PROSIDING Seminar Nasional SISFOTEK ISSN 2597-3584 (Media Online)
Sistem Informasi dan Teknologi
website : www.seminar.iaii.or.id | email : seminar@iaii.or.id

Representasi Pengetahuan Sistem Pakar Menggunakan Struktur Data Redis

Haddad Sammir[1], Khairil Hamdi[2], Budi Sunaryo[3]
[1,2]Sistem Informasi, STMIK Jaya Nusa
[3]Teknologi Rekayasa Komputer Jaringan, Fakultas Teknologi Industri, Universitas Bung Hatta
hsammir@jayanusa.ac.id

Abstract

This research discusses knowledge representation in expert systems using the Redis data structure, with a focus on the knowledge-based inference process. Inference in an expert system involves comparing existing facts with an existing database. When the elements in the rule match the database, the rule is considered TRUE and processed further in the database. For efficiency and speed in the inference process, this research utilizes the Redis data structure, especially the "sets" data structure type, which allows set operations to be carried out quickly and effectively. This research also suggests a convention for writing keys and values in Redis so that data consistency and integrity is maintained. Through this implementation, it is hoped that the expert system can process and make decisions more efficiently, utilizing Redis' advantages in set-based data management and access.

Keywords: Expert System, Redis Data Structure, Inference Process, Data Sets Structure

Abstrak

Penelitian ini membahas representasi pengetahuan dalam sistem pakar menggunakan struktur data Redis, dengan fokus pada proses inferensi berbasis pengetahuan. Inferensi dalam sistem pakar melibatkan perbandingan fakta yang dimiliki dengan basis data yang ada. Ketika elemen pada aturan (rule) sesuai dengan basis data, aturan tersebut dianggap benar (TRUE) dan diproses lebih lanjut dalam basis data. Untuk efisiensi dan kecepatan dalam proses inferensi, penelitian ini memanfaatkan struktur data Redis, khususnya jenis struktur data “sets”, yang memungkinkan operasi himpunan dilakukan dengan cepat dan efektif. Penelitian ini juga menyarankan konvensi penulisan key dan value dalam Redis agar konsistensi dan integritas data terjaga. Melalui implementasi ini, diharapkan sistem pakar dapat memproses dan mengambil keputusan dengan lebih efisien, memanfaatkan keunggulan Redis dalam manajemen dan akses data berbasis himpunan.

Kata kunci: Sistem Pakar, Struktur Data Redis, Proses Inferensi, Struktur Data Sets

1. Pendahuluan

Sistem pakar merupakan perangkat lunak yang sangat banyak ditemui saat ini. Dukungan data dan metoda yang ada memberi dukungan kuat berkembangnya system tersebut. Sistem pakar dikembangkan dan digunakan untuk berbagai keperluan pada berbagai domain kepakaran. Pada domain kesehatan, sistem pakar dikembangkan untuk diagnosa penyakit malaria [1], diagnosa awal penyakit jantung [2], serta diagnosa penyakit selama kehamilan [3].

Selain pada domain kesehatan, implementasi sistem pakar juga dilakukan pada domain pendidikan yang dilakukan oleh Haerudin, Iqbaludin, F. Noer et al yang membahas pemilihan pendidikan berdasarkan minat dan bakat siswa [4]. Pada domain pertanian, sistem pakar juga diterapkan pada diagnosa penyakit dan hama tanaman jagung [5].

Metoda yang umum digunakan pada sistem pakar diantaranya certainty factor dibahas pada penelitian Marbun et al tentang diagnosa penyakit kolesterol pada remaja, berikutnya metoda naïve bayes, digunakan pada penelitan Dwiramadan et al untuk mendiagnosa penyakit kulit kucing [6] dan algoritma *forward chaining* yang dibahas Nawangnugraeni pada artikel yang berjudul “Sistem Pakar Berbasis Android untuk Diagnosis Diabetes Melitus dengan Metode *Forward Chaining”* [7].

Penelitian ini berfokus pada sistem pakar berbasis aturan dengan pendekatan metoda *forward chaining* sebagai alur inferensinya. Proses inferensi dalam sistem pakar adalah inti dari operasionalnya, di mana fakta-fakta yang dimiliki dibandingkan dengan basis data untuk menentukan hasil atau keputusan. Ketika aturan (*rule*) dalam sistem sesuai dengan fakta yang ada, aturan tersebut diaktifkan dan diproses lebih lanjut.

