilustrasikan bagaimana *word* penanda “aba” yang diinputkan oleh pengguna diterima oleh sebuah gFSA yang gFSA bertindak sebagai padanan gramatika bahasa pengguna.



**Gambar 5.8. Generalized Finite State Automata yang Digunakan untuk Menerima Sebuah Word Penanda.**

Gambar 5.8. adalah sebuah pemaknaan dan penggunaan gFSA yang sederhana untuk digunakan membangun bahasa pengguna. Sebuah pengertian yang rinci dan dalam untuk definisi 5.1 diuraikan pada paragraf berikut dan ditunjukkan bagaimana dia disederhanakan menjadi seperti Gambar 5.8 untuk digunakan membangun bahasa pengguna.

Definisi 5.1. secara lengkap, jika diilustrasikan, adalah sebagaimana contoh gFSA pada Gambar 5.9.



**Gambar 5.9. Contoh gFSA Sesuai Definisi 5.1.**

Untuk setiap *edge* atau transisi dari graf gFSA, terdapat sebuah bahasa reguler di sana, misal antara transisi *state* A ke *state* B, terdapat bahasa  $L_{AB}$ . Pada contoh Gambar 5.9., bahasa  $L_{AB} = \{abcbd, dceg, acda, fgh\}$  sehingga sebuah bahasa yang diterima oleh gFSA, adalah gabungan dari semua konkatenasi bahasa transisi, untuk semua transisi yang berakhir di *state accept*. Untuk menyatakan

bahasa yang diterima oleh sebuah gFSA, dibawah ini adalah deskripsi bahasa yang diterima oleh gFSA pada Gambar 5.9.

Sebagai penjelasan pertama dinyatakan tentang konkatenasi dua bahasa. Buat simbol operasi konkatenasi  $\circ$  yang memiliki arti sebagai berikut:

Misalkan  $L_{AB}$   $\circ$   $L_{BD}$  maka  $L_{AB} \circ L_{BD}$  adalah menyatakan konkatenasi bahasa  $L_{AB}$  dan  $L_{BD}$  yang dinyatakan dalam notasi himpunan

$$L_{AB} \circ L_{BD} = \{xy \mid x \in L_{AB} \text{ dan } y \in L_{BD}\}.$$

Misalkan  $D$  adalah *accept state* dan  $A$  adalah *start state* maka bahasa yang diterima oleh gFSA dapat dinyatakan sebagai  $L(\text{gFSA})=L_{AD}$

$L_{AD}$  adalah sambungan semua bahasa yang dikonkatenasi untuk semua kemungkinan transisi dari *start state*  $A$  ke *state* finish  $D$ . Setiap kemungkinan jalur transisi itu dikaji sebagai berikut:

Kemungkinan 1:  $A \rightarrow B \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AB} \circ L_{BD}$

Kemungkinan 2:  $A \rightarrow C \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AC} \circ L_{CD}$

Kemungkinan 3:  $A \rightarrow B \rightarrow C \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AB} \circ L_{BC} \circ L_{CD}$

Kemungkinan 4:  $A \rightarrow B \rightarrow D \rightarrow \dots \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AB} \circ L_{BD} \circ L_{DD} \dots \circ L_{DD}$

Kemungkinan 5:  $A \rightarrow C \rightarrow D \rightarrow \dots \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD}$

Kemungkinan 6:  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow \dots \rightarrow D$

maka bahasa yang dibentuk pada jalur ini adalah  $L_{AB} \circ L_{BC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD}$

kemudian bahasa  $L_{AD}$  dapat dinyatakan sebagai gabungan seluruh kemungkinan jalur transisi sebagai berikut:

$$L_{AD} = (L_{AB} \circ L_{BD}) \cup (L_{AC} \circ L_{CD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD}) \cup (L_{AB} \circ L_{BD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD})$$

Ini berarti bahasa yang diterima oleh gFSA dapat dinyatakan sebagai berikut:

$$L(\text{gFSA}) = L_{AD} = (L_{AB} \circ L_{BD}) \cup (L_{AC} \circ L_{CD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD}) \cup (L_{AB} \circ L_{BD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD})$$

Akan tetapi, di sana mungkin ada lebih dari satu *state accept*, bukan hanya D. Misalkan B dan C adalah juga *state accept*. Jika di sana ada 3 *state accept*, yaitu B, C dan D maka bahasa yang diterima oleh gFSA dapat dinyatakan sebagai berikut:

$$L(gFSA) = L_{AD} \cup L_{AC} \cup L_{AB}$$

Secara umum, misalkan bahasa pengguna dinyatakan oleh sebuah gFSA dan  $x \in S$  adalah start *state* dan  $y \in F$  *state* finish maka bahasa yang diterima oleh gFSA adalah :

$$L(gFSA) = \bigcup_{x \in S, y \in F} L_{xy} \quad (5.1)$$

dan

$$L_{xy} = \bigcirc L_{x_i x_{i+1}} \quad (5.2)$$

dan  $x = x_0$ ,  $y = x_n$ ,  $i = 0, 1, 2, \dots, n$  dan  $x_i x_{i+1}$  adalah *adjacent node* (*state* yang tepat bersebelahan dalam satu jalur atau satu transisi).

Pada hakikatnya,  $L(gFSA)$  adalah sebuah bahasa reguler. Ini dapat ditunjukkan dengan mengemukakan argumen berikut:

***Teorema 5.1***  $L(gFSA)$  adalah sebuah bahasa reguler.

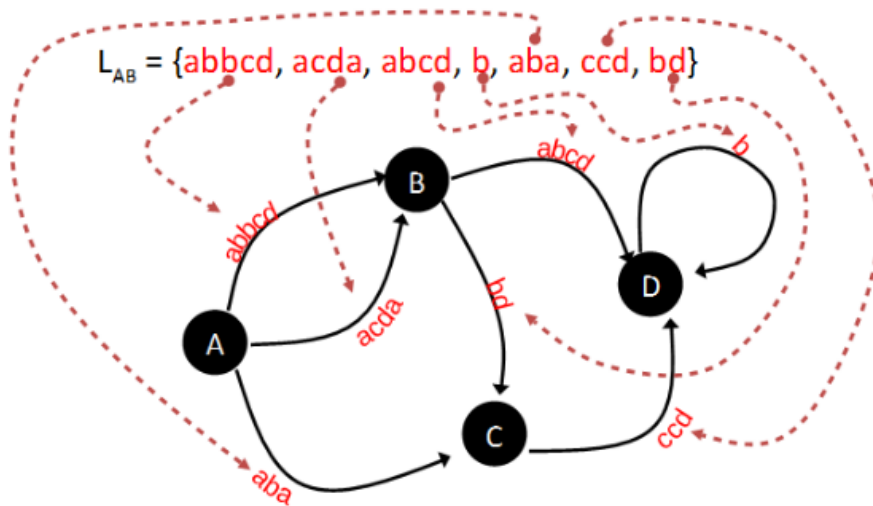
### **Bukti**

Untuk setiap *word* pada label transisi gFSA nyatakan dalam sebuah simbol tunggal. Kumpulkan semua simbol tunggal sebagai sebuah himpunan terminal. Tulis ulang seluruh gFSA menjadi semata sebuah FSA dengan mengganti setiap label yang berupa *word* menjadi sebuah simbol tunggal, selanjutnya diperoleh sebuah FSA.

Berdasarkan teorema ekivalensi antara FSA dengan sebuah bahasa reguler maka pastilah di sana ada sebuah bahasa reguler yang ekivalen yang himpunan terminalnya adalah himpunan simbol tunggal tadi. Ini berarti, setiap gFSA ekivalen dengan sebuah FSA, sedang FSA ekivalen dengan sebuah bahasa reguler maka gFSA ekivalen dengan sebuah bahasa reguler dengan demikian dapat dibuktikan bahwa  $L(gFSA)$  adalah sebuah bahasa reguler.

Akan tetapi, usaha untuk menerapkan (5.1) dan (5.2) pada gagasan membangun suatu bahasa formal yang reguler untuk pengguna dalam berinteraksi dengan ARS secara lebih sederhana, (5.1) dan (5.2) dapat disederhanakan dengan menulis Gambar 5.9 sebagaimana Gambar 5.10.

Gambar 5.10. adalah penyederhanaan yang didasarkan pada asumsi bahwa  $L_{AB} = L_{BD} = L_{DD} = L_{AC} = L_{CD} = L_{BC}$  atau dapat juga dinyatakan dalam bentuk yang lebih luas yaitu  $L_{BD}, L_{DD}, L_{AC}, L_{CD}, L_{BC} \subseteq L_{AB}$ .