Namun, dengan meningkatnya kompleksitas data dan jumlah aturan yang harus diproses, tantangan terkait efisiensi dan kecepatan dalam proses inferensi menjadi lebih signifikan. Oleh karena itu, diperlukan solusi yang dapat menangani operasi data secara cepat dan efektif. Redis, sebagai sistem manajemen basis data berbasis

memori yang mendukung berbagai struktur data, menawarkan potensi untuk mengatasi tantangan ini.

Dalam penelitian ini, fokus diberikan pada penerapan struktur data Redis untuk representasi pengetahuan dalam sistem pakar. Struktur data "sets" dalam Redis memungkinkan operasi himpunan dilakukan dengan efisien, yang berpotensi mengoptimalkan proses inferensi dengan membandingkan fakta dengan aturan-aturan dalam basis data. Penelitian ini juga membahas pentingnya konvensi penulisan key dan value dalam Redis untuk menjaga konsistensi dan integritas data. Dengan menerapkan pendekatan ini, diharapkan sistem pakar dapat meningkatkan efisiensi dan akurasi dalam pengambilan keputusan, memanfaatkan keunggulan Redis dalam manajemen dan akses data berbasis himpunan.

2. Metode Penelitian

Sistem pakar atau *expert system* merupakan system informasi yang memili basis pengetahuan para pakar sehingga dapat digunakan sebagai representasi kepakaran serta dapat digunakan untuk berkonsultasi [8]. Sistem pakar sendiri bertujuan untuk menciptakan sarana konsultasi yang mampu memberikan saran dan jawaban setara dengan seorang pakar [9]. Manfaat yang diberikan system pakar menurut Y. Anggraini, M. Indra, M. Khoirusofi, et al. dalam artikel Systematic Literature Review: Sistem Pakar Diagnosa Penyakit Gigi Menggunakan Metode Forward Chaining adalah memberikan hasil diagnosa yang akurat dan meningkatkan efisiensi waktu [10].

Representasi pengetahuan adalah cara informasi dan pengetahuan dikodifikasikan agar dapat diproses oleh sistem komputer. Dalam sistem pakar, representasi pengetahuan mencakup teknik-teknik untuk menyimpan dan mengorganisasi pengetahuan agar dapat digunakan dalam proses inferensi dan pengambilan keputusan. Teknik representasi pengetahuan meliputi penggunaan basis data, ontologi, *frame*, dan aturan. Representasi yang efektif memungkinkan sistem pakar untuk menyimpan pengetahuan dengan cara yang mendukung akurasi dan efisiensi dalam proses pengambilan keputusan.

Redis merupakan salah satu sistem pengelolaan basis data berbasis *key : value*. Redis merupakan salah satu basis data NoSql yang tidak memiliki definisi formal. Pada Redis data disimpan secara redundant sehingga membuat operasi *query* menjadi sederhana [11]. Sifat dari redis yang menyimpan data pada memory memberikan keuntungan dalam hal kecepatan hingga 150.000 kali lebih cepat dibandingkan dengan basis data yang menyimpan datanya di dalam *hard disk* [12].

Penelitian ini dirancang dengan memodelkan metode system pakar berbasis aturan ke dalam teori himpunan, memadankannya dengan struktur data “sets” pada redis lalu memanfaatkan operasi sets pada struktur data redis tersebut untuk mensimulasikan peroses inferensi.

Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Memodelkan konsep “IF .. THEN” ke dalam konsep himpunan.
2. Memodelkan operasi “AND” dan “OR” ke dalam teori himpunan.
3. Mengimplementasikan model himpunan ke dalam struktur data redis “sets”.
4. Mengimplementasikan operasi “AND” adan “OR” menggunakan operasi sets pada redis.
5. Pengujian menggunakan metoda *forward chaining*.

3. Hasil dan Pembahasan

3.1. Pemodelan Rule Ke Dalam Teori Himpunan

Proses inferensi pada system pakar berbasis pengetahuan adalah dengan membandingkan fakta-fakta yang dimiliki dengan basis data yang tersedia. Jika elemen pada rule tersedia pada basis data, maka rule bernilai *TRUE* dan rule tersebut di-*fired* ke dalam basis data. Misalkan basis data memiliki kumpulan fakta: A, B, C, D, E, lalu terdapat aturan: “IF C is true AND D is true THEN F is true” maka proses pencocokan bernilai TRUE karena C dan D terdapat di dalam fakta, sehingga aturan tersebut di-fired dan database memuat fakta baru yaitu: A, B, C, D, E, F. Dengan mengamati proses inferensi di atas, dapat dimodelkan mekanisme inferensi menggunakan himpunan.

Jika diketahui himpunan fakta: {A, B, C, D, E} dan himpunan aturan F : {C, D}, maka aturan F bernilai benar jika himpunan aturan adalah *subset* dari himpunan fakta. Contoh di atas dapat ditulis aturan $\subseteq$ fakta. Jika himpunan aturan bukan subset dari fakta, maka aturan tersebut bernilai salah.

Pemodelan data menggunakan sub set hanya berlaku jika aturan menggunakan operator “AND”. Jika aturan menggunakan operator “OR”, maka operasi yang dilakukan adalah operasi irisan himpunan. Misalkan diketahui rule sebgai berikut: “IF SEASON is SPRING AND COLOR is WHITE OR COLOR is YELLOW OR COLOR is RED THEN flower is Freesia” maka himpunan yang dibuat menjadi: aturan freesia_and = {SEASON_SPRING} dan aturan freesia_or = {COLOR_WHITE, COLOR_YELLOW, COLOR_RED}. Jika diketahui himpunan fakta = {SEASON_SPRING, COLOR_WHITE} maka agar aturan flower is freesia bernilai benar himpunan aturan freesia_and harus sub set dari himpunan fakta dan himpunan aturan freesia_or memiliki irisan dengan himpunan fakta.

3.2. Implementasi Himpunan Ke Struktur Data Redis

Proses operasi himpunan di atas dapat dikomputasikan menggunakan struktur data redis “sets”. Penelitian ini memanfaatkan baik key dan value dalam stuktur data redis sehingga perlu membuat konvensi penulisan key dan value agar tetap konsisten. Konvensi penulisan ini

memanfaatkan perintah “keys” untuk melakukan pencarian key. Konvensi penulisan key / value adalah sebagai berikut:

1. Key / value adalah alfanumerik dan ditulis menggunakan huruf kecil.
2. Key / value ditulis dengan menyertakan pemisah menggunakan karakter ”:” sebagai identifikasi dan penglompokkan dengan format [prefix]:[objek]:[nilai]:[suffix and/or].
3. Key / value tidak boleh memiliki spasi. Spasi dapat digantikan dengan underscore “_”.
4. Prefix terdiri dari “rule” (mengidentifikasikan bahwa set key bersangkutan adalah rule, “db” mengidentifikasi bahwa set adalah database dan “ruleset” yang mengidentifikasikan set bersangkutan adalah ruleset.
5. Suffix and / or dapat berisi “and” yang mengidentifikasikan rule bersangkutan memiliki operator “and” atau “or” jika rule tersebut memiliki operator “or”.
6. Suffix or memerlukan tambahan informasi objek agar dapat menangani lebih dari satu klausa or, sehingga premis “OR color is purple” di tulis sebagai “rule:flower_name:iris:or:color” yang dapat diartikan bahwa rule tersebut menghimpun kalusa “or” pada konteks color.
7. Ruleset adalah kumpulan rule untuk sebuah premis. Ruleset terdiri dari sekumpulan key yang menampung rule dari sebuah premis.
8. Object dan value ditulis dalam format object:value. Sebagai contoh, “color is blue” ditulis sebagai “color:blue”.

3.3. Contoh Penerapan Rule Ke Dalam Struktur Data Redis

Diketahui sebuah rule dengan premis berikut ini.

```
IF season is summer
AND color is blue
OR color is purple
OR color is yellow
AND life type is perennial
AND root type is bulb
THEN flower name is iris
```

Konversi ke dalam struktur data redis dilakukan dengan pertama sekali mengidentifikasi operator yang digunakan pada rule. Rule di atas menggunakan operator AND dan operator OR dengan satu konteks (color) sehingga dapat dikelompokkan menjadi dua set rule yaitu “rule:flower_name:iris:and” dan “rule:flower_name:iris:or:color”.

```
sadd rule:flower_name:iris:and season:summer
sadd                  rule:flower_name:iris:and
life_type:perennial
sadd rule:flower_name:iris:and root_type:bulb
```

Struktur data redis di atas menciptakan sebuah set dengan key “rule:flower_name:iris:and” yang memiliki elemen season:summer, life_type:perennial, root_type:bulb. Struktur ini mewakili premis season is summer AND life type is perennial AND root type is bulb AND life type is perennial.

```
sadd rule:flower_name:iris:or:color color:blue

sadd            rule:flower_name:iris:or:color
color:purple

sadd            rule:flower_name:iris:or:color
color:yellow
```