**Gambar 5.10. Penyederhanaan gFSA**

Bahasa pengguna yang diterima oleh gFSA dapat ditulis sebagai berikut:

$$L(\text{gFSA}) = L_{AD}$$

$$L_{AD} = (L_{AB} \circ L_{BD}) \cup (L_{AC} \circ L_{CD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD}) \cup (L_{AB} \circ L_{BD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD}) \cup (L_{AB} \circ L_{BC} \circ L_{CD} \circ L_{DD} \dots \circ L_{DD})$$

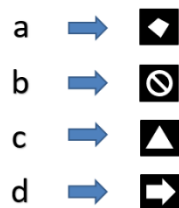
$$L_{AD} = (L_{AB} \circ L_{AB}) \cup (L_{AB} \circ L_{AB}) \cup (L_{AB} \circ L_{AB} \circ L_{AB}) \cup (L_{AB} \circ L_{AB} \circ L_{AB} \dots \circ L_{AB}) \cup (L_{AB} \circ L_{AB} \circ L_{AB} \dots \circ L_{AB}) \cup (L_{AB} \circ L_{AB} \circ L_{AB} \circ L_{AB} \dots \circ L_{AB})$$

$$L_{AD} = \{abcbd \oplus abcd\} \cup \{aba \oplus ccd\} \cup \{abcbd \oplus bd \oplus ccd\} \cup \{abcbd \oplus abcd \oplus b \dots \oplus b\} \cup \{aba \oplus ccd \oplus b \dots \oplus b\} \cup \{abcbd \oplus bd \oplus ccd \oplus b \dots \oplus b\}$$

$$L_{AD} = \{abcbdabcbd\} \cup \{abaccd\} \cup \{abcbd b ccd\} \cup \{abcbdabcbd b \dots b\} \cup \{abaccd b \dots b\} \cup \{abcbd b ccd b \dots b\}$$

$$L_{AD} = \{ abcbdabcbd, abaccd, abcbd b ccd, abcbdabcbd b \dots b, abaccd b \dots b, abcbd b ccd b \dots b \}$$

Selanjutnya, abjad dipetakan kepada penanda untuk berinteraksi, yaitu dapat dibuat himpunan *word* penanda yang menjadi bahasa pengguna terhadap ARS. Contoh Gambar 5.11 mengilustrasikan pemetaan abjad kepada penanda.