Struktur data redis di atas menciptakan sebuah set dengan key “rule:flower_name:iris:or:color” yang memiliki elemen color:blue, color:purple, color:yellow. Stuktur data tersebut mewakili premis color is blue OR color is purple OR color is yellow.

sadd ruleset:flower_name:iris rule:flower_name:iris:and

sadd ruleset:flower_name:iris rule:flower_name:iris:or:color

Strutur data tersebut menciptakan set dengan key ruleset:flower_name:iris dan elemen rule:flower_name:iris:and, rule:flower_name:iris:or:color . Ruleset menghimpun daftar key yang merepresentasikan premis rule flower name is iris di atas.

3.4 Studi Kasus White Lily

Kasus white lily adalah kasus system pakar yang bertujuan untuk menentukan jenis bunga berdasarkan sekumpulan rule menggunakan data yang dimiliki. Operasi kasus white lily menggunakan struktur data redis adalah seperti tabel 1 berikut ini.

Tabel 1. Konversi Rule ke Struktur Data Redis

Rule	Struktur data redis
Rule 1: IF size > 10 AND size <50 THEN height is small	sadd rule:height:small:or:size size:10 sadd rule:height:small:or:size size:11 sadd rule:height:small:or:size size:12 sadd rule:height:small:or:size size:13 sadd rule:height:small:or:size size:14 sadd rule:height:small:or:size size:15 … … sadd rule:height:small:or:size size:45 sadd rule:height:small:or:size size:46 sadd rule:height:small:or:size size:47 sadd rule:height:small:or:size size:48 sadd rule:height:small:or:size size:49 sadd rule:height:small:or:size size:50

Rule	Struktur data redis
Rule 2: IF size > 50 AND size <150 THEN height is medium	sadd rule:height:medium:or:size size:51 sadd rule:height:medium:or:size size:52 sadd rule:height:medium:or:size size:53 sadd rule:height:medium:or:size size:54 sadd rule:height:medium:or:size size:55 … … sadd rule:height:medium:or:size size:146 sadd rule:height:medium:or:size size:147 sadd rule:height:medium:or:size size:148 sadd rule:height:medium:or:size size:149 sadd rule:height:medium:or:size size:150
Rule 3: IF size > 150 THEN height is tall	sadd rule:height:tall:or:size size:151 sadd rule:height:tall:or:size size:152 sadd rule:height:tall:or:size size:153 sadd rule:height:tall:or:size size:154 sadd rule:height:tall:or:size size:155 … … sadd rule:height:tall:or:size size:165 sadd rule:height:tall:or:size size:166 sadd rule:height:tall:or:size size:167 sadd rule:height:tall:or:size size:168 sadd rule:height:tall:or:size size:169 sadd rule:height:tall:or:size size:170
Ruleset	sadd ruleset:height:small rule:height:small:or:size sadd ruleset:height:medium rule:height:medium:or:size sadd ruleset:height:tall rule:height:tall:or:size
Rule 4: IF life cycle is one year THEN life type is annual	sadd rule:life_type:annual:and life_cycle:one_year sadd ruleset:life_type:annual rule:life_type:annual:and
Rule 5: IF life cycle is more than one year THEN life type is perennial	sadd rule:life_type:perennial:and life_cycle:more_than_one_year sadd ruleset:life_type:perennial rule:life_type:perennial:and
Rule 6: IF season is summer AND color is blue OR color is purple	sadd rule:flower_name:iris:and season:summer sadd rule:flower_name:iris:and life_type:perennial sadd rule:flower_name:iris:and root_type:bulb
OR color is yellow AND life type is perennial AND root type is bulb THEN flower name is iris	sadd rule:flower_name:iris:or:color color:blue sadd rule:flower_name:iris:or:color color:purple sadd rule:flower_name:iris:or:color color:yellow sadd ruleset:flower_name:iris rule:flower_name:iris:and sadd ruleset:flower_name:iris rule:flower_name:iris:or:color
Rule 7: IF season is autumn AND color is white OR color is pink OR color is pinkish-red THEN flower name is anemone	sadd rule:flower_name:anemone:and season:autumn sadd rule:flower_name:anemone:or:color color:white sadd rule:flower_name:anemone:or:color color:pink sadd rule:flower_name:anemone:or:color color:pinkish-red sadd ruleset:flower_name:anemone rule:flower_name:anemone:and sadd ruleset:flower_name:anemone rule:flower_name:anemone:or:color
Rule 8: IF season is autumn AND height is medium AND color is yellow OR color is while OR color is purple OR color is red THEN flower name is Chrysanthemum	sadd rule:flower_name:chrysanthemum:and season:autumn sadd rule:flower_name:chrysanthemum:and height:medium sadd rule:flower_name:chrysanthemum:or:color color:yellow sadd rule:flower_name:chrysanthemum:or:color color:white sadd rule:flower_name:chrysanthemum:or:color color:purple sadd rule:flower_name:chrysanthemum:or:color color:red sadd ruleset:flower_name:chrysanthemum rule:flower_name:chrysanthemum:and sadd ruleset:flower_name:chrysanthemum rule:flower_name:chrysanthemum:or:color
Rule 9: IF season is spring AND root type is bulbs AND color is white	sadd rule:flower_name:freesia:and season:spring sadd rule:flower_name:freesia:and root_type:bulb sadd rule:flower_name:freesia:and perfumed:true