**Gambar 5.11. Pemetaan Abjad ke Penanda**

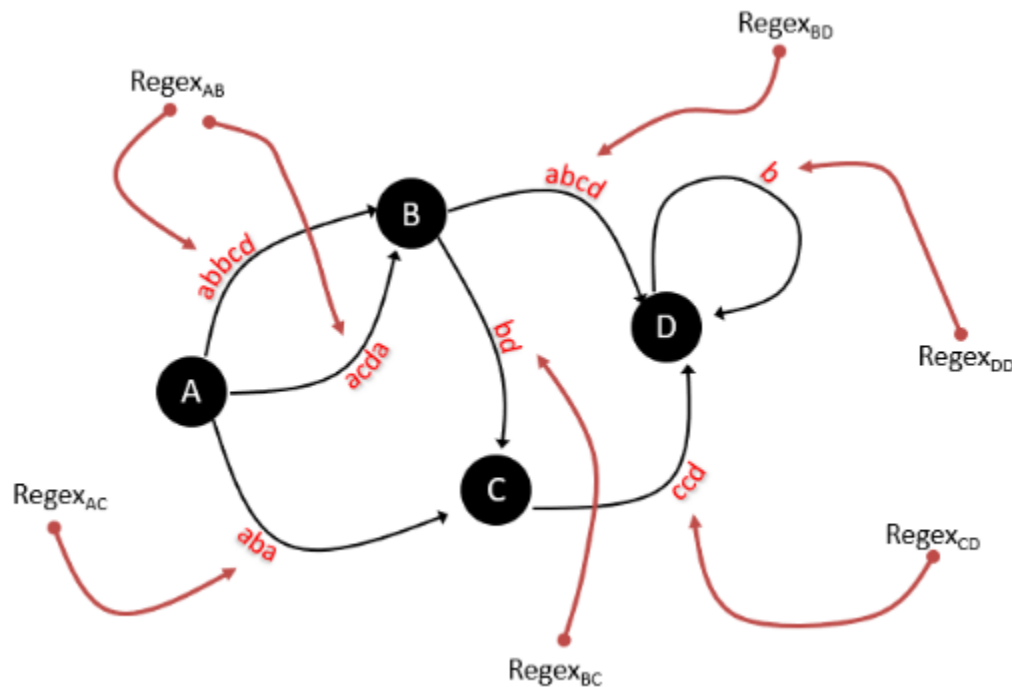


Dengan pemetaan Gambar 5.11, misalkan  $abbcdbcd \in L_{AD} = L(gFSA)$  disajikan pengguna dalam kombinasi penanda sebagai *word* penanda sebagaimana Gambar 5.12.



**Gambar 5.12. Word Penanda yang Disajikan Pengguna Menggunakan  $L(gFSA)$**

Penyajian suatu *generalized finite state automata* dapat diperluas kepada penggunaan ekspresi regular. Gambar 5.12.1 menyajikan ilustrasi tentang bagaimana jika sebuah gFSA dinyatakan dalam transisi-transisi regex.



**Gambar 5.12.1. Ekspresi sebuah gFSA menggunakan ekspresi regular**

### 1.1. Kesimpulan Proposisi 2

Dengan demikian, sampai disini telah ditunjukkan sebuah untai argumentasi yang menunjukkan bahwa proposisi 2 adalah benar di sana ada sebuah bahasa reguler menggunakan gFSA yang dapat dikonstruksikan untuk membangun bahasa interaksi antara pengguna dan ARS. Ini tentunya juga memberi argumentasi bagi konjektur 2 dan konjektur 3.

Contoh tambahan tentang bagaimana menghasilkan 10 respon dengan hanya 3 penanda (marker) saja

Misalkan terdapat 3 buah penanda, yaitu:



**Gambar 5.13. Contoh 3 buah marker dengan enkodingnya**

Rancangan ini dapat membangun lebih dari 3 respon AR yang mungkin dengan membangun *word* atau *multiword* dari kombinasi ketiga *marker* ini. Setiap kombinasi mendapatkan 1 respon AR yang unik. Di bawah ini contoh kombinasi *word marker* yang bisa diperoleh sehingga dibuat sebuah bahasa L sebagai bahasa pengguna untuk interaksi pengguna-sistem AR.

Dari enkoding *marker* ke a,b,c diperoleh kombinasi *word*:

Untuk panjang 3:

1. abc
1. acb
2. bac
3. bca
4. cab
5. cba
6. aab
7. aba
8. baa
9. aac
10. aca
11. caa
12. ccb
13. cbc
14. bcc
15. cca
16. cac
17. acc
18. bbc
19. bcb
20. cbb

21. bba

22. bab

23. abb

Untuk panjang 2:

24. aa

25. bb

26. cc

27. ab

28. ac

29. ba

30. bc

31. ca

32. cb

Untuk panjang 1:

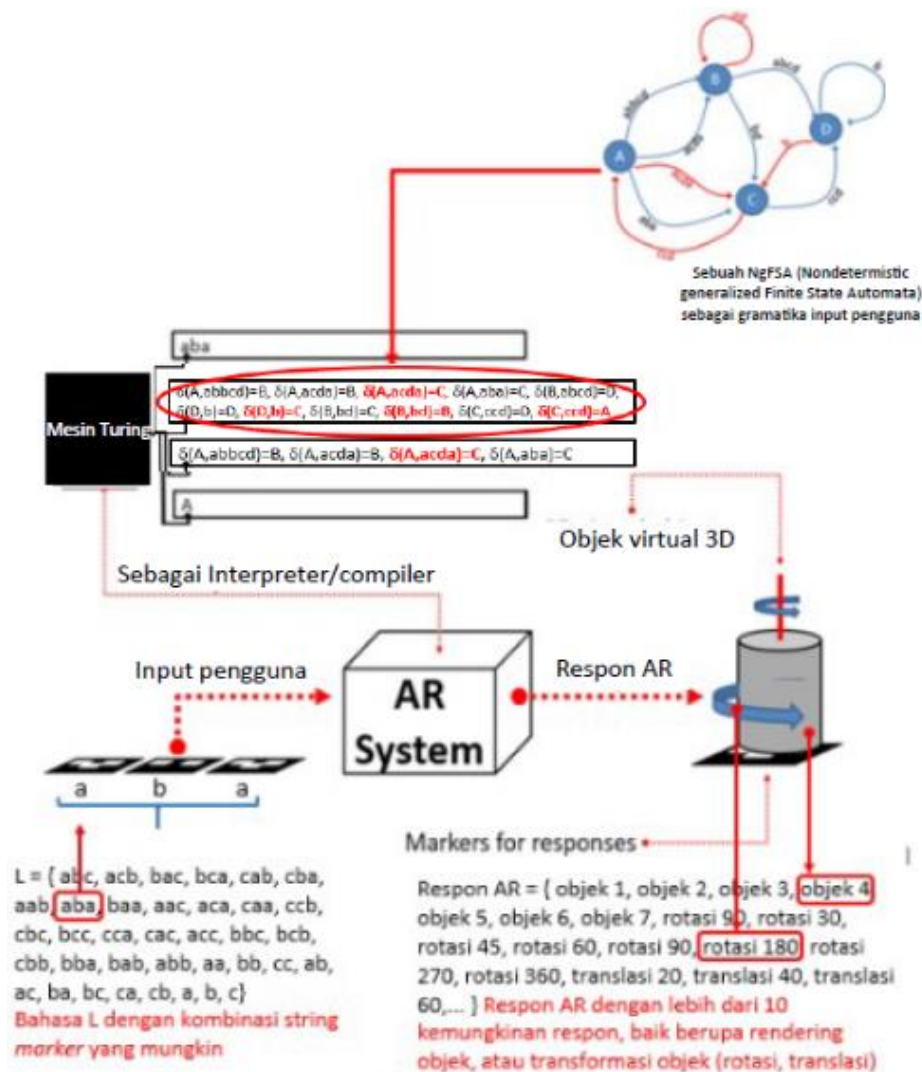
33. a

34. b

35. c

Dari semua kombinasi ini dapat diperoleh sebuah bahasa  $L$  yang himpunan wordnya adalah  $L = \{ abc, acb, bac, bca, cab, cba, aab, aba, baa, aac, aca, caa, ccb, cbc, bcc, cca, cac, acc, bbc, bcb, cbb, bba, bab, abb, aa, bb, cc, ab, ac, ba, bc, ca, cb, a, b, c \}$ .

Dari uraian di atas dapat dilihat bahwa hanya dengan 3 *marker* dapat diperoleh maksimal 36 respon AR yang unik satu sama lain. Lebih dari 10 respon AR yang dibutuhkan.



**Gambar 5.14. Jumlah 3 buah marker yang daripadanya dapat diperoleh lebih dari 10 respon yang berbeda**

## PENUTUP

Dari semua deskripsi di atas, telah dikemukakan dengan lengkap gagasan rancangan interaksi antara sistem Augmented Reality dengan manusia berbasis marker menggunakan bahasa interaksi yang gramatiknya sepadan dengan *generalized Finite State Automata*, yaitu  $L(gFSA)$ .

## DAFTAR PUSTAKA

Alur, R. and Dill, D. L. (1994) 'A theory of *timed automata*', *Theoretical Computer Science*, 126(2), pp. 183–235. doi: 10.1016/0304-3975(94)90010-8.