Rule	Struktur data redis
OR color is yellow OR color is orange OR color is purple OR color is red OR color is blue AND perfumed is true THEN flower is Freesia	sadd rule:flower_name:freesia:or:color color:white sadd rule:flower_name:freesia:or:color color:yellow sadd rule:flower_name:freesia:or:color color:orange sadd rule:flower_name:freesia:or:color color:purple sadd rule:flower_name:freesia:or:color color:red sadd rule:flower_name:freesia:or:color color:blue sadd ruleset:flower_name:freesia rule:flower_name:freesia:and sadd ruleset:flower_name:freesia rule:flower_name:freesia:or:color
Rule 10: IF life type is perennial AND height is tall AND root type is bulbs AND season is summer THEN flower name is Dahlia	sadd rule:flower_name:dahlia:and life_type:perennial sadd rule:flower_name:dahlia:and height:tall sadd rule:flower_name:dahlia:and root_type:bulb sadd rule:flower_name:dahlia:and season:summer sadd ruleset:flower_name:dahlia rule:flower_name:dahlia:and
Rule 11: IF season is spring AND root type is bulbs AND color is yellow OR color is white THEN flower name is Narcissus	sadd rule:flower_name:narcissus:and season:spring sadd rule:flower_name:narcissus:and root_type:bulb sadd rule:flower_name:narcissus:or:color color:yellow sadd rule:flower_name:narcissus:or:color color:white sadd ruleset:flower_name:narcissus rule:flower_name:narcissus:and sadd ruleset:flower_name:narcissus rule:flower_name:narcissus:or:color
Rule 12: IF soil is acidic AND color is white OR color is pink OR color is red	sadd rule:flower_name:camellia:and soil:acidic sadd rule:flower_name:camellia:and life_type:perennial sadd rule:flower_name:camellia:and root_type:roots

Rule	Struktur data redis
AND life type is perennial AND root type is roots THEN flower name is Camellia	sadd rule:flower_name:camellia:or:color color:white sadd rule:flower_name:camellia:or:color color:pink sadd rule:flower_name:camellia:or:color color:red sadd ruleset:flower_name:camellia rule:flower_name:camellia:and sadd ruleset:flower_name:camellia rule:flower_name:camellia:or:color
Rule 13: IF season is spring AND root type is bulbs AND perfumed is true AND height is small AND life type is perennial THEN flower name is Lily	sadd rule:flower_name:lily:and season:spring sadd rule:flower_name:lily:and root_type:bulb sadd rule:flower_name:lily:and perfumed:true sadd rule:flower_name:lily:and height:small sadd rule:flower_name:lily:and life_type:perennial sadd ruleset:flower_name:lily rule:flower_name:lily:and
Rule 14: IF height is small AND life type is annual AND soil is rich OR soil is loose OR soil is fertile THEN flower name is Begonia	sadd rule:flower_name:begonia:and height:small sadd rule:flower_name:begonia:and life_type:annual sadd rule:flower_name:begonia:or:soil soil:rich sadd rule:flower_name:begonia:or:soil soil:loose sadd rule:flower_name:begonia:or:soil soil:fertile sadd ruleset:flower_name:begonia rule:flower_name:begonia:and sadd ruleset:flower_name:begonia rule:flower_name:begonia:or:soil
Rule 15: IF season is winter AND color is white OR color is pink OR color is red THEN flower name is Azalea	sadd rule:flower_name:azalea:and season:winter sadd rule:flower_name:azalea:or:color color:white sadd rule:flower_name:azalea:or:color color:pink sadd rule:flower_name:azalea:or:color color:red