- Bradley, D. (2013) 'Adaptive Thresholding using the Integral Image', *the Journal of Graphics Tools. Volume 12, Issue 2.*, (January 2007). doi: 10.1080/2151237X.2007.10129236.
- D, G., Kumar, O. and Ram, S. (2016) 'Marker Based Augmented Reality Application in Education: Teaching and Learning', *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 4(Viii), pp. 153–158. Available at: [www.ijraset.com](http://www.ijraset.com).
- Hincapié, M. *et al.* (2011) 'An introduction to Augmented Reality with applications in aeronautical maintenance', *International Conference on Transparent Optical Networks*, pp. 1–4. doi: 10.1109/ICTON.2011.5970856.
- Hosch, W. L. (2017) *augmented reality | computer science | Britannica.com, Encyclopædia Britannica, Inc.* Available at: <https://www.britannica.com/technology/augmented-reality> (Accessed: 1 September 2017).
- Ilievski, M. and Trajkovik, V. (2013) 'An Approach to Both Standardized and Platform Independent Augmented Reality Using Web Technologies', *ICT Innovations 2012, AISC 207, Springer-Verlag Berlin Heidelberg 2013*, pp. 195–203. doi: 10.1007/978-3-642-37169-1.
- Khalifa, F. *et al.* (2015) 'Local Detectors and Descriptors for Object Class Recognition', *Ijisa*, (September), pp. 12–18. doi: 10.5815/ijisa.2015.10.02.
- Kim, D., Moon, W. and Kim, S. (2014) 'A Study on Method of Advanced Marker Array', *International Journal of Software Engineering and Its Applications*, 8(6), pp. 1–16.
- Kurpyte, D. and Navakauskas, D. (2014) 'An Efficiency Analysis of Augmented Reality Marker Recognition Algorithm', *Electrical, Control and Communication Engineering, De Gruyter Open*. doi: 10.2478/ecce-2014-0008.
- Lim, C. *et al.* (2016) 'Mobile Augmented Reality Based on Invisible Marker', *2016 IEEE International Symposium on Mixed and Augmented Reality Adjunct Proceedings*, pp. 78–81. doi: 10.1109/ISMAR-Adjunct.2016.38.
- Mota, J. M. *et al.* (2018) 'Augmented reality mobile app development for all', *Computers and Electrical Engineering*, 65, pp. 250–260. doi: 10.1016/j.compeleceng.2017.08.025.
- Pagani, A. (2014) 'Modeling Reality for Camera Registration in Augmented Reality Applications', *Springer-Verlag Berlin Heidelberg 2014*, pp. 321–324. doi: 10.1007/s13218-014-0320-5.
- Pawade, D. and Sakhapara, A. (2018) 'Augmented Reality Based Campus Guide Application Using Feature Points Object Detection', *Ijitcs*, (May), pp. 76–85. doi: 10.5815/ijitcs.2018.05.08.
- Rafał Wojciechowski (2012) *Modeling Interactive Augmented Reality*. Springer-Verlag London Limited 2012 137. doi: 10.1007/978-1-4471-2497-9.

- Sánchez, A. P. and Shapiro, M. (2016) ‘Growth in higher Baumslag-Solitar groups’, *arxiv1605.01131v4*, pp. 1–17. Available at: <http://arxiv.org/abs/1605.01131>.
- Shetty, M., Lasrado, V. and Mohammed, R. (2015) ‘Marker Based Application in Augmented Reality Using Android’, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 3(Special Issue 7, October 2015), pp. 146–151.
- Siltanen, S. (2012) *Theory and applications of marker-based augmented reality, Espoo 2012. VTT Science Series 3. JULKAISIJA – UTGIVARE – PUBLISHER*. Available at: <http://www.vtt.fi/publications/index.jsp>.
- Sin, A. K. and Zaman, H. B. (2009) ‘Tangible Interaction in Learning Astronomy through Augmented Reality Book-Based Educational Tool’, *IVIC 2009, LNCS 5857, Springer-Verlag Berlin Heidelberg 2009*, pp. 302–313.
- TATENO, K., KITAHARA, I. and OHTA, Y. (2017) ‘A Nested Marker for Augmented Reality’, *ResearchGate*, (April). doi: 10.1109/VR.2007.352495.
- Turing, A. M. (1938) ‘On computable numbers, with an application to the entscheidungsproblem. a correction’, *Proceedings of the London Mathematical Society*, s2-43(1), pp. 544–546. doi: 10.1112/plms/s2-43.6.544.
- Vera, L. *et al.* (2011) ‘Augmented Mirror: Interactive Augmented Reality System Based on Kinect’, *INTERACT 2011, Part IV, LNCS 6949, IFIP International Federation for Information Processing 2011*, pp. 483–486.
- Waruwu, A. F., Bayupati, I. P. A. and Darma Putra, I. K. G. (2015) ‘Augmented Reality Mobile Application of Balinese Hindu Temples : DewataAR’, *Ijcnis*, (January), pp. 59–66. doi: 10.1109/IJCNIS.2015.7388888.
- Yao, Y. *et al.* (2008) ‘Augmented Reality Interactive Interface for Defective Bone Repair System’, *APCMBE 2008, IFMBE Proceedings 19, Springer-Verlag Berlin Heidelberg 2008*, pp. 733–736.
- Zhang, Q. and Lew, M. S. (2012) ‘The Leiden Augmented Reality System ( LARS ) Leiden Augmented Reality System ( LARS )’, *Springer-Verlag Berlin Heidelberg 2012*, pp. 639–642.