Rule	Struktur data redis
	sadd ruleset:flower_name:azalea rule:flower_name:azalea:and sadd ruleset:flower_name:azalea rule:flower_name:azalea:or:colo r
Rule 16: IF life type is perennial AND root type is root AND color is white OR color is red OR color is blue OR color is yellow THEN flower is Anemone	sadd rule:flower_name:anemone_2:and life_type:perennial sadd rule:flower_name:anemone_2:and root_type:roots sadd rule:flower_name:anemone_2:or:c olor color:white sadd rule:flower_name:anemone_2:or:c olor color:red sadd rule:flower_name:anemone_2:or:c olor color:blue sadd rule:flower_name:anemone_2:or:c olor color:yellow sadd ruleset:flower_name:anemone_2 rule:flower_name:anemone_2:and sadd ruleset:flower_name:anemone_2 rule:flower_name:anemone_2:or:c olor
Rule 17: IF life type is perennial AND root type is roots AND color is white OR color is pink OR color is red OR color is yellow AND perfumed is true AND soil is well- drained THEN flower is rose	sadd rule:flower_name:rose:and life_type:perennial sadd rule:flower_name:rose:and root_type:roots sadd rule:flower_name:rose:and perfumed:true sadd rule:flower_name:rose:and soil:well-drained sadd rule:flower_name:rose:or:color color:white sadd rule:flower_name:rose:or:color color:pink sadd rule:flower_name:rose:or:color color:red sadd rule:flower_name:rose:or:color color:yellow sadd ruleset:flower_name:rose rule:flower_name:rose:and sadd ruleset:flower_name:rose rule:flower_name:rose:or:color
Rule 18: IF flower name is Lily AND perfumed is true THEN flower name is White lily	sadd rule:flower_name:white_lily:and flower_name:lily sadd rule:flower_name:white_lily:and perfumed:true sadd ruleset:flower_name:white_lily rule:flower_name:white_lily:and

3.5 Operasi Forward Chaining

Jika diketahui fakta “season: spring, root type: bulbs, perfumed: true, size: 16-18 cm, life cycle more than one year, color:orange,red, white, pink”, maka proses forward chaining untuk mendapatkan bahwa fakta ini mengarah ke “white lily” adalah sebagai berikut.

Pertama sekali fakta tersebut harus disimpan dalam database. Sesuai konvensi yang disampaikan sebelumnya, database ditulis dalam format “db:[objek]:[value]” dalam kasus ini database dapat ditulis sebagai “db:flower_name:white_lily” .

Langkah berikutnya adalah melakukan input rule mengikuti konvensi yang telah dijabarkan sebelumnya. Dalam contoh ini, rule didefinisikan seperti pada tabel 1 di atas. Operasi selanjutnya adalah melakukan operasi forward chaining dengan memanfaatkan struktur data dan operasi data pada redis.

Proses forward chaining secara umum melibatkan aktifitas perulangan / loop pada sekumpulan rule. Perulangan dilakukan selama masih ada pengetahuan baru yang ditambahkan (fired) ke dalam database. Perulangan dapat dihentikan apabila goal sudah tercapai atau sudah tidak ada lagi pengetahuan yang dapat ditambahkan ke dalam database.

Implementasi rule pada struktur data redis mengharuskan pemisahan bagian rule yang menggunakan operator AND dan bagian rule yang menggunakan operator OR. Oleh karena itu, sebuah rule dihimpun menggunakan set “ruleset”. Perulangan yang dilakukan menguji setiap elemen ruleset. Ruleset didapatkan dengan menggunakan perintah keys seperti pada Gambar 1.

```
[db1] > keys ruleset:*
1) "ruleset:life_type:annual"
2) "ruleset:flower_name:lily"
3) "ruleset:height:small"
4) "ruleset:flower_name:narcissus"
5) "ruleset:flower_name:iris"
6) "ruleset:height:medium"
7) "ruleset:flower_name:camellia"
```

Gambar 1. Mendapatkan Ruleset

Anggota rule set beserta elemen dari masing-masing anggotanya diakses menggunakan perintah yang diperlihatkan pada Gambar 2.

```
[db1] > smembers ruleset:flower_name:chrysanthemum
1) "rule:flower_name:chrysanthemum:or:color"
2) "rule:flower_name:chrysanthemum:and"

[db1] > smembers rule:flower_name:chrysanthemum:or:color
1) "color:yellow"
2) "color:purple"
3) "color:white"
4) "color:red"

[db1] > smembers rule:flower_name:chrysanthemum:and
1) "season:autumn"
2) "height:medium"

[db1] >
```

Gambar 2. Rule yang Mendefinisikan Bunga Chrysanthemum

Struktur data redis di atas menggambarkan rule bunga Chresanthemum sebagai berikut: “IF season is autumn AND height is medium AND color is yellow OR color is white OR color is purple OR color is red THEN flower name is Chrysanthemum “. Struktur data tadi akan diiriskan (intersect) ke database yang terdefinisi sebeumnya untuk mengetahui apakah rule tersebut dapat diterapkan. Sebuah ruleset dinyatakan berhasil jika semua anggotanya berhasil diocokkan dengan database. Jika sebuah ruleset memliki lebih dari 1 anggota, maka semua anggota ruleset tersebut harus berhasil dicocokkan, baru setelah itu objek / pengetahuan baru ditambahkan ke dalam database.

Mendapatkan irisan pada struktur data redis dilakukan melaui perintah “sinter [key1] [key2]” operasi ini dilakkukan untuk semua member (rule) yang terdapat pada ruleset. Kaidah yang digunakan adalah jika rule memiliki operator “AND”, maka semua member harus beririsan dengan database, sedangkan jika rule memiliki oprator “OR”, maka cukup salah satu member saja yang beririsan dengan database. Jika salah satu dari member ruleset gagal beririsan dengan database sesuai dengan kaidah yang digunakan, maka rule tersebut gagal, sebaliknya, jika semua rule berhasil beririsan, maka member dari setiap rule ditambahkan ke database sebagai pengetahuan baru.

Penambahan member / objek kedalam database dilakukan dengan cara yang diperlihatkan pada Gambar 3.

```
[db1] > smembers rule:life_type:perennial:and
1) "life_cycle:more_than_one_year"

[db1] > sinterstore tmp rule:life_type:perennial:and db:flower_name:white_lily
(integer) 1

[db1] > sinter rule:life_type:perennial:and db:flower_name:white_lily
1) "life_cycle:more_than_one_year"

[db1] > sadd db:flower_name:white_lily life_type:perennial
(integer) 1

[db1] > smembers db:flower_name:white_lily
1) "life_type:perennial"
2) "life_cycle:more_than_one_year"
```

Gambar 3. Menambahkan Objek ke Database

Gambar 3 memperlihatkan penambahan member dari objek “db:flower_name:white:lily”. Selanjutnya ruleset yang berhasil ditambahkan ke database dapat di-rename agar tidak digunakan lagi pada perulangan selanjutnya. Operasi rename terlihat seperti Gambar 4.

```
[db1] > rename ruleset:life_type:perennial 1ruleset:life_type:perennial
"OK"
```

Gambar 4. Menggati Nama Ruleset

Langkah terakhir adalah memberikan penanda / flag untuk menentukan apakah dalam perulangan tersebut terdapat operasi fired / penambahan pengetahuan ke dalam database. Flag berfungsi untuk menandakan apakah perulangan masih akan dilakukan. Jika pada penanda menginformasikan tidak ada fired, maka perulanga dapat dihentikan karena dianggap tidak ada pengetahuan baru yang dapat ditambahkan ke dalam database. Flag dapat dibuat seperti pada Gambar 5.

```
> set flag 1
"OK"
```

Gambar 5. Set Penanda / Flag

4. Kesimpulan

Menggunakan struktur data redis sebagai metoda penyimpanan basis data dan basis pengetahuan pada system pakar tidak hanya memberikan kemudahan dalam penyimpanan dan pengorganisasian, namun juga memberikan kinerja operasi yang baik mengingat redis beroperasi pada RAM yang memberikan kecepatan operasi yang jauh melebihi kecepatan operasi database berbasis disk. Selain itu, redis menyediakan antar muka kepada banyak bahasa pemrograman popular sehinggi operasi yang dijabarkan pada pemahasan di atas dapat diotomatisasi menggunakan bahasa pemrograman.

Dibalik kelebihannya, struktur data redis juga mempunyai kelemahan pada penyimpanan data yang besifat diskrit sehingga diharuskan menambahkan secara eksplisit sebuah jangkauan nilai. Sebagai contoh, struktur data redis mengharuskan menggunakan size:5, size:6, size:7 secara eksplisit dan tidak dapat menggunakan size > 4 ; size < 8.

Redis adalah salah satu basis data NoSql yang tidak hanya memberikan struktur data yang efisien, namun juga memberikan kinerja yang sangat baik. Penelitian selanjutnya dapat mengekplorasi basis data NoSql lainnya seperti MonggoDB yang menggunakan basis data berbasis dokumen dalam merancang representasi pengetahuan.

Daftar Rujukan

[1] A. L. Kalua, Veronika H, and D. T. Salaki, “Sistem Pakar Diagnosa Penyakit Malaria dengan Certainty Factor dan Forward Chaining,” *J. Inf. Technol. Softw. Eng. Comput. Sci.*, vol. 1, no. 1, pp. 22–34, 2022, doi: 10.58602/itsecs.v1i1.10.

[2] H. H. A. Rabbani, A. Jamaluddin, and A. Solehudin, “Sistem Pakar Diagnosa Awal Penyakit Jantung Menggunakan Metode Forward Chaining Dan Certainty Factor Berbasis Website,” *INFOTECH J.*, vol. 9, no. 2, pp. 442–451, 2023, doi: 10.31949/infotech.v9i2.6401.

[3] M. Ridho Handoko, “Sistem Pakar Diagnosa Penyakit Selama Kehamilan Menggunakan Metode Naive Bayes Berbasis Web,” *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 50–58, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI.

[4] Haerudin, Iqbaludin, F. I. Noer, and P. Rosyani, “Implementasi Metode Forward Chaining dalam Sistem Pakar Pemilihan Pendidikan Berdasarkan Minat dan Kemampuan Siswa,” *J. Ilmu Komput. dan Sci.*, vol. 2, no. 6, pp. 1681–1687, 2023.

[5] B. B. Suherman, “Sistem Pakar Diagnosa Penyakit Dan Hama Pada Tanaman Jagung Menggunakan Metode Naive Bayes,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 2, no. 3, pp. 390–398, 2021, doi: 10.33365/jatika.v2i3.1251.

[6] F. Dwiramadhan, M. I. Wahyuddin, and D. Hidayatullah, “Sistem Pakar Diagnosa Penyakit Kulit Kucing Menggunakan Metode Naive Bayes Berbasis Web,” *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 6, no. 3. pp. 429–437, 2022, doi: 10.35870/jtik.v6i3.466.

[7] D. A. NAWANGNUGRAENI, “Sistem Pakar Berbasis Android untuk Diagnosis Diabetes Melitus dengan Metode Forward Chaining,” *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 19–27, 2021, doi: 10.34010/komputika.v10i1.3553.

[8] E. T. Marbun, K. Erwansyah, and J. Hutagalung, “Sistem Pakar

Mendiagnosa Penyakit Kolesterol Pada Remaja Menggunakan Metode Certainty Factor," *J. Sist. Inf. Triguna Dharma (JURSI TGD)*, vol. 1, no. 4, p. 549, 2022, doi: 10.53513/jursi.v1i4.5686.

[9] A. Doni, A. Fadli, R. H. Maulana, Y. V. Putri, and P. Rosyani, "Analisis Metode Backward Chaining pada Sistem Pakar: Systematic Literature Review," *J. Manajemen, Ekon. Kewirausahaan,kesehatan,Pendidikan dan Inform.*, vol. 1, no. 4, pp. 144–151, 2023.

[10] Y. Anggraini, M. Indra, M. Khoirusofi, I. N. Azis, and P. Rosyani, "Systematic Literature Review: Sistem Pakar Diagnosa Penyakit Gigi Menggunakan Metode Forward Chaining," *BINER J. Ilmu Komputer, Tek. dan Multimed.*, vol. 1, no. 01, pp. 1–7, 2023, [Online]. Available: http://garuda.ristekdikti.go.id/.

[11] T. Taipalus, "Database management system performance comparisons: A systematic literature review," *J. Syst. Softw.*, vol. 208, p. 111872, 2024, doi: https://doi.org/10.1016/j.jss.2023.111872.

[12] A. I. Sanka, M. H. Chowdhury, and R. C. C. Cheung, "Efficient High-Performance FPGA-Redis Hybrid NoSQL Caching System for Blockchain Scalability," *Comput. Commun.*, vol. 169, pp. 81–91, 2021, doi: https://doi.org/10.1016/j.comcom.2021.01.017.